Manhattan 2

$1B R&D Initiative To Solve Global Warming & Diminishing Fossil Fuel Problems

By Glenn Weinreb

www.Manhattan2.org | February 5, 2019
ISBN: 9781794041530 |

Table of Contents

Chapter 1) $1B R&D Initiative to Create Low-Carbon Society

Problem

The world is facing two related long term crises. One is global warming the other is the accelerating depletion of coal, oil and natural gas reserves. Scientists that monitor global climate conditions recommend a reduction of $C0_2$ emissions yet many countries are not doing so. World Energy Council data of fossil fuel consumption rates measured against known reserves indicate most countries will run out over the next 100 years.

Solution: $1B R&D Initiative to Create Low-Carbon Society

We propose the United States undergo a $1B Manhattan-like project that brings together 1000 of the brightest engineering minds to develop low cost alternatives to existing infrastructure over a 5 year period.

The MA2 project supports mechanical, electrical, manufacturing and computer engineers. Research is performed by a combination of university professors, PhD candidates, postdoctoral fellows, and top engineers from industry on sabbatical. They produce designs, write software, simulate, create prototypes, test prototypes; and make all materials available to the public on a free gnu public license basis. Also, they create interconnection standards that glue together components in systems

MA2 is exclusively engineering design and *not* scientific research. MA2 engineers work with *existing* technology and are therefore in a position to produce a working product.

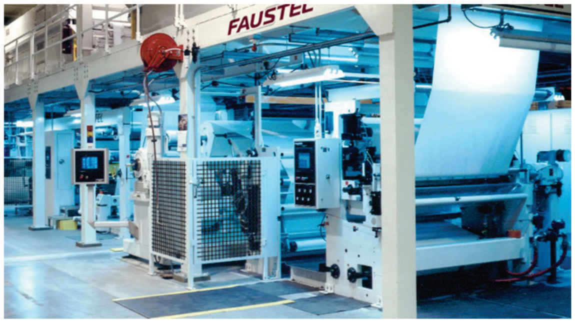
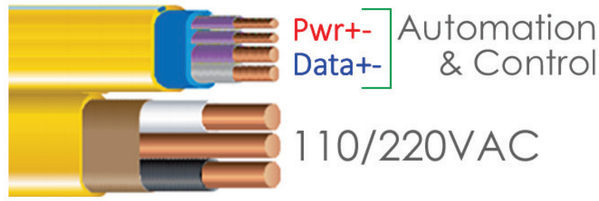
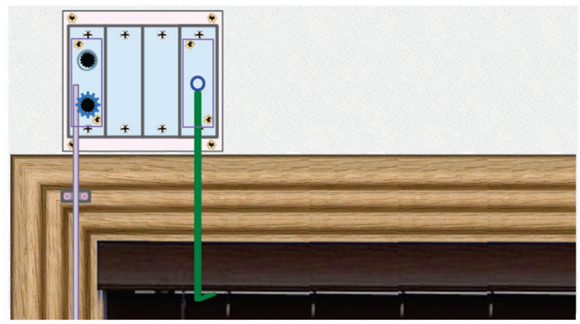
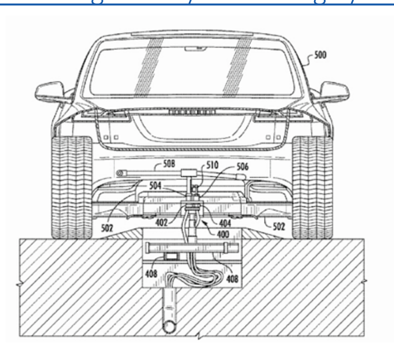
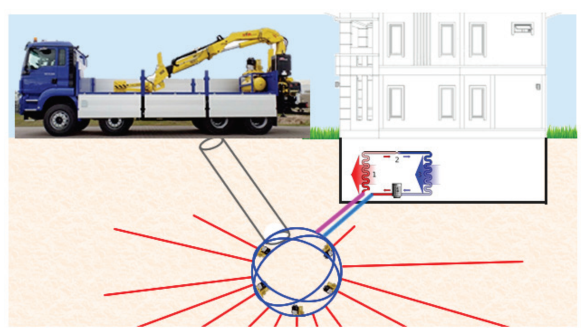
Now Is the Time to Act

This document is a call for action from engineers the world over to harness our comprehensive skills of engineering, invention and implementation to bring revolutionary thinking and solutions to bear on the looming energy crisis and, by doing so, to cause a measurable reduction of greenhouse gases. What other profession can accomplish this? No other.

Our model for the dream team approach was notably assembled by Alfred L. Loomis in the 1930s. His dream team consisted of Albert Einstein, Werner Heisenberg, Niels Bohr, Enrico Fermi, and others. They met at Loomis' state-of-the-art lab in Tuxedo Park, New York. There and at MIT, they conducted secret defense research into high-powered radar detection systems that helped defeat German forces in the air and at sea. Later, with Ernest Lawrence, Loomis pushed FDR to fund research for a second dream team for the Manhattan Project, research that led us into the "Atomic Age".

Today, climate issues coupled with diminishing fossil fuel reserves calls for another dream team, revolutionary thinkers to attack the problem with a combination of theory and invention that could be layered on top of current state-of-the-art, commercial-off-the-shelf technology serving to short cut the timeline to success.

For more details, please visit www.manhattan2.org.

Mass Produce Smart Solar Material  	Extend Network to Windows, Doors, Duct, Ceiling 
Automate Installation Of Solar On Plywood Roof/Walls 	Create Standards that Automate Windows/Doors 
Automate Installation of Solar on Corrugated Steel 	Add Low Cost Vacuum Insulation To Walls 
Create Big Battery Borrowing System 	Reduce HVAC Energy w/ Low Cost Geothermal 

Chapter 2) Manhattan 2 Summary

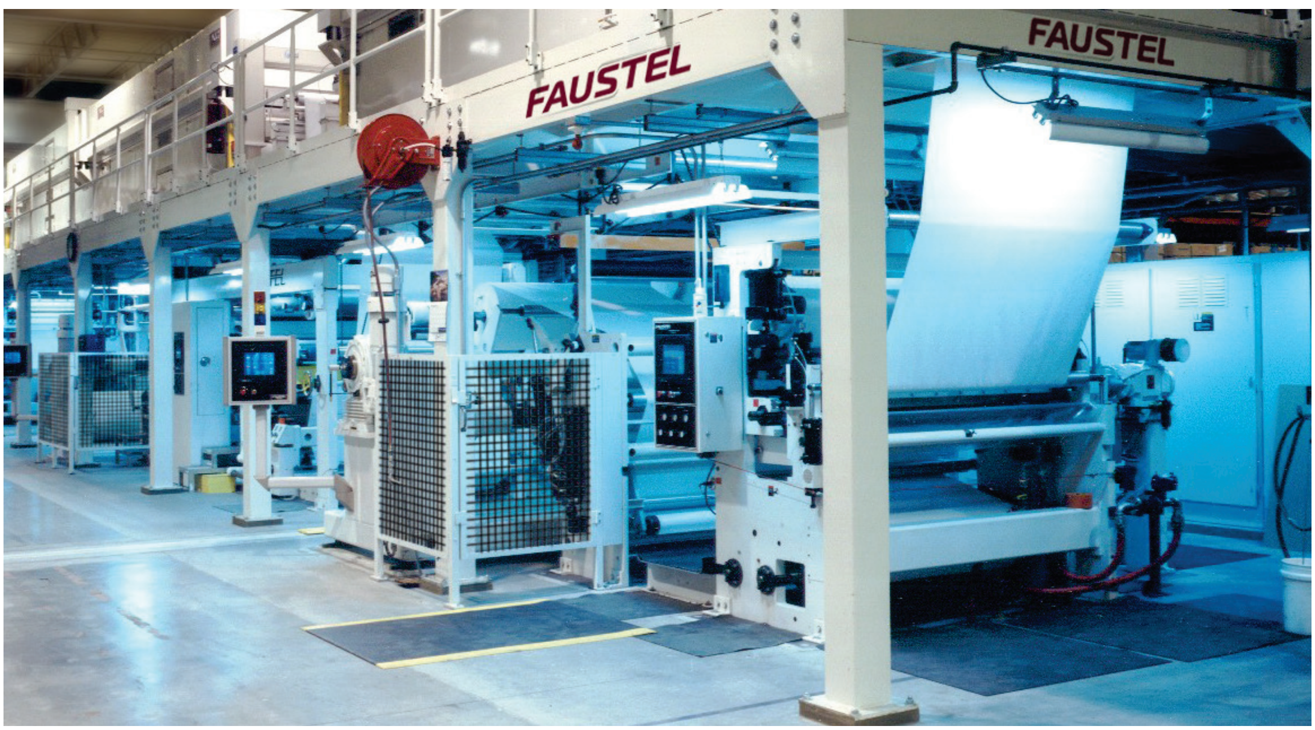
This chapter summarizes primary Manhattan 2 projects.

Mass Produce Smart Solar Material That Includes Switches/Processor

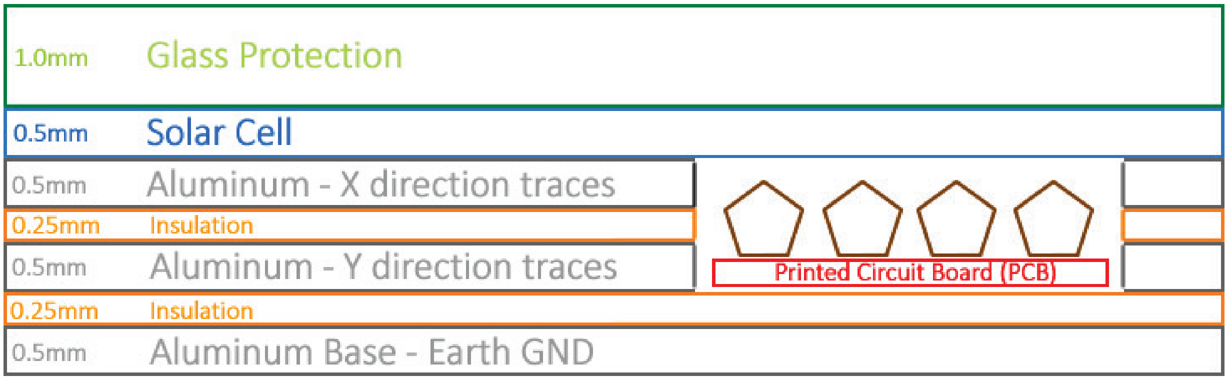
Every building in the world should be completely covered with solar photovoltaic material, from roof edge to roof edge. Manufacturing engineers need to:

> *Develop a continuous factory process that manufactures 1 to 3mm thick solar material that includes solar cells, conductors, switches, processors, and diagnostics. This thin material is later placed directly onto plywood roofs or onto corrugated steel panels.*

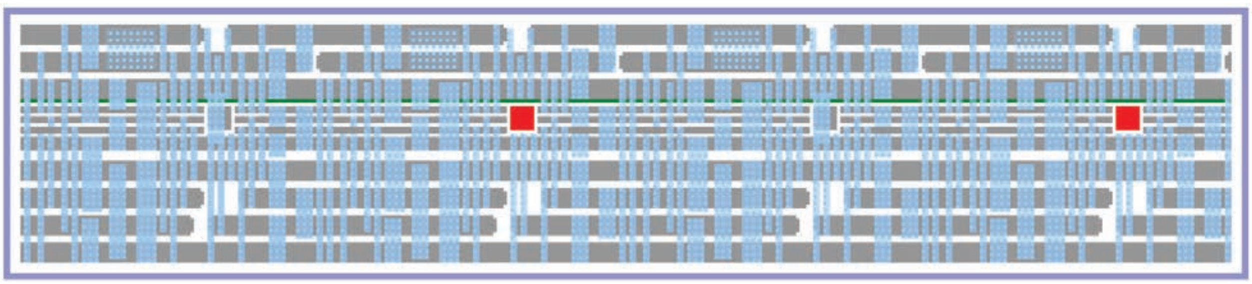
The below photo shows factory machines that laminate together different layers in a continuous process.



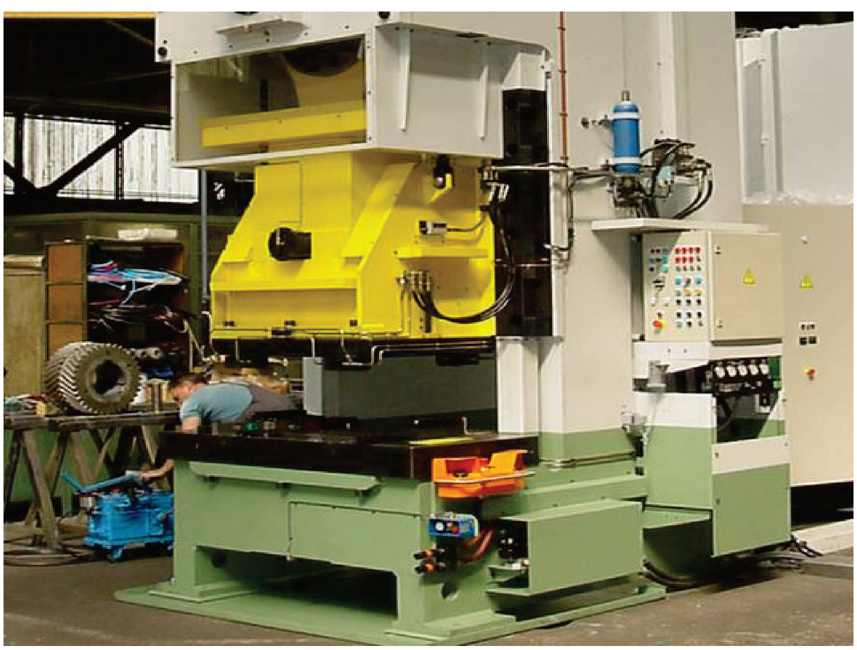
We need a similar process that supports layers consisting of solar cells, conductors, internal printed circuit boards, switches, processors, and metal backing for strength and fire protection. The blue layer shown below is a solar cell that converts sun light to electricity. Total thickness is 1 to 3mm (0.04 to 0.12").



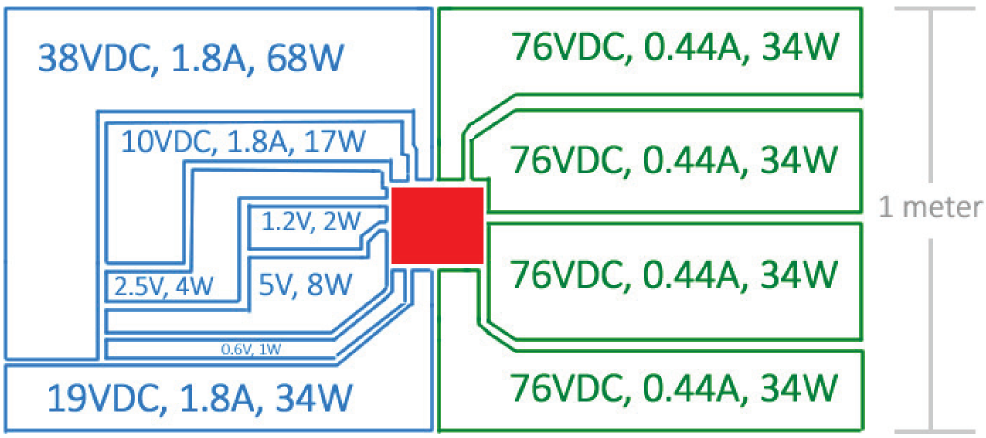
In the below 1x4meter illustration we show one layer for aluminum conductors in the X direction (gray), one layer for conductors in the Y direction (blue), and a control PCB (red).



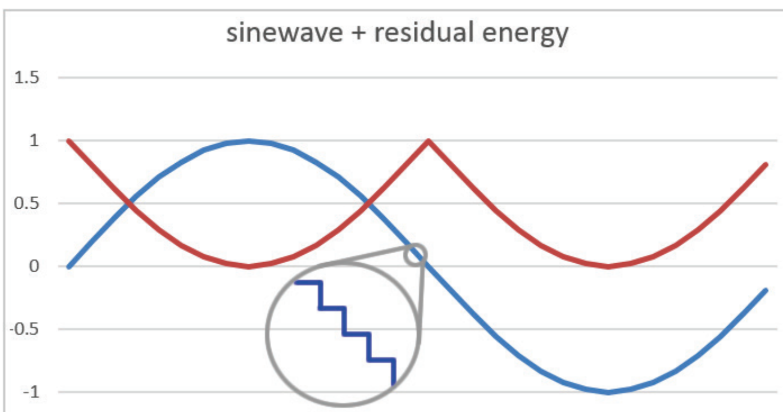
Metal stamping machines create patterns in aluminum at low cost.



Solar cells produce different voltages (e.g. 76V, 38V, 19V, 10V, etc.) that route to internal PCB; as shown in 1x2meter concept below.



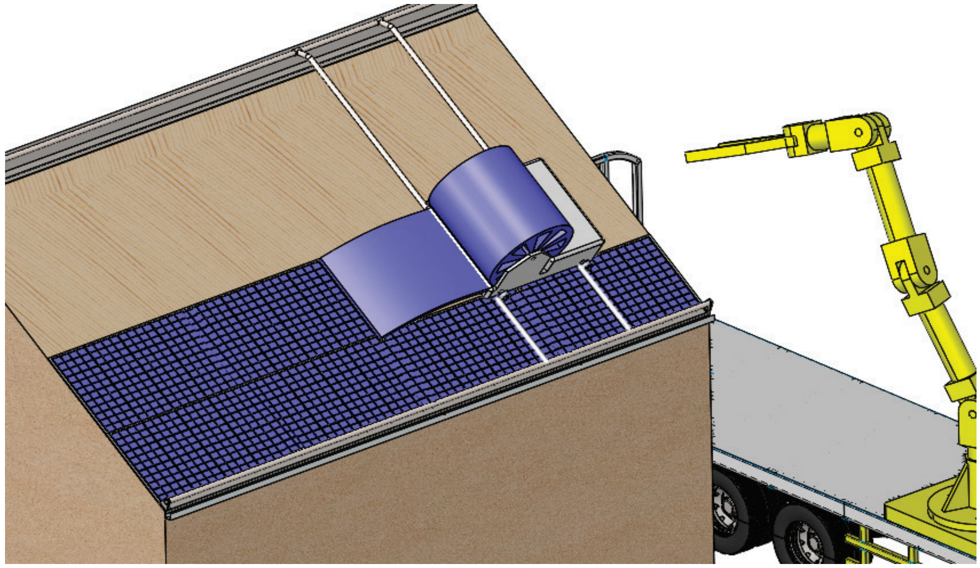
Switches on PCB produce AC and DC voltages at different voltage levels and frequencies, as required by the various components (e.g. ideal power into compressor motor might be 35Hz at 82VAC). The PCB does much of the DC to AC conversion while relying on external filter components for cleanup. For details on smart solar fabric, click here.



Design & Build Factory Production Line That Makes Smart Solar Material

MA2 engineers design and build multiple prototype production lines that fabricate smart solar material. An example of what this might look like is shown to the right. In some cases, machines already exist. In other cases they need to be adapted, programmed, or designed from scratch. MA2 engineers do what it takes to build a production line, create sample material, and accurately calculate cost. MA2 funds prototype production lines at universities interested in automated solar (e.g. CMU, GIT, UICU, UCB, UMA, and MIT).

Automate Installation of Solar Material Direct-to-Plywood Roof



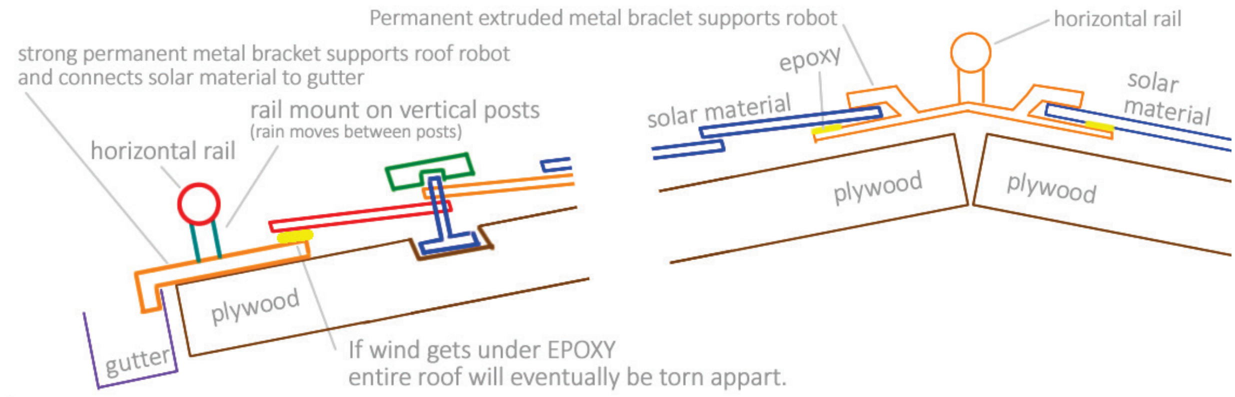
When mounting directly onto plywood a factory fabricates material, cuts strips as needed for each customer, and places them into a large shipping tube. At the building site a truck mounted mechanical arm places the tube onto a roof robot. The robot unrolls the material and places a bead of epoxy at each seam or uses metal molding on 1x1m grid to hold in place. If roof is 10m high by 20m wide for example, one might unroll 11 strips that are each 1m wide and 20m long. We use 11 instead of 10 since we need rain water to flow *across* overlapping horizontal joint.

Shown to the right is an example solar panel cleaning robot which is similar in concept to our proposed robotic system which would support more weight and torque (e.g. wind against spool). Also shown is an analog film canister, which is similar in concept to our solar material, yet smaller in scale.

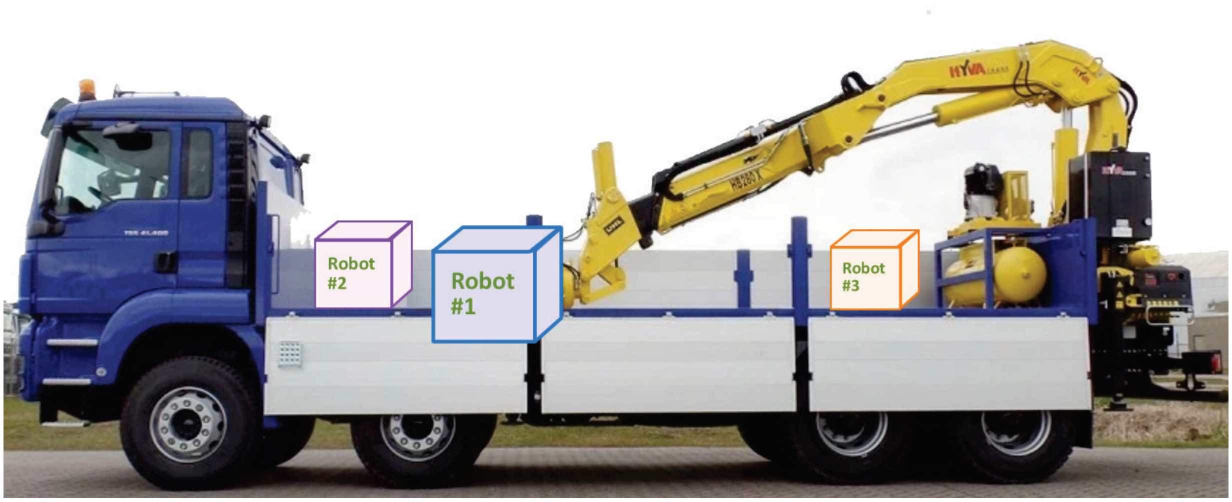
Standardized permanent rails at roof top and roof bottom support a truss that allows robot to move horizontally and vertically for purposes of installation, cleaning, and repair.

The illustration to the right shows solar material mounted on both roof and on wall; and held in place with metal molding on 1x1m grid (shown in gray). Permanent rails that support robot truss are shown in purple.

The below illustration shows how rain water flows across joints and into gutters.



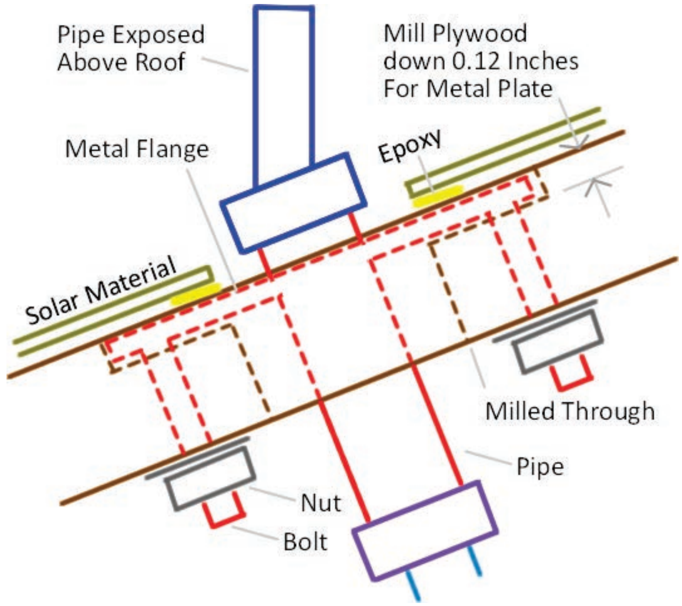
MA2 designs a standard Robotic Construction Vehicle ([RCV](RCV)) that consists of a flatbed truck and a mechanical arm. Robots attach to the end of the arm via a mechanical, electrical and software standard defined by MA2 engineers. Multiple robots are stored on the back of the truck and are accessed in a manner similar to that which is done with numerical controlled milling machine accessing different milling tools.



Different robots focus on a different activity:

- shingle removal
- prepare plywood with holes, milled surfaces, and cut slots
- install custom designed pieces of solar material (e.g. 1m wide strips in spool) and epoxy glue edges.
- repair solar material
- clean solar material, clean gutters, remove leaves

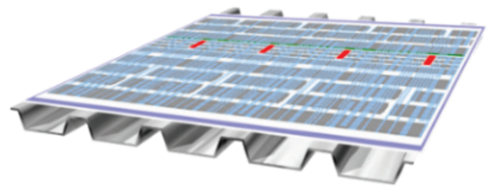
Roof robots install standardized metal ports in plywood that enable pipes, vents and power/control wires to transition from the internal attic to the rooftop. To the right is an example 3cm (1") diameter pipe vent. Plywood (brown) is milled 3mm (.12 inches) for a port metal flange and also milled all-the-way-through for the 1" pipe.



To achieve longevity we work with stable materials on building surfaces. Plastic, rubber, caulking, tar, asphalt, wood and paint are not stable over long periods of time due to ultraviolet decomposition and decay from oxidation and rot. Therefore, glass, stainless steel, anodized aluminum and copper are the materials of choice for long term durability and low life-cycle costs. MA2 engineers explore concepts in simulation that lasts 20, 50, and 100 years; and compares costs. Is a low cost 100 year roof possible?

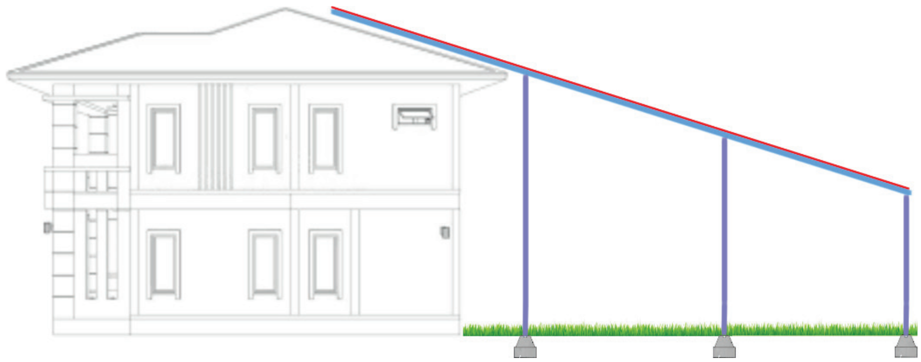
Automate Installation of Solar Material onto Corrugated Steel Panels & Frames

It is a common practice to ship corrugated steel panels on a flatbed truck to a building site and then install onto a metal frame via crane. An example of this is shown below. Also, smart solar material can be factory installed to the outside surface of corrugated steel via adhesive or spot weld. This should be popular with commercial buildings, carports, parking lots, and multifamily residential structures.



The additional cost of adding smart solar material is estimated at $106 per square meter for material above corrugated steel (this does not include cost of electronics below panel). If one sells the electricity from this square meter to the grid, they can pay for this addition in 5 years. Why are these numbers so good? Factory mass production.

Here are several suggestions of what one might do with a low-cost solar on corrugated steel.

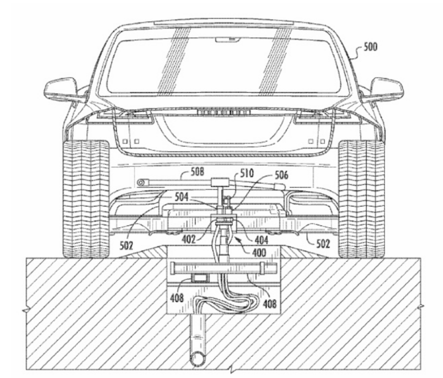


To the right is a 15x12m (40x50ft, 180m^2) 24kW solar array that partially covers the back yard of a two story house. This might look strange at first blush yet 20 years from now fossil fuels will likely be much more expensive. A homeowner with several electric cars might sleep soundly knowing they have a 24kW back yard.

If solar costs are low, relative to revenue from selling electricity, then incentive to build becomes massive. Therefore, it is imperative that MA2 engineers figure out how to mass produce material and automate solar installation. If they are successful, we can expect to see much solar over homeowner yards, patios, parking lots, farm land, and gov't land near highway.

Create Big Battery Borrowing System for Cars and Buildings

Manhattan 2 forms the Big Battery Foundation (BBF) to help coordinate multiple companies involved with large batteries. BBF consists of electrical, mechanical and computer engineers who work near Tesla and Stanford University in Palo Alto, CA. This group consists of experienced engineers on sabbatical from industry (e.g. Tesla, GM, Nissan, BMW, Volkswagen, Renault, Honda, and Toyota), postdoctoral fellows, and PhD students.



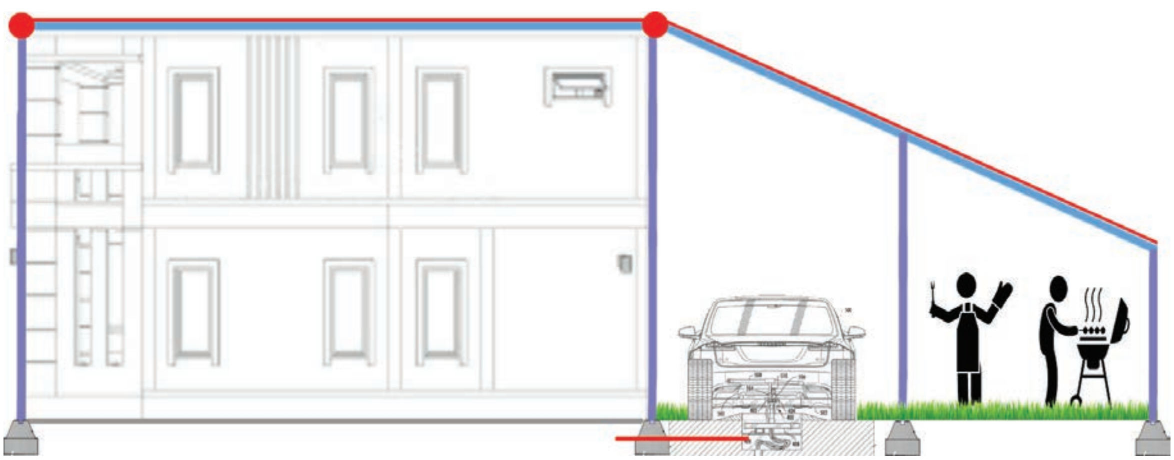
BBF designs a worldwide system to coordinate the borrowing and handling of large batteries. This system supports replacing a depleted battery w/ a fresh battery in less than 1 minute via a mechanism built into cement under car.

When gas runs out (which it will), gas stations will disappear. Instead, car owners will drive into a ~~gas~~ Battery Swap Station (BSS), swap out battery, and be charged for electricity consumed *and* wear on deposited battery.

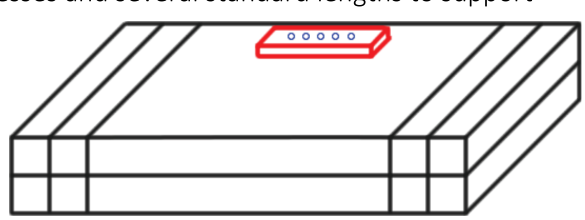
This solves the electric car range anxiety problem, where drivers worry about running out of electricity while on long trips.

This reduces cost since low cost and low range batteries become more feasible.

Homeowners install swap mechanism in their driveway, own multiple batteries, and connect batteries to house power system. This means solar panels on roof can charge batteries during the day, and batteries can power the house when the sun is not shinning. The later point solves the solar energy storage problem (i.e. homeowner has electricity when sun is not shinning). And keeping big batteries out of house keeps Jim happy.



BBF defines several standard widths, several standard thicknesses and several standard lengths to support everything from a large truck to a small hybrid gas car.



Let's assume cars driving on a countryside highway periodically stop at an electric station to swap out depleted batteries for fully charged. Let's assume this is done by a fast automated mechanism and also assume the station charges 1000 batteries each day. Cost of materials above corrugated steel are estimated at $10M. If we devoted 60ft of median strip between lanes to solar, we would need an array 3 miles long for each electric station. Alternatively, one could devote 1000ft x 1000ft of land (20m * 5Km = 60ft x 3miles = 300m x 300m = 100,000 m^2= 25acres).

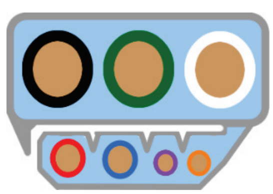
The US Interstate highway system is 48K miles long. If one places stations every 8 miles, for example, we would have a total of 6K stations (48K/8). Cost of material above corrugated steel works out to $60B ($10M*6K). This is reasonable, especially if spent over 10 years. The total interstate highway system cost $500B. The $60B number does not include cost of framing, foundations, and installation. One would need significant automated installation to keep those costs down.

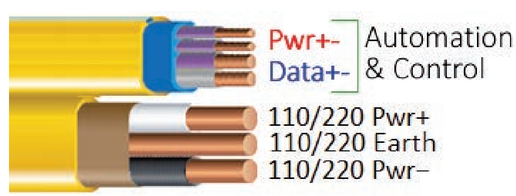
This system works best in regions that receive more sun per day, less snow per year, and have more gov't land nearby. Obvious, many areas of the country are *not* suited for this kind of system, especially urban areas, which is where most cars are driven. Eventually, safer nuclear might be required to support energy intensive urban areas.

Extend Computer Network to Windows, Doors, Ducts & Ceiling

In networking there is a term called "The Last Mile". This refers to how one routes internet data between communications buildings in a city and individual homes. The answer is typically cable modem or fiber. Yet what is more interesting today is how one routes power and data to the last 10 meters -- out to things like doors, windows, occupancy detectors in ceiling, fans in ducts and dampers in ducts.

New 7 Wire Cable (3 power, 4 automation & control)

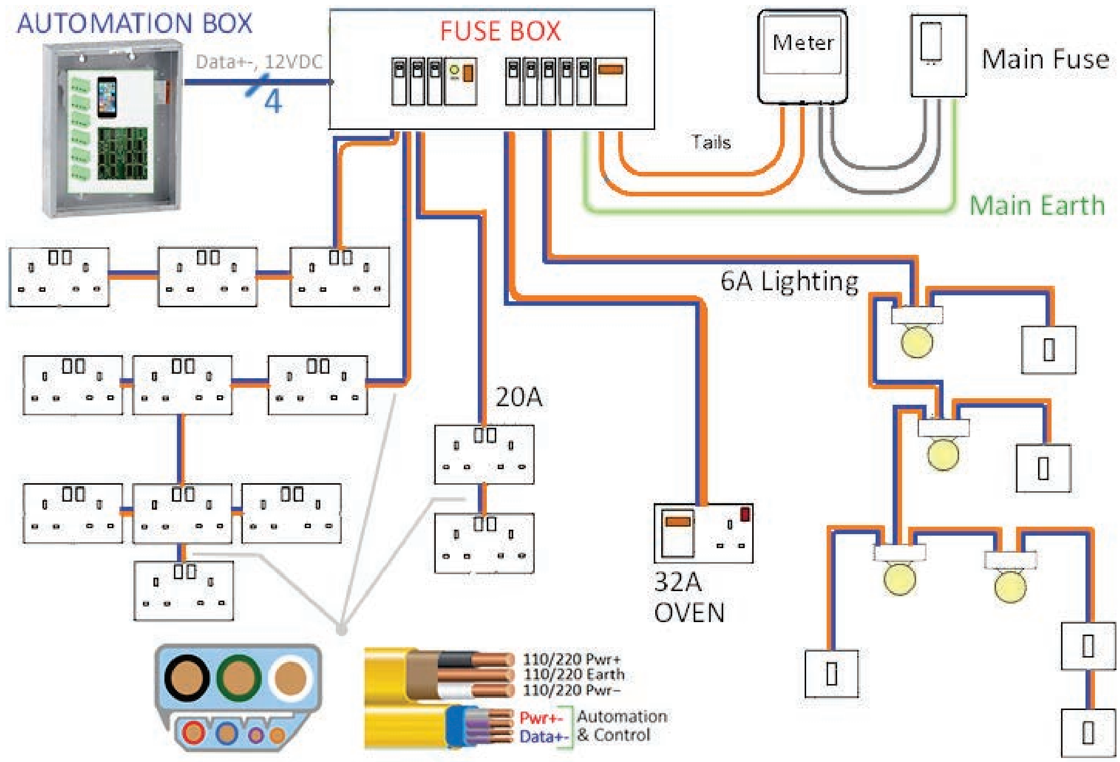




We are already pulling 3-wire power cables from fuse box (e.g. in basement) to every electrical box. Adding 4 more wires to this cable does not involve significant additional cost. Therefore, we can route power and data wires from a central location to every electrical box.

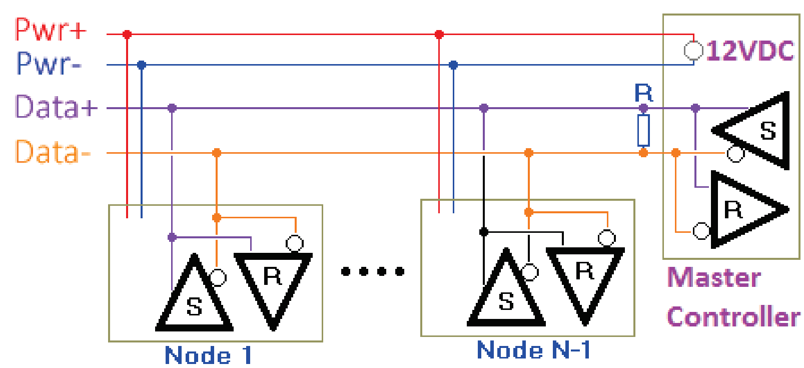
4Wire Bus to Every Door, Window and Ceiling

New 7 Wire Cable (3 power, 4 automation & control) routes to a central location (e.g. house basement) where we have Fuse Box and Automation Box. At this location, electrician separates cable with knife and routes 3 wires to one box and 4 to the other box.



$1 Tiny Nodes on 4Wire Cable

If there are 15 fuses in a house, for example, then one would have 15 of these 4wire networks. Each network (i.e. 4wire cable) can support dozens of nodes since they are wired in parallel. An example node would be a small 1" x 1" PCB with $1 of electronics (e.g. XMC1100, $.58) that connects the 4wire bus to motors, switches or sensors as described here.



Thermally Shut Down Rooms Not In Use

This bus allows us to gain thermal control over every room by closing and opening doors, and adjusting fans and dampers inside ducts. Rooms are shut down thermally if not in use as described in here.

The Building Brain

The Automation Box contains the building "Brain". The Brain absorbs information and makes control decisions every 100mSec. Its primary goal is to reduce energy consumption while maintaining comfort.

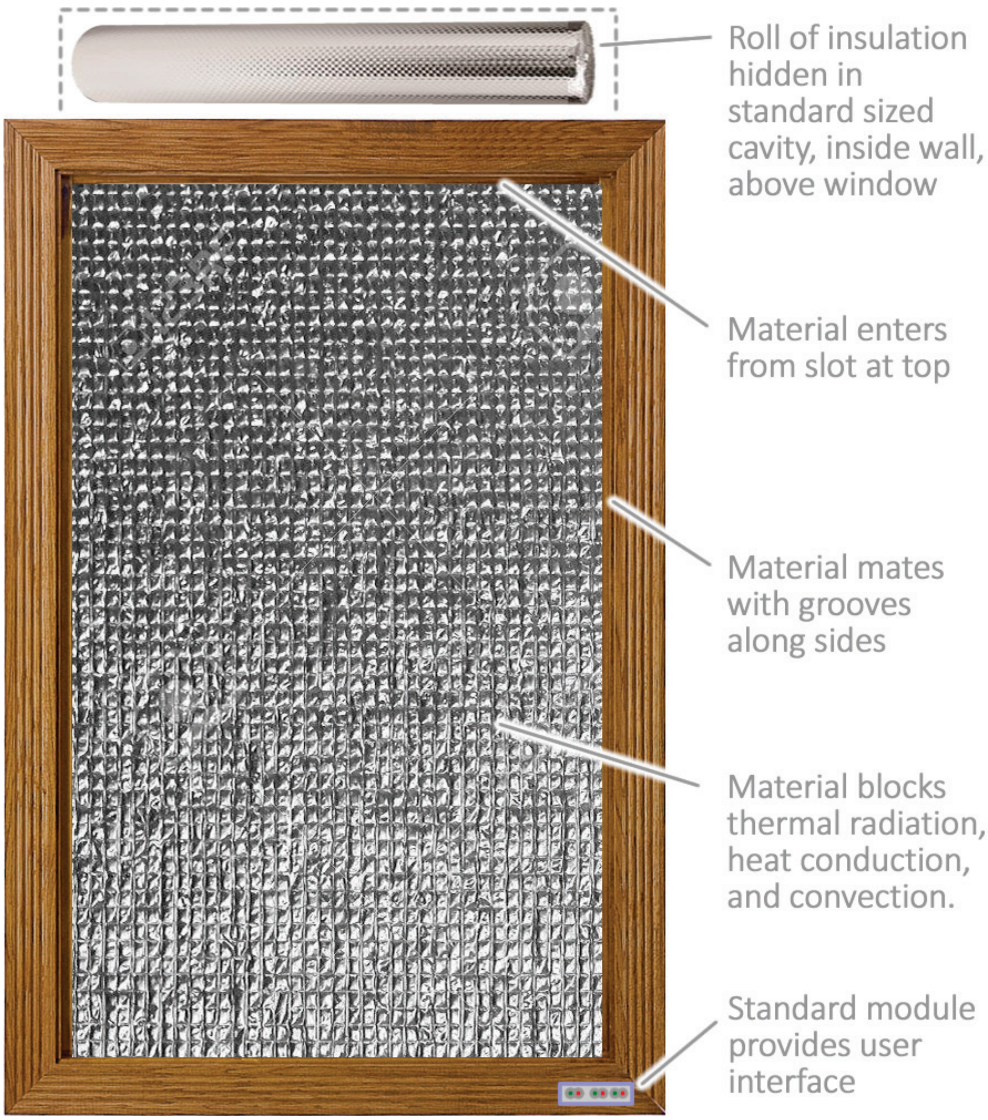
Stop Ringing With Snail Signaling

Notice in the above home power wiring diagram we see nodes configured more like Tree branches than nodes along a line (e.g. daisy-chain). With a Tree, there is no way to control impedances with termination. If one drives a Tree with a fast rise time (e.g. 15nSec), then reflections will cause the signal to ring, which leads to bit errors. If one slows down rise time to 1/30th of the time for the electricity to get from one end of the network to the other, then this goes away. In other words, if your rise time is 15uSec instead of 15nSec (1000 times slower), you get a reliable and very low cost connection. Snail slow is ok with our application since it involves sending very simple commands to tiny nodes (e.g. "open cloth curtain").

Create Standards that Automate Windows and Doors

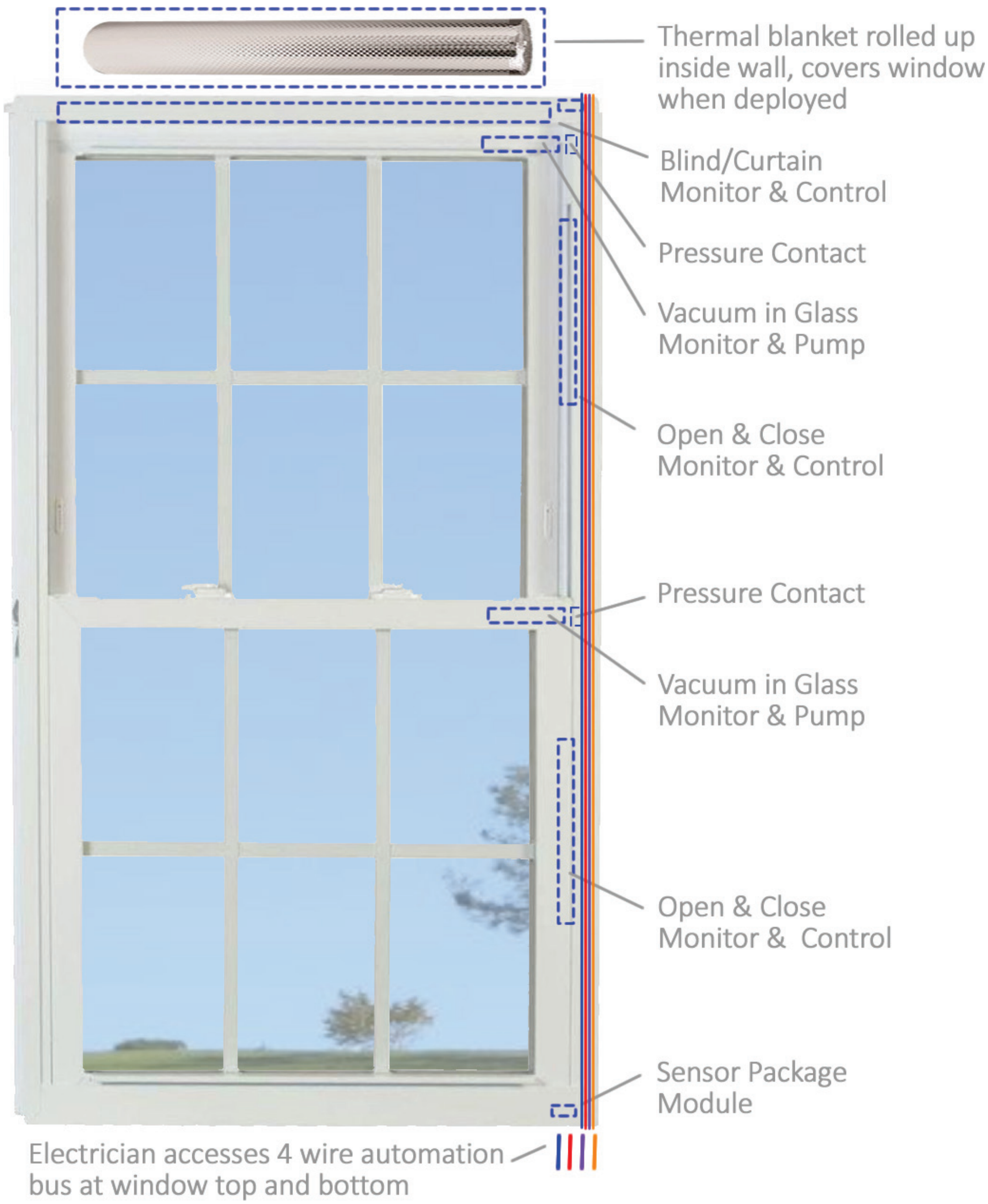
Every Window on the Planet should have an Automated Thermal Blanket

When ceiling occupancy sensor detects a room is not in use, a thermal blanket, deploys to significantly reduce energy loss. These roll up in the wall above each window and mate with grooves at each side. This significantly reflects sun light, reflects thermal radiation, reduces airflow and reduces heat conduction.

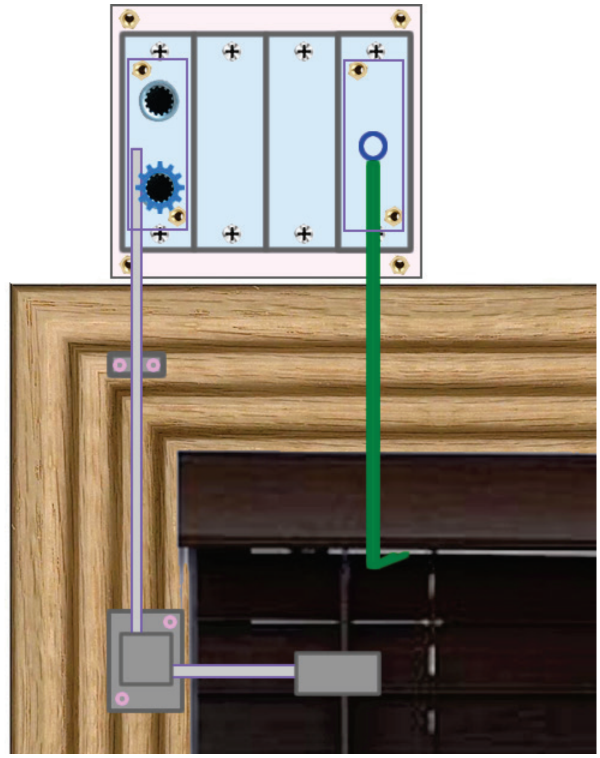


Create Standard for Modules and Cavities in Windows

Windows are manufactured with standard sized internal cavities that are covered with small hatches. The 4wire bus routes to each cavity; subsequently, one can install standard modules to perform various functions such as open/close window or deploy thermal blanket. Also supported are small low-cost vacuum pumps that maintain a vacuum in-between two panes of glass.



Create Modular Chassis Standard above Window, above Door and in Ceiling



Obviously, one does not have the luxury of internal cavities built into a window when working with existing construction. Subsequently, MA2 creates standard chassis system that is installed into wall above window, into wall above door, and into ceiling. For example, to control all windows, one might place a chassis above each window, possibly installed by an electrician, as illustrated to the right.

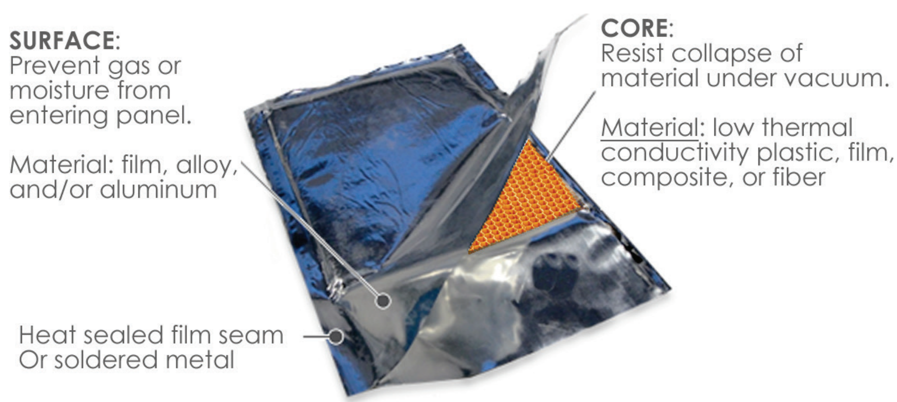
Standard Wall Bus modules plug into chassis to support things like occupancy detection, fire detection, Wi-Fi range extension, curtain/blind motor control, door open/close control, and Alexa/Google Home microphone/speaker (via Wi-Fi).

The motor control module, pictured to the right, provides a standard rotating gear that mates with a variety of linkages that interface to existing curtains and blinds. A market for different linkages enables one to connect the standard gear to many different existing window coverings.

We work with doors in a similar manner as windows since the building Brain needs to open and close doors when thermally controlling each room.

Add Low Cost Vacuum Insulation to Walls

MA2 forms the Advanced Wall Group (AWG) to study advanced techniques for improving wall insulation. AWG consists of mechanical, manufacturing and computer engineers who focus on adding vacuum insulation to walls. This group consists of experienced engineers on sabbatical from industry, postdoctoral fellows, and PhD students.

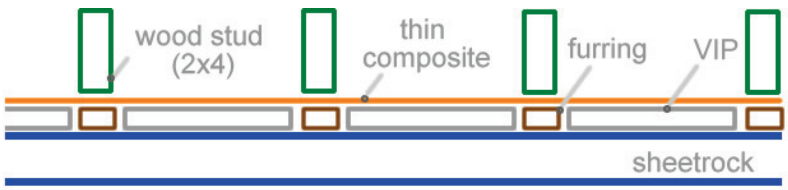


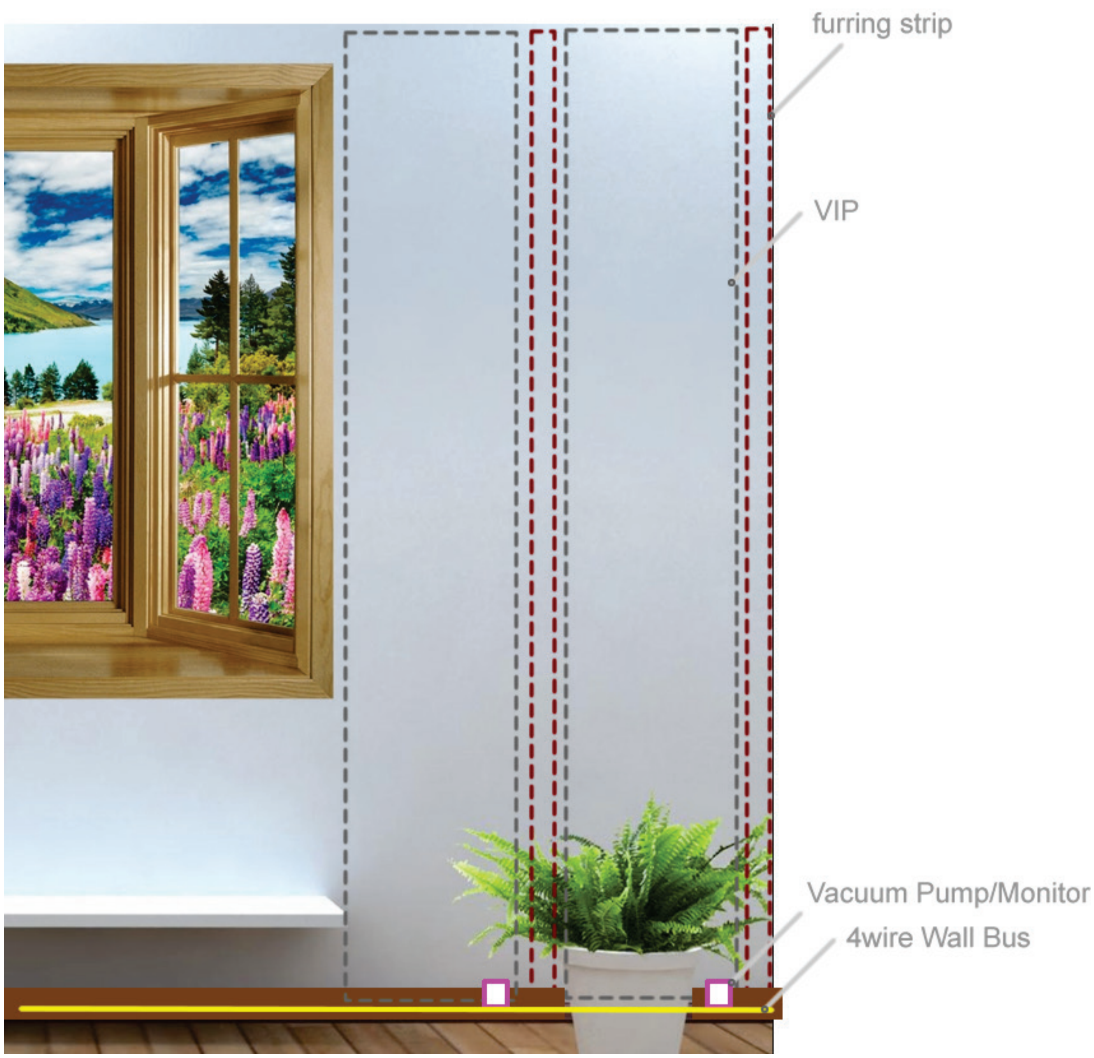
A different MA2 group, called "Vacuum Insulation Group" (VIG), creates a factory production line to mass produce low cost vacuum insulation panels (VIP) and low cost pumps that are connected to Wall Bus. The Advanced Wall Group (AWG) focuses on installing these products into walls.

Typical panel is heavy aluminum foil w/ internal light honey comb material to keep panel from collapsing when under vacuum pressure.

Add Low Cost Vacuum Insulation to Walls

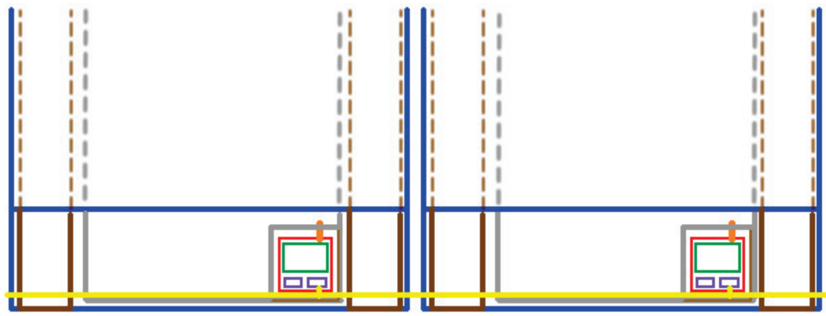
What can we do with Wall Bus everywhere and small $3 vacuum pumps? AWG engineers study multiple concepts, including the one shown below which places 14.0" x 0.3" vacuum insulation panels (VIP) at 16" intervals directly behind sheetrock.

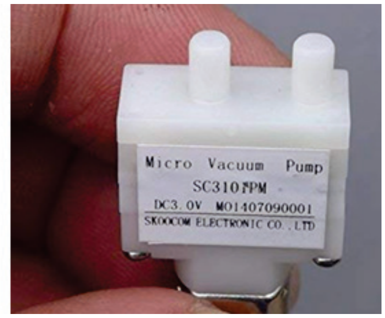




Flexible vacuum insulated panels (VIP) slide into cavities from a base location, under the lower (or upper) horizontal molding. Molding is held in place with screws; therefore one can easily install, maintain, and replace panels, pumps and wiring. Evacuated VIP panel (gray in above illustration) can be thin (e.g. 0.25mm to 4mm) with small internal insulated spacers on ~1x1cm grid to reduce collapsing. VIP wall might be a heavy aluminum foil, possibly coated with a plastic film. Furring strips (brown) are made of a strong thermal insulator (e.g. plastic honey comb w/ internal air pockets). Small vacuum pumps (green) are connected together with Wall Bus 4 wire cable (yellow). A thin 1 to 3mm composite material (orange) protects VIP back surface.

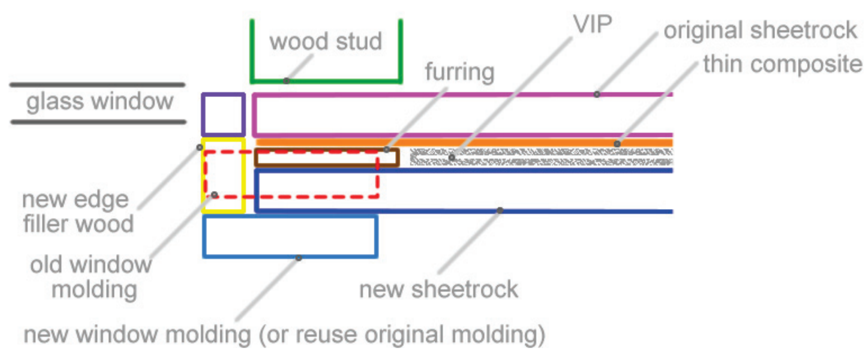
The below illustration shows the bottom of two vacuum insulation panels after removing the lower horizontal molding. The VIP's (gray) are attached to small vacuum pumps (green), which are connected together with a 4wire Wall Bus cable (yellow). The existing pump product shown to the right is similar to our desired pump, yet only pumps to 0.3ATM. The VIG group invests considerable resources to designing tiny vacuum pumps that meet our needs. We have plenty of time to pump, therefore flow rates can be low.





Add Vacuum Insulation to Existing Construction

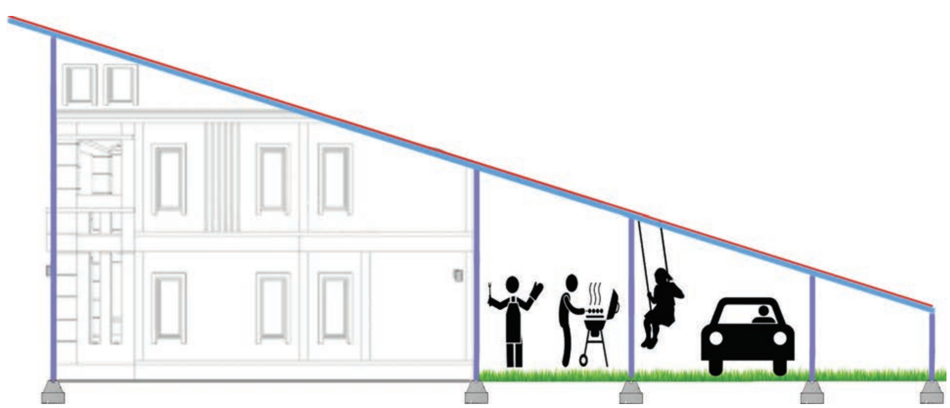
What can we do with existing construction and how much does it cost? AWG engineers study multiple concepts, including the one shown below. In this scheme, molding around window (dotted red) is removed, new sheetrock + VIP + thin composite wall is added, edge treatment molding (yellow) is added, and new molding (light blue) is placed around window (or reuse original).



For more details on how this works, click here.

Zero Energy (ZE) Architectural PhD Program

MA2 encourages architectural schools to set up a PhD discipline called "Zero Energy" (ZE). This entails designing and prototyping buildings that use (close to) zero energy; with less emphasis on aesthetics. Pumping energy into the grid and then pulling it out later is allowed. ZE students utilize PV solar, geothermal heat pump, tanks of water for energy storage, vacuum (VIP) insulation, and thermal control over doors/windows/rooms. Manhattan 2 encourages at least one school in a hot climate (e.g. Rice University, UT @ Austin) to set up a graduate ZE program to test ZE concepts in hot climates. An example hot ZE concept, illustrated above, would be a 200 square meter (e.g. 13x16m, 40x50ft) 28kW solar



array made of corrugated steel panels on aluminum framing. This replaces traditional wood roof and sits above house, cars, and patio. Can we do this at less cost than traditional wood roof via automation? If so, this would become very popular. For details, click here.


Fully Zone Each Room

Normally a building does not fully zone each room due to the high labor cost of running power wires from duct dampers and fans to a central location. The fully deployed 4wire bus allows us to easily support motorized duct accessories at more locations.


Reduce Heating/Cooling Energy Consumption w/ Low Cost Geothermal Heat Pump

The temperature 4m (13ft) underground is typically ~12°C (55°F) throughout the year. If one embeds plastic pipe and runs water though it, they can deliver this 55°F to a heat pump that supplies heating and cooling to a building. This is called "shallow geothermal heat pump" and is a common technique that reduces energy consumption 25% to 50% for *both* heating and cooling.

The only problem is the $16K cost to install the underground pipe for a typical home.

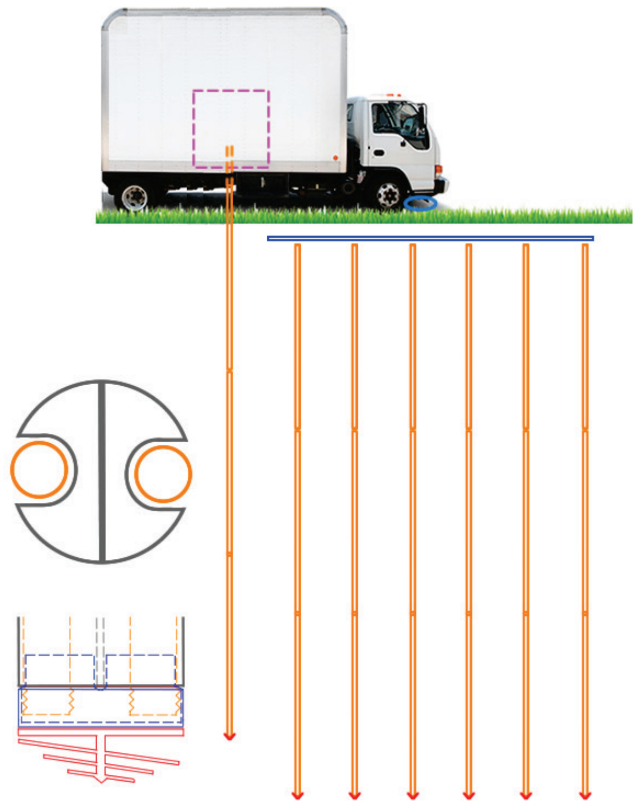
Subsequently, MA2 engineers explore multiple concepts for automated machines that install pipe at low cost. For each concept, MA2 engineers create drawings, simulate, and produce cost models. If a design is feasible, a prototype is created and tested. The following pages show several different concepts that would be evaluated by MA2 engineers. For more details, click here.

Concept #1: Automated $4k Vertical Boring Machine

Currently, we have expensive services that drill two ~50m (150ft) deep 1ft diameter vertical bore holes to provide chilled water to a house. Can we create a machine that drills 4 to 10 holes that are 50 to 120 ft deep in several hours, with one operator, at 4 to 6 times less cost? One method of installing 3" to 4" diameter HDPE plastic pipe is to place removable steel pipes to left and right while drilling, as shown in orange. This helps in the following ways:

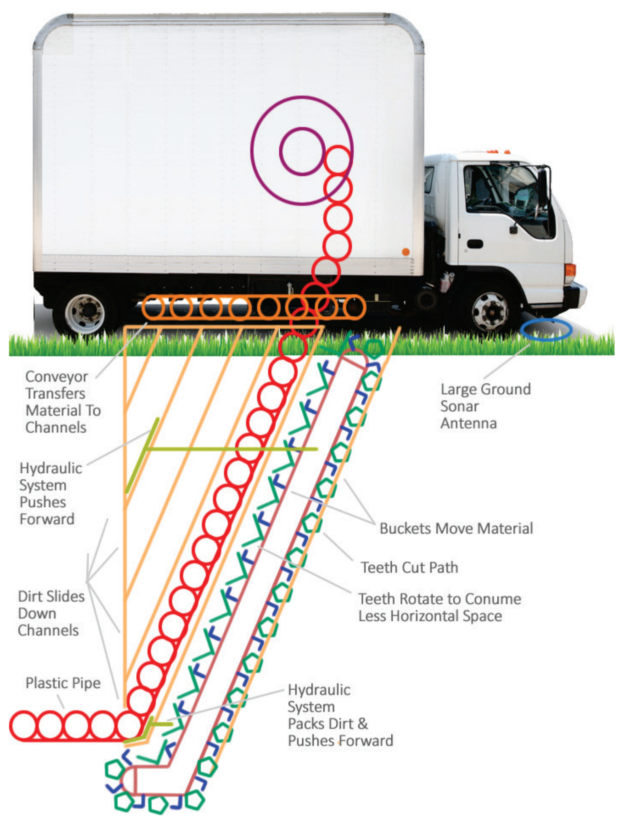
- Guide HDPE pipe in a straight line.
- Move soil between head and surface via water.
- Power rotating drill head via water in metal pipes.
- Deliver grout to hole as pipes are withdrawn.

For details, click here.



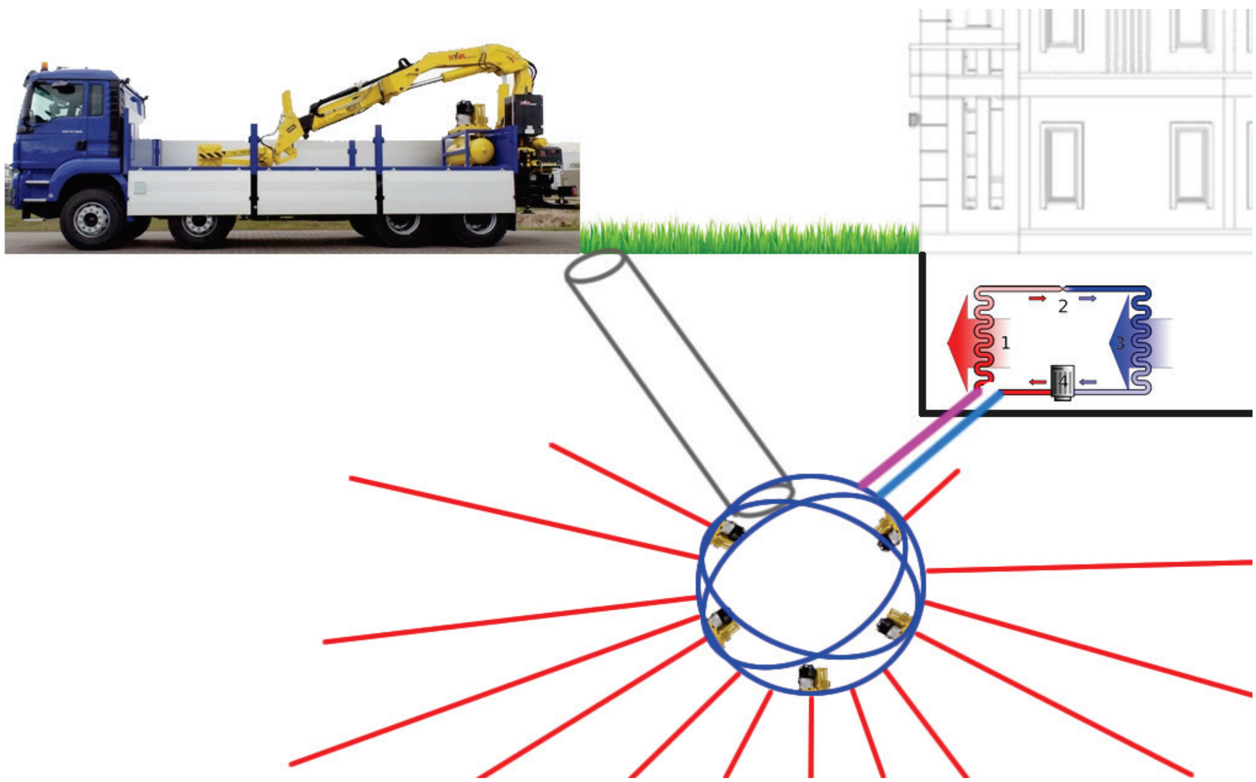
Concept #2: Advanced Chain Trenching Machine

The illustration at the right shows an automated machine that creates a 0.3m (1ft) wide trench that is 4m (13ft) deep. The chain system cuts a path, moves soil to the top, places it onto a conveyor, and then returns soil back into the trench. Coiled 2" diameter plastic pipe, shown in red, is inserted into trench and tightly packed using hydraulics. Machine travels on highway with center of mass low to the ground, unfolds, submits pipe to hole in basement wall, and cuts a trench. For details, click here.



Concept #3: Underground Robotic Gimbal Chamber

MA2 Engineers explore creating a permanent chamber 6m (20ft) underground that is accessible via a 0.3m (12") diameter permanent pipe. After installing access pipe with auger, machine digs out chamber with high pressure water. Gimbal is constructed inside chamber to avoid cave-in and to support multiple drilling robots that affix to gimbal. Robots drill holes simultaneously for water pipes that emanated from chamber. Pipes are connected together and routed to house. This approach might work well in urban areas where truck on street accesses soil via flower bed at front of building. For details, click here.



Create Universal Language for Building Devices to Communicate

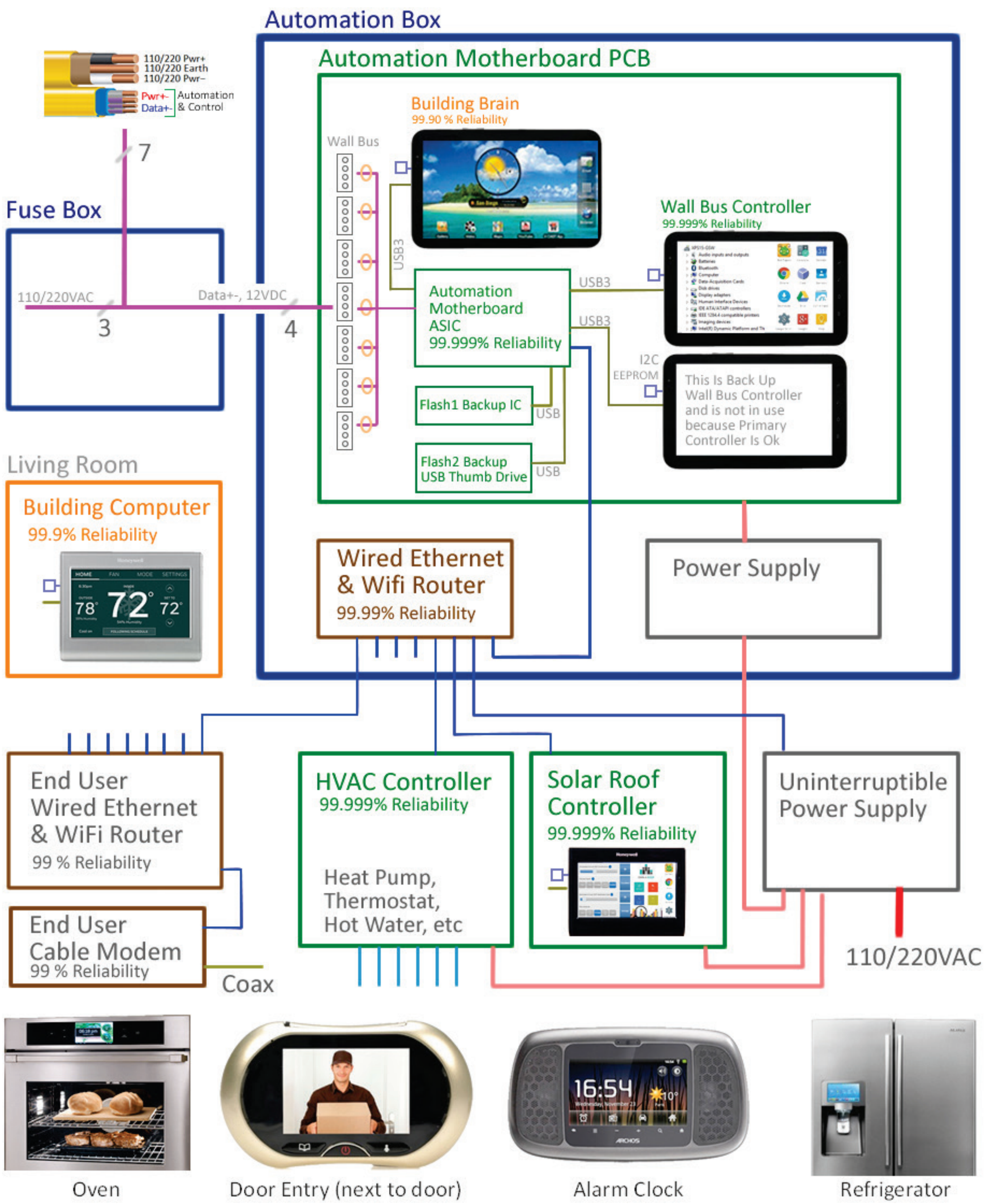
Currently, a common universal language for building devices to communicate does not exist. Subsequently, Manhattan 2 forms the Data Interoperability Foundation (DIF). This organization creates data models for many common devices and defines a method by which they can communicate.

Electronically Connect Architect, City, Suppliers, and Robots

MA2 forms the Architectural Data Foundation (ADF) to electronically coordinate architects, cities (who hold data files), pre-fabrication factories, general contractors, and robots. This group consists of computer scientists, manufacturing engineers, robotics engineers and a few architects. Less than 10% of this group are architects since the focus is software development. Their headquarters is placed near a school strong in these areas and all work is made available to the public on a free gnu public license basis. ADF consists of industry engineers on sabbatical, postdoctoral fellows, and PhD students.

Standardize Building System Design

Devices inside a building do not coordinate well due to industries and companies who focus on their own needs. Devices from Amazon, Apple, Samsung, Microsoft and Google often don't talk and HVAC controllers are typically proprietary. This all begs the question, "How can we unify the various building components to save cost, increase convenience, and reduce energy consumption?" MA2 engineers explore multiple concepts, one of which is illustrated below for a single family home. For details on this works, click here.



Physically Embed Android-like Devices Everywhere

A mechanical standard is created for android tablet devices that enables them to fit into a standard sized socket, inside other products. Subsequently, low cost devices with great capability easily connect to the rest of the building. Embedded Android tablets are shown below in dotted red. For details, click here.

Why Has Our R&D Projects Not Been Done Previously?

Many of the proposed projects involve *components* that interconnect with other components to build a *system*. And currently there are no standards that define these interconnections. A company will not develop a component if it cannot connect and be used. This is a classic chicken and egg problem. The MA2 initiative resolves this by developing multiple components in parallel and MA2 engineers insure they interconnect in a working system. Ultimately, prototyped components evolve into well-defined standards and then Industry can build. In effect MA2 spawns industries for low carbon technologies that involve somewhat complex systems.

An example of this is AT&T Bell Laboratories. This organization developed technologies and components used by the US wired telephone system. They invented things like the transistor, laser, and solar cell; in addition to many other important technologies. These were then used to design the international communication system.

We now need an international low-carbon *energy* system. Our situation is much easier since we can utilize existing technologies; whereas AT&T relied on scientific inventions.

Conclusion

By necessity, Manhattan 2 is a not-for-profit endeavor. As such, software, designs, inventions, and mechanical plans are given away for free on a gnu license basis. The goal of Manhattan 2 engineers is design, simulate, write software, build prototypes and publish their designs to launch forward thinking entrepreneurs and established industries alike toward achieving the promise of non-fossil fuel energy technologies.

Engineering projects are initiated only if they utilize existing technology and lead to products that are low cost, reliable, and stable over time.

Top electrical, mechanical, manufacturing and computer engineering minds are identified, given money, and given control.

Manhattan 2 is funded with private or gov't money.

If you are a billionaire and find Manhattan 2 interesting, please phone the university president of your choice and ask him or her what they think about the concepts discussed in this document.

Now is the time for Manhattan 2.

Chapter 3) Mass Produce Smart Solar Material That Includes Switches/Processor

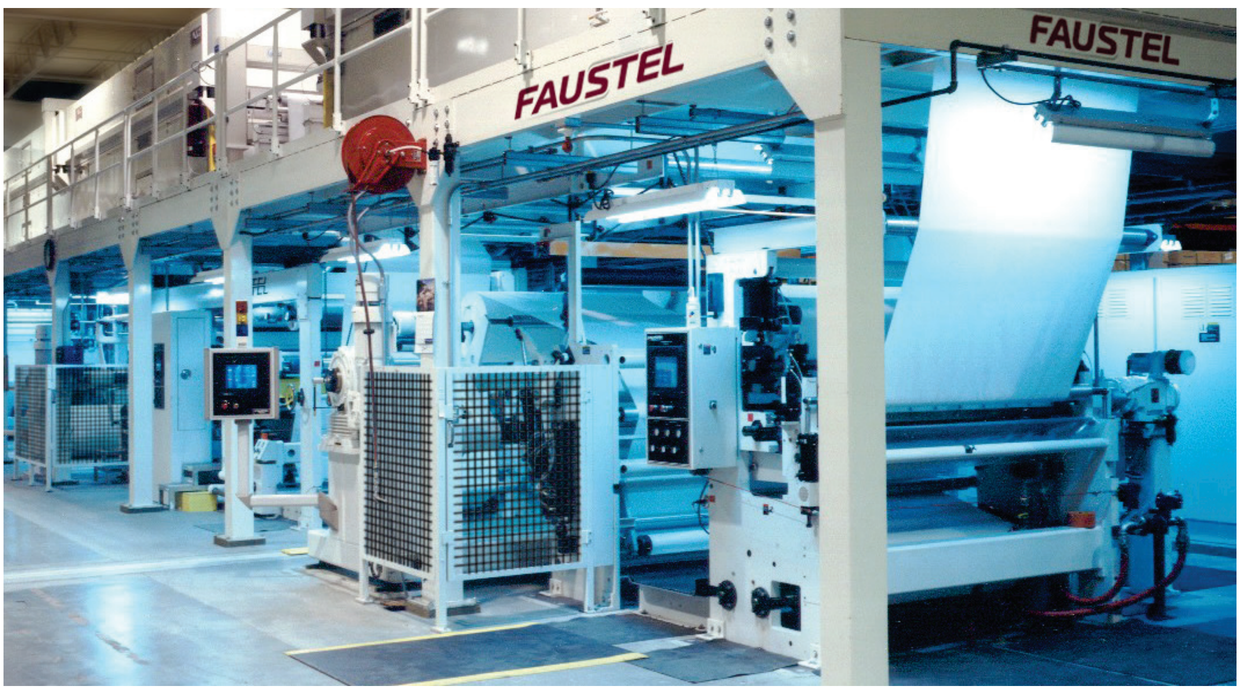
Imagine a world where one can fully cover a roof with solar material, and this material is also the water barrier. Now imagine this cost less than a traditional roof with shingles. What happens next in this theoretical world? Only one thing happens. Massive adoption.

Is this possible? Today, a worker installs each shingle manually, one at a time. This is costly. Also, this is great because it makes it easier for solar to beat shingle.

This document talks about how Manhattan 2 engineers can create a solar industry that beats shingle.

Continuous Lamination Process

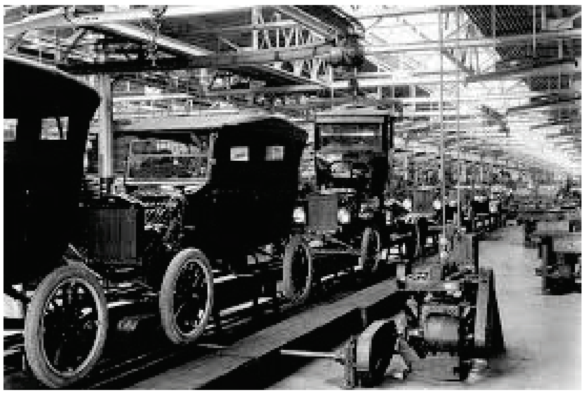
The below photograph shows a production line that forms material in a continuous process. Companies throughout the world manufacture machines that do this, an example of which is Hymmen. One can create a complicated laminated material using heat, compression, adhesives, and stamping. A spool of aluminum is available in different thicknesses and widths for approximately $2.2 per Kg ($1 per LBs).



Solar Mass Production

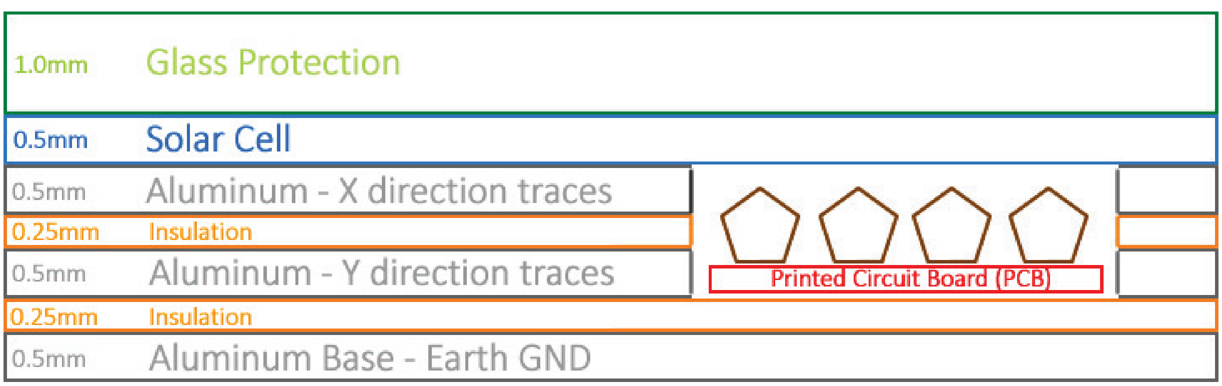
Henry Ford revolutionized transportation with mass production. We need to do the same with solar. This has not been done previously due to a lack of standards that define how components interconnect. MA2 needs to create these in order to propel solar forward.

The solar industry needs to think more like Henry Ford and less like Rolls Royce.



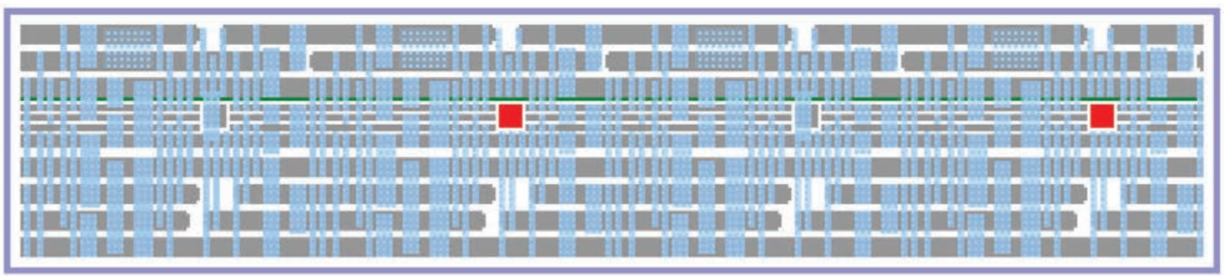
Solar Lamination

Solar material produced in a continuous process could provide protection (glass top), solar cells, conductors to transfer electricity, MOSFET/transistor switches, and rigidity (aluminum base). A layer of aluminum carries current in the physical X direction and another layer of aluminum carries current in the Y direction. The bottom metal layer provides rigidity and insulates the underlying surface from electricity. Below is an example approach that has a total thickness of 3.5mm (0.14 in).



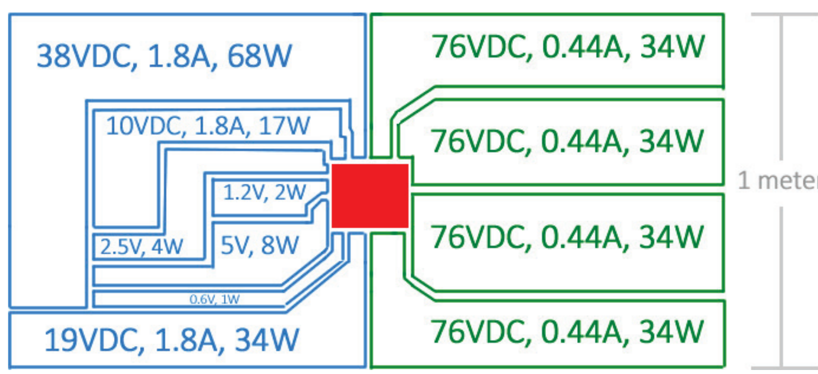
Built In Conductors & Switches

The below 1 x 4 meter concept shows the X conductor layer in gray, the Y conductor layer in blue, and a MOSFET/Transistor Control PCB in red. This is similar to the internal layers in a Printed Circuit Board, yet we use stamped aluminum that is 15 times thicker than typical PCB copper. Our cost is much less due to a low cost stamping process. Our mechanical accuracy is lower, yet acceptable. Connectors attach to the bottom of the PCB, or to metal tabs accessible from the bottom. The green colored line near the PCB carries several control/power wires. Layers are interconnected via spot welded tabs. This illustration is not a working electrical circuit, merely a conceptual illustration.



Typical 110VAC 1x2meter Bank

The illustration to the right shows a typical solar cell layer that produces a 110VAC sinewave at 0.88Amps and a residual waveform (1-|sine|) in 1x2meters (3x6ft, 268Watts, 21% efficiency solar cells, sun at a 45° angle). The right half provides four "batteries" that each produce 76V at 0.44Amps (.88Amps when in parallel). These are controlled with high voltage TRIAC switches. The cells on the left are controlled with more efficient lower voltage MOSFETS. These provide 1.3Amps at 38V, 19V, 10V, 5V, 2.5V, 1.2V, and 0.6V. This 1x2meter bank can produce an AC or DC voltage at any voltage level between 0.6V and 110VAC (or 156VDC), and at any frequency between 1 and ~500Hz. Matching the type of electricity to each requirement allows us to gain great efficiencies and benefits, which we will discuss later in detail.



The PCB, shown in red, controls all switches, monitors all voltages, and communicates with a central controller. Several different standard PCB's are available and are selected to match solar cells and requirements.

Current Solar Systems are Deficient

Today's residential solar panels are inefficient and costly for several reasons:

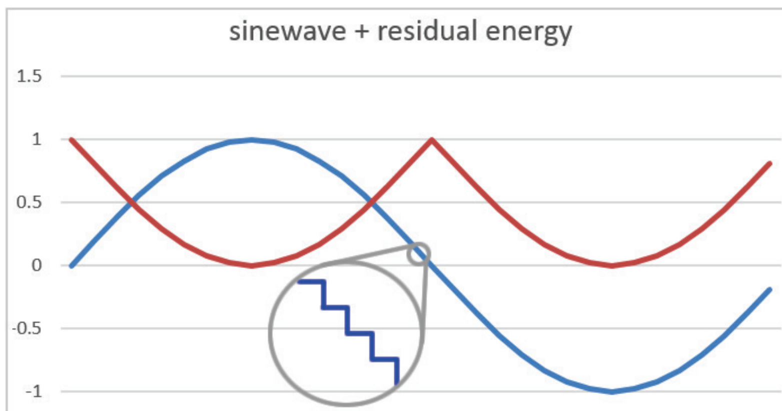
- covers shingles and is therefore expensive when one needs new shingles
- does not make use of rigidity of underlying plywood and instead duplicates that effort in a frame
- no mechanical, electrical, signaling standards support solar
- installation involves much manual labor and is therefore expensive

- system does not create the ideal AC or DC voltages needed by the various components (i.e. heat pump, end user 110/220VAC), due to a lack of standards
- outputting fixed 24V to DC to AC converter is inefficient compared to multiple solar cell voltages and switches creating pure sinewaves

Many Switches can do Great Things

Solar cells typically produce ~0.6Volts yet can be combined in different arrangements with MOSFET and TRIAC switches to produce more voltage. These switches can be embedded into solar material and are extremely helpful since they can bypass faulty cells and they can create AC or DC electricity at different voltage levels. Having a computer control individual solar cells with thousands of switches is terrific since it allows one to create pure sinewave AC directly and efficiently, and feed large machines such as heaters and air conditioners with a variable AC or DC voltage, and variable frequency, for each machine. Why should an HVAC system turn on and then off on a 45minute cycle making a room a little too cold and then a little too hot? Many energy consumers would prefer a variable energy source, not fixed 110/220VAC.

Build Beautiful Pure Sinewaves



One can turn switches on and off to produce a sine wave, as shown below in blue. When one does this, they also produce a residual waveform, shown in red. The residual can be fed into a heating element, switching power supply, or motor that accepts this type of waveform. In many cases further filtering improves waveform post switching. Current DC-to-AC converter products put stress on capacitors and fan bearings; and this is alleviated if solar the material produces a waveform close to the desired shape.

Factory Builds to Order

The continuous factory fabrication process does *not* make the same pattern over and over again. Instead it is under computer control to assemble material as needed. Here are several examples:

a) Parking lot for 150 cars needs 300 pieces that are 1m x 7m mounted on corrugated steel panels. Each piece has 0.5m of dead zone at each end (no solar cells, no conductors). Each piece consists of three 1x2m banks similar to above, yet with 60VDC (480VDC/8) batteries to better match automotive needs.

b) House with 110VAC power, no heat pump, no geothermal water, warm climate (e.g. Texas), and most energy goes to traditional air conditioning compressor. The 1x1m panels focus on creating 110VAC power and sending residual energy from making sinewave (one third energy) to air conditioning compressor motor. Also, chimney and pipes in roof are avoided with dead zone between several 1x1m panels.

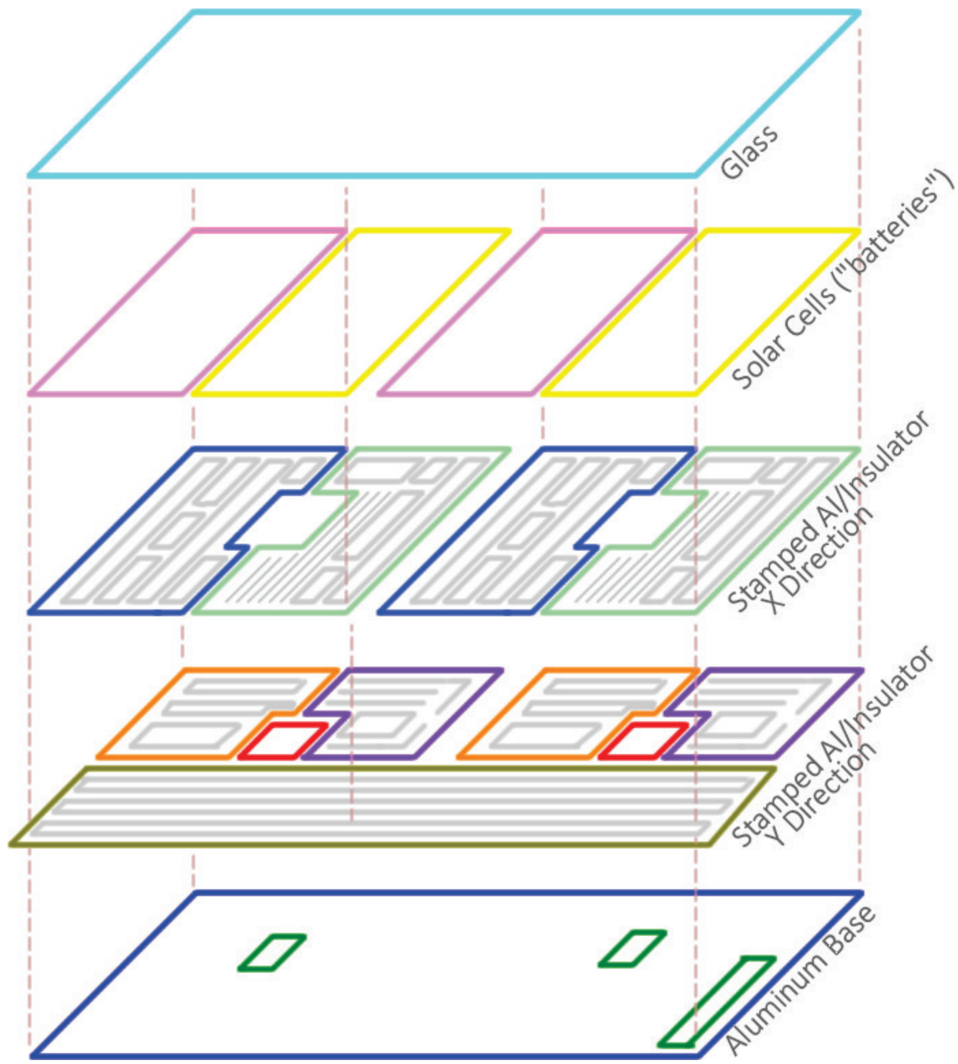
c) House with 220VAC power in cold climate where most energy goes to heat pump connected to geothermal 12°C (55°F) water.

Assembly

The factory continuous assembly process is fed with a roll of glass for the top and a roll of aluminum for the bottom, yet internal pieces are standard 1x1m sized inserts that are made from a different process.

Some layers can be thought of as continuous “roll-to-roll”, yet others are inserts in-between those layers to support circuitry needed by power engineer.

MA2 engineers work with solar cells mounted on wafers as well as films (plastic sheets).



Glass Layer -- Protects solar cells and holds together assembly. For an example of thin rolled glass, see Corning Willow Glass.

Solar Cell Layer -- Converts sun to electricity. Standard sized panels (e.g. 1x1m) are fabricated in a different automated process and assembled in the continuous process as needed. Several different standard types are available.

Conductor Top Layer -- Consists of a 1x1m stamped aluminum pattern on plastic insulator, where computer selects a standard pattern depending on one's needs.

Conductor Bottom Layer -- Similar to above, yet conductors instead move in direction parallel to rolled material. Also, the standard inserts are not 1x1m in size and are instead 1 x 0.6m. This is due to 0.4m wide aluminum plus insulation material that runs the length of each piece and is used to electrically connect multiple 1x1m panels and multiple 1x2m banks.

Aluminum Base Layer -- Standard inserts are mounted on top of this solid aluminum support layer which binds together panels along the entire length of each cut piece. Cut-outs (green) are made inside this layer to provide access to internal electronics from under the material, which is discussed in more detail later.

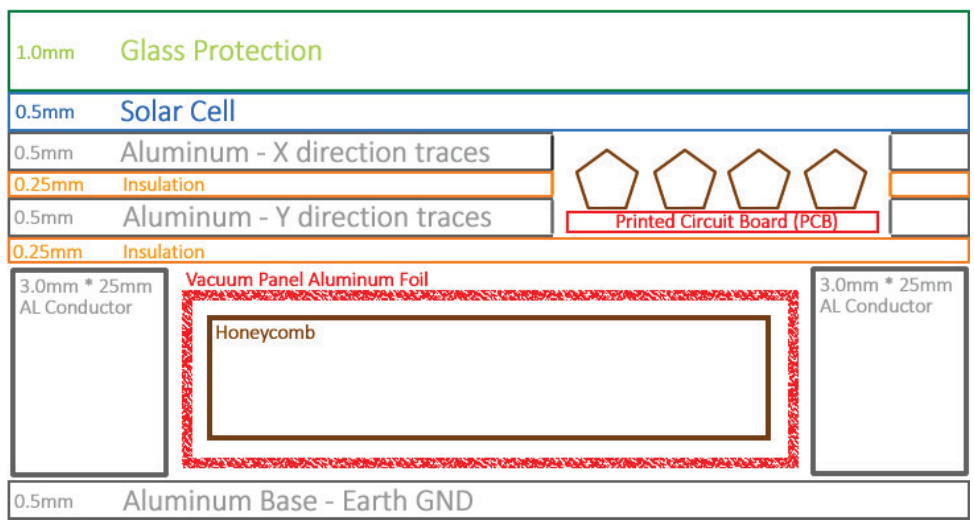
Create Solar Design Foundation (SDF, SolarDesign.org)

In order for all this to work, we need a standard way for customers to communicate with factories. Subsequently, MA2 creates the Solar Design Foundation (SDF). This involves a handful of programmers who create a website where Elon registers as a customer and Canadian registers as a manufacturer. Elon has lunch with Jerry and they decide to place 10 more recharging stations along the highway between SF and LA. While Jerry is in the men's room, Elon goes to SolarDesign.org and specifies an 800 strip 350VDC project where each strip is mounted on a 2m x 10m corrugated steel panel, to be delivered in stacks of 15 on the back of a flatbed truck. While Jerry washes his hands, Elon asks multiple companies to quote.

SDF programmers create the SolarDesign.org website which has a UI for both customers and for manufacturers. SDF also defines a standard data format that communicates a project. Ideally, SDF programmers sit in the same room as power engineers who design smart solar material, and manufacturing engineers who design the production line that makes it. As a team, they build the system. Later, they submit this as a proposed standard to a standards body organization.

SDF is a non-profit organization that consists of a handful of programmers, is privately funded, and receives no money from industry participants (customers and manufacturers). Initially, SDF is funded and controlled by MA2.

Place Vacuum Insulation Panel (VIP) into Smart Solar Material



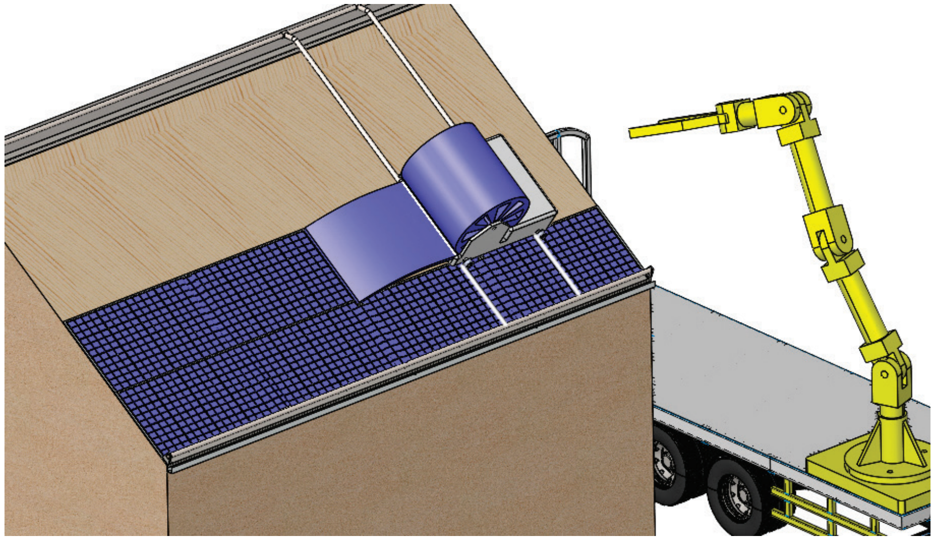
MA2 engineers develop vacuum insulation panels (VIP) that are cut to length in factory under programmable control, and supported by a small vacuum pump to mitigate permeation in the field. VIP might be made out of heavy aluminum foil w/ internal light honey comb material to keep panel from collapsing when under vacuum pressure.

Can we add this as one layer in our smart solar material? This is a fantastic time to do so. Why? Because we already are doing ~90% of what is needed with the smart solar material. We already: acquire dimensions, transmit dimensions to factory, cut material to order, provide power and processor (e.g. for vacuum pump), support transportation system, support installation system, support repair system, and support removal system, provide front protective surface (insulation), and provide back protective surface (aluminum base layer). Wow. The additional cost of adding VIP is small compared to the sum of all these other costs. To access the VIP for repair/replacement, one might remove corner molding from building to gain access to cavity that encloses VIP.

For more details, click here and here.

Chapter 4) Automate Installation of Solar Direct to Plywood Roof & Walls

Roof Robot Installs Direct to Plywood

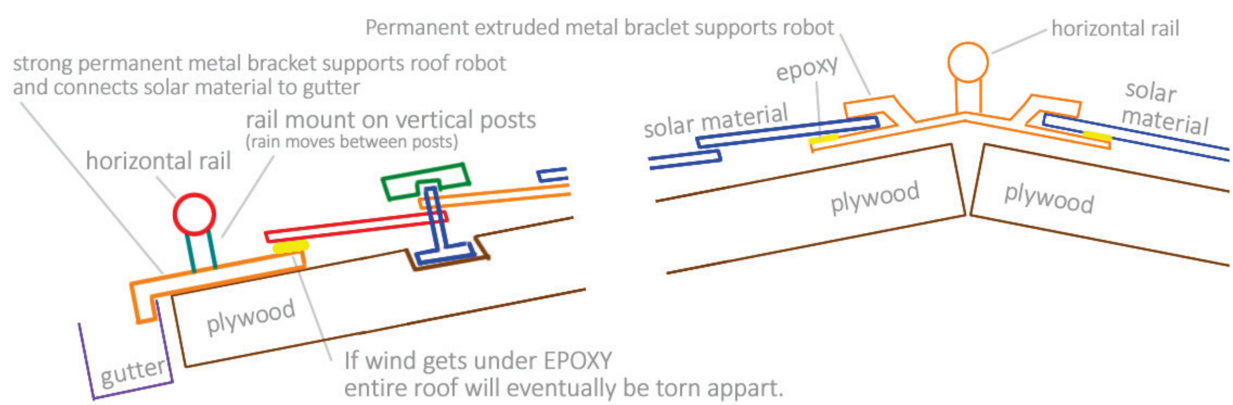


When mounting directly onto plywood a factory fabricates material, cuts strips as needed for each customer, and places them into a large shipping tube. At the building site a truck mounted mechanical arm places the tube onto a roof robot. The robot unrolls the material and places a bead of epoxy at each seam or uses metal molding on 1x1m grid to hold in place. If roof is 10m high by 20m wide for example, one might unroll 11 strips that are each 1m wide and 20m long. We use 11 instead of 10 since we need rain water to flow *across* overlapping horizontal joint.

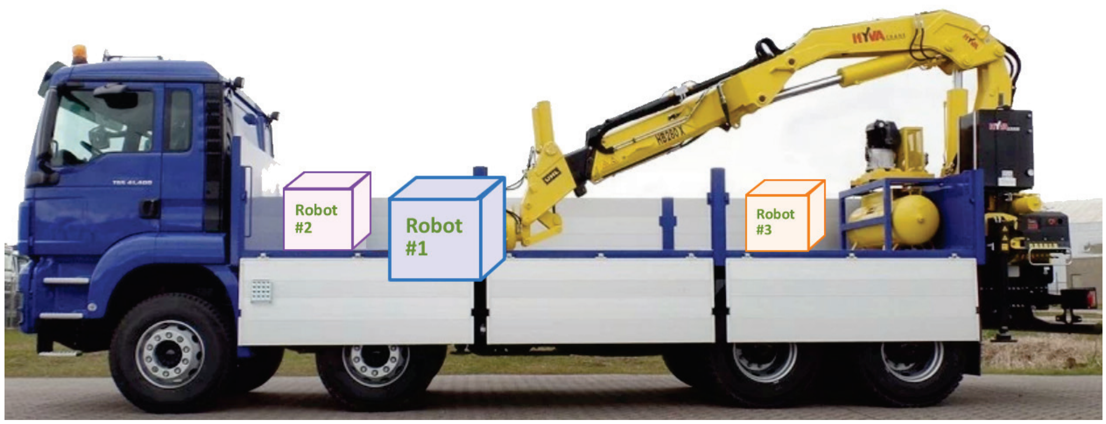
Roof Robots on Permanent Roof Rails

Shown to the right is an example solar panel [cleaning robot] which is similar in concept to our proposed system.

Standardized permanent rails at roof top and roof bottom (illustrated below) allow robots to move horizontally and vertically for purposes of installation, cleaning, and repair. Robots mount on a telescoping vertical truss which rests on the two permanent horizontal rails. Rubber wheels under robot and under truss help manage load. A standard electrical power connector for robot is available near roof, and robot manages electrical power cord as it moves. A standard wireless communication system enables robot to talk with operator via smartphone or tablet.



MA2 Designs Standard Robotic Construction Vehicle (RCV)



MA2 designs a standard Robotic Construction Vehicle (RCV) that consists of a flatbed truck and a mechanical arm. Robots attach to the end of the arm via a mechanical, electrical and software standard defined by MA2 engineers. These standards encourage industry participation since it causes their robots to be compatible with multiple trucks and multiple applications. Multiple robots are stored on the back of the truck and are accessed in a manner similar to that which is done with numerical controlled milling machine accessing different milling tools.

Manhattan 2 considers this a top priority and sets up a team to oversee engineers who design robots, build prototypes, and work toward accepted standards. The headquarters of this team is split between two locations, one close to a top robotics research facility such as CMU Robotics Institute, and the other close to construction vehicle manufacturing companies (e.g. UIUC is close to Caterpillar

Roof Robots

Different robots focus on a different activity:

- Shingle removal.
- Prepare plywood with holes, milled surfaces, and cut slots.
- Install custom designed pieces of solar material (e.g. 1m wide strips in spool) and epoxy glue edges.
- Repair solar material.
- Clean solar material, clean gutters & remove leaves.

This is similar in concept to a tool changer on a CNC machine, pictured to the right.

Controlling edges of solar material is important. If wind presses under solar material, a vertical force on material will cause end of epoxy joint to open like a zipper. This will result in entire roof being torn apart, slowly at first and then fast.

A flatbed truck with mechanical arm, pictured above, moves truss from truck bed to roof, moves robot from truck bed to truss, and finally moves spool of solar material from truck bed to robot.

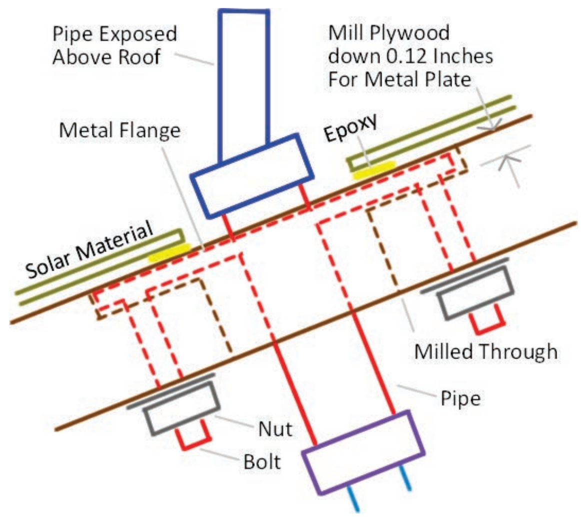
Repair Robots

Repair involves many options. One is to ignore the problem and let the internal printed circuit board bypasses broken components, which it does automatically. After 10 years, your smartphone might report that 3% of your roof is busted and display a map of broken areas.

If a tree branch smashes a 1x1 meter panel and causes disfigurement, one can cover with a 1x1m piece of dormant material that is glued to the faulty panel. This is not active electrically yet does provide a low cost cosmetic-only fix. Alternatively, a repair robot can mill areas where solar material interfaces with underside connectors, glue new sheet over old, and attach connectors from bottom.

Ports in Plywood

Roof robots install standardized metal ports in plywood that enable pipes, vents and power/control wires to transition from the internal attic to the rooftop. To the right is an example 3cm (1") diameter pipe vent. Plywood (brown) is milled 3mm (.12 inches) for a port metal flange and also milled all-the-way-through for the 1" pipe. Bolts welded to the flange (red) secure the port to plywood via washers and nuts (gray). Solar material (gold) is secured to the flange (red) with epoxy (yellow). Pipes at the specific angle that matches roof pitch (blue) attach to port upper coupling, while PVC pipe attaches to port lower coupling (purple).



Walls are important too

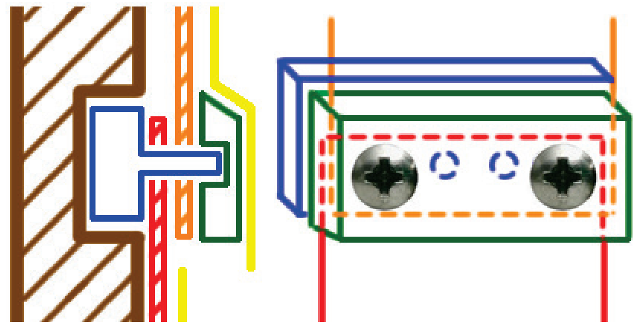
If you are on a ~[40° latitude](), then on average the sun will hit a vertical wall approximately [as much as]() a horizontal surface over the course of a year. Washington DC is approximately at this latitude, for example. In summary, walls are important too.

MA2 devises a system for attaching 1m to 2m wide rolled made-to-order *removable* material to walls *and* roof, with overlapping joints for rain water (similar to roof). Made-to-order entails taking out material for windows as specified on architectural drawings via sheer or mill. If material internally places circuitry on a 1m x 1m grid, then architect might place windows within grid to reduce wasted circuitry that is disabled via a cut.

MA2 Develops Standard Mounting System For both Roof and Walls that supports Removal

A milling/drilling robot performs preparatory work before attaching solar material. This includes milling down plywood for port flanges (e.g. pipes, vents), milling grooves for solar mounting rails (blue in below illustration), and milling out holes for electrical connections.

The illustration to the right shows how one might attach solar material to a vertical wall *or* roof. Horizontal metal rail (blue) attaches to plywood (brown). Metal pins protrude from rail and interface with solar material both above (orange) and below (red) horizontal rail. If solar material is 1m wide, then horizontal rails are spaced at this distance minus overlap. Horizontal stainless steel cover (green) is bolted to base rail (blue) via stainless steel bolts. Material that is both above and below pins is clamped between these two rails. Rain water (yellow) flows across rail system.



A stamping machine places holes in solar material at fixed intervals along both edges (e.g. 3mm diameter holes every 3cm). This is similar in concept to an analog film canister with perforations along film edges, yet 25 times larger. Perforations also help control material via sprockets. Robotically controlled router mills slot in plywood for horizontal rail (blue) enabling one to press solar material flat against plywood.

Optionally, one uses vertical rails that interface with horizontal rails to help fix material in place and avoid flapping in wind (adhesive can help as well).

System supports installation, repair & replacement.

MA2 engineers also explore the following concepts:

- Should this rail system be used on a roof as well, instead of glue, to allow for easy replacement?
- This could be an electrical shock hazard if material is low to the ground and internal layers are exposed; therefore engineers need to adequately address safety issues. Perhaps solar panels on first floor are low voltage with respect to earth ground?

Three Approaches to Solar on Plywood Walls and Roof

Fundamentally there are three ways to attach solar to plywood walls and roofs:

1) Roll material onto plywood, as described previously.
2) Place material onto thin & light corrugated steel or aluminum panels, stack on back of flatbed truck, install via crane.
3) Same as above, yet skip plywood and instead use heavier metal panels (for rigidity) and connect them directly to wood beams (e.g. 2x6 framing) without plywood.

MA2 engineers simulate all approaches and identify costs, advantages and disadvantages of each.

In all cases, architectural software must interface with factory and robots to manage the unique geometry of each case.

We should be able to achieve massive adoption due to the fact that each shingle and each clapboard (horizontal wood board on side of building) is installed by hand, one at a time. This high degree of manual labor increases cost dramatically. Subsequently, the automated low carbon alternative that pays you money each time the sun shines should be very popular.

The 100 Year Roof

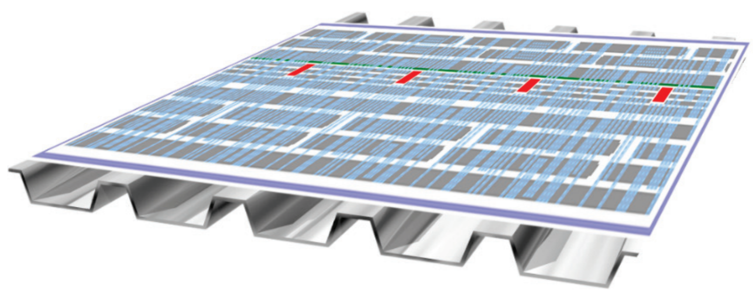
To achieve viral adoption of energy saving technologies, solar systems need to be less expensive than traditional roofing without solar. A measurement in the life-cycle cost calculation is longevity (i.e. costs). To achieve longevity we work with stable materials on building surfaces. Plastic, rubber, caulking, tar, asphalt, wood and paint are not stable over long periods of time due to ultraviolet decomposition and decay from oxidation and rot. Therefore, glass, stainless steel, anodized aluminum and copper are the materials of choice for long term durability and low life-cycle costs. MA2 engineers explore concepts in simulation that lasts 20, 50, and 100 years; and compares costs. Is a low cost 100 year roof possible?

Chapter 5) Automate Installation of Solar onto Corrugated Steel Panels & Frames

It is a common practice to ship corrugated steel panels on a flatbed truck to a building site and then install onto a metal frame via crane. An example of this is shown below.

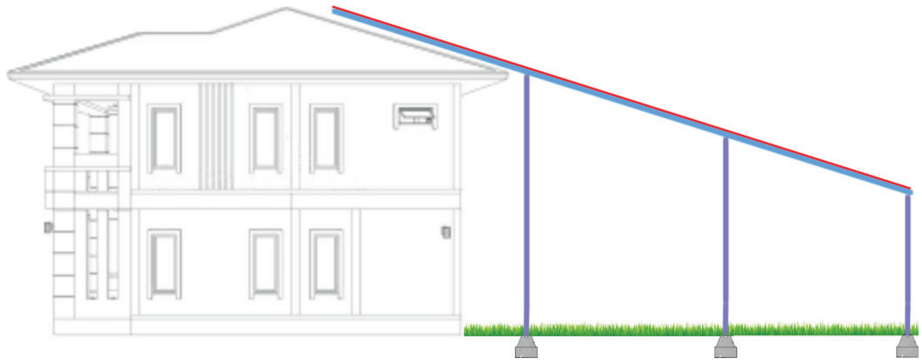
Also, smart solar material can be factory installed to the outside surface of corrugated steel via adhesive or spot weld, for an additional cost of ~$106/m^2$.

This should be popular with commercial buildings, warehouses and multifamily residential structures. To avoid damage to solar components while spot welding, one can first weld solar material aluminum bottom layer to corrugated steel and later add upper solar layers via adhesive.



Solar ‘Coated” Corrugated Steel is Exciting

Here are several suggestions of what one might do with a low-cost solar on corrugated steel.



The illustration at the right shows a 15x12m (40x50ft, 180m^2) 24kW solar array that partially covers a back yard. This might look strange at first blush yet 20 years from now fossil fuels will likely be much more expensive. A homeowner with several electric cars might sleep soundly knowing they have a 24kW back yard.

If solar costs are low, relative to revenue from selling electricity, then incentive to build becomes massive. Therefore, it is imperative that MA2 engineers figure out how to mass produce material and automate solar installation. If they are successful, we can expect to see much solar over homeowner yards, patios, parking lots, farm land, and gov't land near highway.

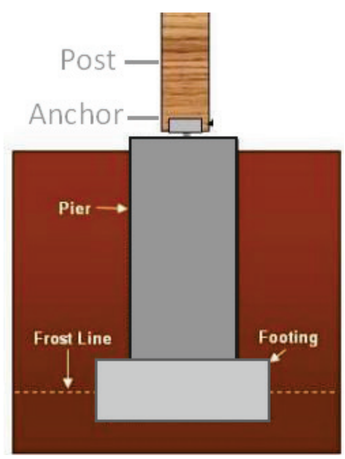
Automated Solar Corrugated Steel Installation

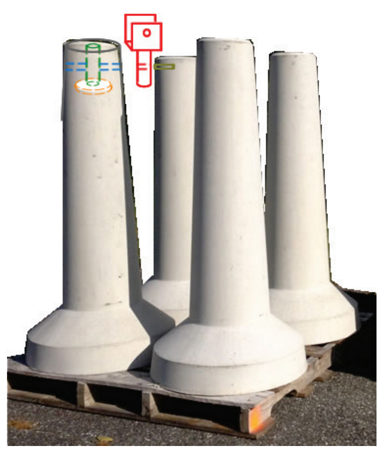
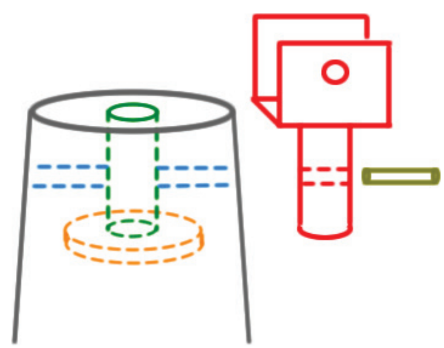
In order to achieve our goal of being less expensive than traditional systems, we keep installation cost low. Below are several ways to automate the installation of solar corrugated steel over metal framing.

Website: Website manages design and installation of a structure made out of corrugated steel panels. This supports projects that range from small to large. A small example is a one vehicle carport, and a large example is a commercial parking lot. The designs are simple free stranding rectangular structures. End user specifies system with key parameters such as width (e.g. 3meters * N), height (e.g. 3 or 4meters), and length (e.g. 2meters * K). Website creates drawings and files that describe the design. This helps obtain building permit from city/country inspector, helps obtain quotes from suppliers, and helps to define project in detail. In many cases, these designs are a starting point from which one customizes.

Ground Penetrating Radar Vehicle: Vehicle with ground penetrating radar and cameras visit site to collect photographs and look for underground obstacles that conflict with foundation. Customer interacts with vehicle in a manner similar to Uber. For example, website shows one of these 30min away, end user clicks OK, vehicle visits, scans, photographs, uploads data to website, and credit card is charged ~$250. This helps city/county to quickly issue building permit. For details, click here ("Article5_55F_at_5m.docx" / "Radar, Rocks, Inspectors and Website").

Foundation Installation Vehicle: The Robotic Construction Vehicle (RCV) automatically installs precast footings, possibly directed by website design files. It drills with an augur attachment and then uses truck mounted mechanical arm to move precast footings from truck to hole. Traditionally, contractors implement this with cement poured onsite, yet since we have a truck with automation close to hole, we make use of its strength and drop this in. Several different standard diameters and heights are offered.



The interface between cement pier and anchor is standardized. Above we show standard sized pipe (green) welded to plate (orange) burred in concrete. Customer selects between one of several standard sizes depending on load and up-force (hurricane & earthquake push up). Different standard sized anchors interface with this hole. An example anchor that attaches to a wood post is shown above in red. Aluminum framing uses a different kind of anchor. In all cases, a pin (gold), bolt or spring loaded tab slide from side to hold anchor in place.

Frame & Panel Installation Vehicle: The Robotic Construction Vehicle (RCV) uses a robotic arm to assemble aluminum framing and then drops corrugated steel panels onto frame. For larger arrays, the RCV might deploy panels stacked on the back of a flatbed truck.

Each vehicle has one operator that drives to site and oversees operation.

Same Platform for Automated Framing, Foundation, Direct-to-Plywood

The standardized RCV robot system described previously relates directly to robotic systems that build solar framing and their foundations since both use the same flatbed truck and same robotic arm, yet with different robots. Drilling pier foundations is done with a robot that contains an auger bit and is steadied with a mechanical link between truck frame and robot.

Manhattan 2 considers PV solar and geothermal robotic installation systems to be extremely important since we need low cost to encourage adoption, and robotics drastically reduces installation costs.

Parking Lot Opportunity

Can Manhattan 2 engineers create robotic technology that enable parking lot owners to cover their parking lot with solar, sell electricity, and break even in ≤ 10 years? If so, every parking lot in the world with a decent breakeven will move in that direction. We estimate the parts and labor cost to make the smart solar material (everything above corrugated steel, still at factory) at \$0.82/Watt (\$106/m^2, 45° to sun). If the retail price of electricity is \$.12/kWh then this works out to a breakeven of 4.7 years at 4hrs sun/day, just for the smart solar material. Can framing and corrugated steel, parts and labor, come in at \$106/m^2 as well? If so, your break-even is 9.6years. If one can sell electricity to city at more than \$.12/kWh, this system becomes more attractive. For more details on parking lots, click here.

Chapter 6) Design & Build Production Line That Makes Smart Solar Material

MA2 engineers design and build multiple prototype production lines that fabricates smart solar material. An example of what this might look like is shown below. In some cases, machines already exist. In other cases they need to be adapted, programmed, or designed from scratch. MA2 engineers do what it takes to build a production line, create sample material, and accurately calculate cost. MA2 funds prototype production lines at universities interested in automated solar (e.g. CMU, GIT, UICU, UCB, UMA, and MIT).

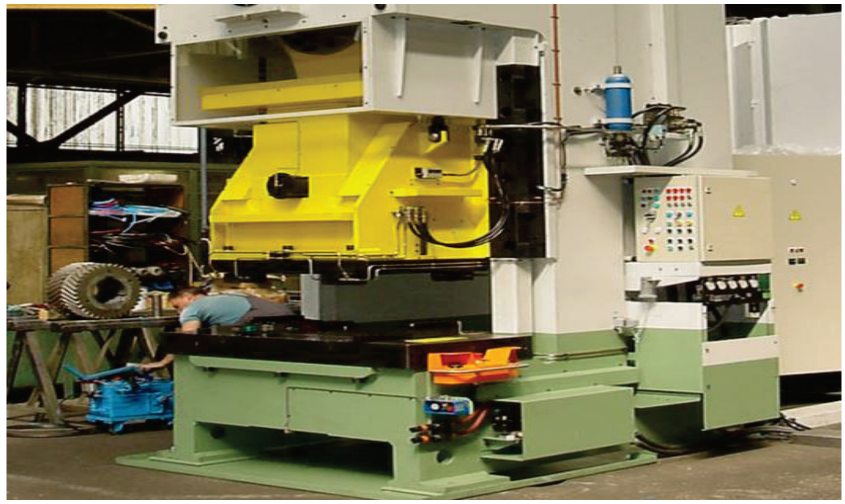
If an 80hr/week production line creates a 1x1m section every 10 seconds, then 1.5M square meters will be produced each year (52*80*60*(60/10)). A $5M capital cost could then be depreciated in 1 year at $3.30 per square meter (5/1.5). Material cost (electronics, metal, glass, solar cell) might be more like \$106/m^2, therefore $3 is negligible. If a home is 100m^2 then this one line could cover 15K homes each year (1.5e6/100).

Manufacturing Universities are invited to Participate

MA2 builds prototype production lines at America's top manufacturing universities who are interested in smart solar production. The University of MI @ Ann Arbor is heavily involved in manufacturing and would be a terrific site. The University of Illinois @ Urbana-Champaign is near Caterpillar and John Deere, and their future machines might be involved in handling solar material. Subsequently, top engineers from those companies are invited to participate in the sabbatical program to help them familiarize themselves with solar fabric, and to help MA2 engineers understand requirements in the field.

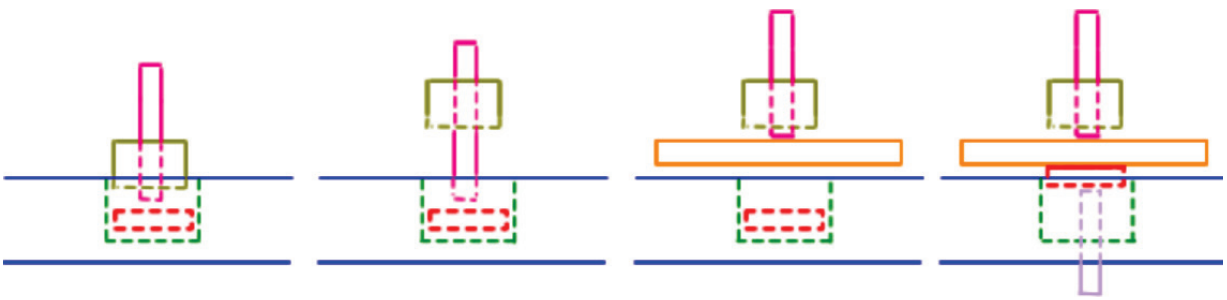
If capital equipment cost is $5M for each line then MA2 might support 6 different sites throughout the USA at a total cost of $30M.

Fabrication of Stamped Aluminum 1x1m Panels

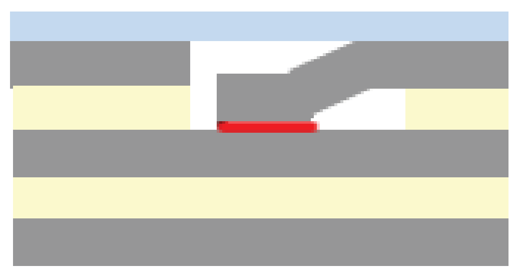


The typical sheet metal press, shown to the right, cuts conductors, lifts, and then ejects items via mechanism.

Our case is a little different: (1) Conductors (shown in red below) are pressed into the mold (green), (2) upper stem (pink) keeps conductors in mold while press (gold) retracts, (3) plastic sheet (orange) with adhesive is placed above mold, and (4) lower stem (violet) presses conductors against plastic. This process allows us to make complex inserts that carry current at extremely low cost.



Connecting Two Layers



A stamping machine punches holes in aluminum and insulation to make room for internal components and spot welded tabs. In the picture below, the red area shows a spot weld that connects top aluminum layer to middle aluminum layer. Welding occurs before the lower insulation layer and the upper solar cell layers are adhered to two conductor layers.

Gold Plate Electrical Contact Areas

One can gold plate aluminum by first applying zinc, then copper, and then gold. Plating is done to small portions of 1x1m panel by only exposing the required regions to liquid chemicals ("selective plating"). One can easily solder to gold; therefore this provides a means by which one can connect aluminum layers to a PCB, or connect aluminum to physical connectors under solar material. If one wants a reliable 100 year connection that involves much current and few failed joints, then soldering to gold must be considered.

Cut, Stamp, Roll, Ship

Material is cut via a sheer under computer control, as needed. After cutting, it is placed into a tube and shipped to building site, or attached directly to a corrugated steel panel. In some cases, cuts are done at an angle to match roof contour. If applying to residential plywood, the solar material factory has a detailed drawing of required shapes and cutouts, and prepares as needed for each building. After sheering, edges are treated to avoid water from entering system (e.g. dip edge into molten glass?).

MA2 Engineers Spawn *Automated* Solar Industry

The largest metal fabrication companies in the world are invited to visit and send their top engineers to these sites to work on sabbatical for 1 to 24mths. All drawings, software, and test results are made available to the public via a free license. In order to solve the global warming and energy crisis, MA2 freely supports all countries with smart solar, instead of having them burn up the planet with CO_2.

Why have the above concepts not been developed?

The primary reason the above systems have not be developed is there are no standards for interconnecting and controlling the above components. This includes physical connectors, electrical signaling, and communications protocols. For example, there no standards that attaches multiple solar elements to produce variable voltages for primary energy consuming subsystems. Companies will not develop a product if it cannot sell, and it will not sell if it cannot be used, and it cannot be used if there is no easy way to assemble the system.

This is a classic chicken and egg problem. Solution? Manhattan 2 simultaneously develops components, creates prototypes, ensures system works properly, turns prototypes into standards, invites industry to participate in all phases, and provides all materials to industry on a free basis.

Big Effort, Big Benefit

Implementing the smart solar projects described in this document involves hundreds of outstanding engineers who work over a 5 year period at a cost of hundreds of millions of dollars. This is a big effort. The benefit is big too, if well done.

Chapter 7) Smart Solar Math

Smart Solar Cost Estimates

We estimate a 100 square meter 13kW roof to have a material cost above plywood or above corrugated steel of \$11.2K (\$.82/Watt, \$106/m^2, 137W/m^2, 45° angle). This provides sun to electricity conversion, DC to AC conversion, and a water barrier. Since this replaces \$15K of shingles (parts and labor), \$11K is attractive. This calculation assumes 21% solar cell efficiency, sun at a 45° angle, and [\$.17/Watt] superior high-efficiency mono-Si cell material.

Alternatively, if sun hits panel directly (0° to normal), we get 194W/m^2 (19kW for 100m^2 array) and our cost per watt is \$.58. This is not realistic, yet is good at comparing to other products that assume sun is normal to surface.

The following table breaks down this estimate.

one 1x1m panel	one 1x2m bank	entire array	per watt (sun 45°)	per watt (sun 0°)	
1	2	100			surface area, square meters
\$ 48.98	\$ 97.97	\$ 4,898	\$ 0.36	\$ 0.25	cost of materials (glass/SolarCell/AL/Insulator/AL/Insulator/AL), no processing costs
\$ 15.58	\$ 31.15	\$ 1,558	\$ 0.11	\$ 0.08	cost of mosfet/triac switches
\$ 6.25	\$ 12.50	\$ 1,250	\$ 0.09	\$ 0.06	PCB + Analog Monitoring Electronics + Digital Processor (no mosfet/triac switches)
\$ 35.00	\$ 70.00	\$ 3,500	\$ 0.26	\$ 0.18	factory processing cost (100% guess, stamping, spot welding, gold plating tabs. adhe
\$ 105.8	\$ 211.62	\$ 11,206	\$ 0.82	\$ 0.58	TOTAL Costs To Make Solar Material (does not include installation labor or central c

The \$11K includes PCB's (electrical parts and labor), factory lamination and stamping (we guess \$35/sq. meter), raw material (glass, solar cell, aluminum), and MOSFET/TRIAC switches. This does not include installation labor and material under the plywood (cabling, connectors, and central controller). For a spreadsheet that models these costs in detail, click here.

With 4 hours of sun each day and grid electricity selling for \$0.12/kW·h, our 13kW array produces \$2.4K in electricity each year (\$0.12*4hr*13kW*365days/yr). If one sold all the electricity from this panel to the grid at \$0.12/kW·h (typical price), they could make \$2.4k/yr. If the price of electricity were higher, low cost solar would become more attractive.

Cost of Solar Material

A square meter of superior high-efficiency 21% mono-Si solar cell material is approximately \$33 per square meter (\$.17/Watt, 194W/m^2). A square meter of 1.5mm thick sheet aluminum cost \$9 (1*1*0.0015= 0.0015 m^3, 0.0015 * 2800 Kg/m^3 = 4.2Kg, \$2.20/kg * 4.2Kg = \$9). One square meter of 1mm glass is approximately \$5. Subsequently, the cost of raw material described in this document is \$48 per square meter (\$5/sq. ft.). This does *not* include labor costs, fabrication costs, and electronic components/PCB costs.

The material-only (glass, AL, plastic, solar cell) cost for a single-family 100m^2 (1076 sq. ft) roof would be \$48 * 100 = \$4.8K.

AL	GLASS	Plastic HD	Solar Cell			
1	1	1			sq meter per 1x1meter panel	
1.5	1.0	0.5			mm	total thickness (all layers of each type of material)
2,800	2300	950			kg/cubic meter	
$ 2.20	$ 2.35	$ 1.00			$ per kg	
4.2	2.3	0.5			Kg of material per sq meter of panel	
3	1	2			# of layers of material	
0.50	1.00	0.25			mm	thickness of each layer, average
$ 9.24	$ 5.41	$ 0.48	$ 33.86		cost per 1x1m panel	
			$ 48.98		total AL/Si/Insulator/SolarCell material cost per 1x1m panel (does not include processi	
			$ 0.36		cost per watt (material only, no processing cost, no labor installation, no electronics)	

Sun Math

Sun provides 900W per square meter given a ray of sun normal to a surface. This is 636W (900*.707) given a ray 45 degrees to a surface. If a solar panel is 21% efficient and sun hits the panel at a 45 degree angle then one will harness 134Watts per square meter (636 * 0.21). With 100m^2, this works out to be 13k watts (134W * 100), which is a decent amount of energy.

Let's think about a 24 hour cycle. Much of the USA receives at least 4 hours of sun per day. If you pull in 13kW for 4 hours this would result in 56kWh (Kilo Watt Hours) of energy (13*4).

The average USA house consumes 35kWh of electricity per day. In theory, the 56kWh from solar could replace this, however, this is not taking into consideration a furnace burning natural gas in the basement for heat.

A house typically burns 30K BTU (9K watts) of natural gas in a furnace for heat in the winter (30 x 40 x 10 ft., 50°F increase, very good insulation). If the 56kWh from the solar went toward this, you would get 5 hours of heat (56/9=6).

A house typically consumes 30K BTU (9K watts) for cooling in the summer in a warm region that receives 6 hours of sun per day (56*6/4=84kWh). If the 84kWh from the solar went toward this, you would get 9 hours of cooling (84/9=9).

Water Storage Math

If one has a 200 gallon tank of water in the basement with vacuum insulation (similar to a thermos for coffee), and they heat 80°F water to 200°F, then they can store 120*8*200=192K BTU. This corresponds to 55kWh (192K*0.000293), which is similar to what we are pulling in from the sun. In other words, we can dump the energy from the sun into heating or cooling a tank of water in the basement. This means the 4 to 6 hours of sun can be used over a 24 hour period for house heating and cooling.

To get through a hot summer night, one can chill water that is later pushed into a forced air heat exchanger. For example, one can chill 200 gallons of water from 75°F to 35°F to store 40*8*200=64K BTU, which corresponds to 19kWh (64K*0.000293). Alternatively, one could chill 200 gallons from 72°F to 32°F and then convert to ice to store 300K BTU ((40+144)*8*200).

In some cases, a 100 to 200 gallon heating/cooling "battery" is a great low cost way to store energy when the sun goes down. For more comments on water storage, click here.

Chapter 8) Example Smart Solar Home

There are many different ways of managing solar cells in an array. We will present one simplified concept that we refer to as our "100m^2 Standard Array". This is 100m^2 (1076 sq ft) in size, 21% efficient (sun to electricity), and assumes sun is at a 45° angle. It produces different voltages at different frequencies, as needed, for multiple consumers in a building. It is estimated solar material switching electrics to waste 4% of the energy and analog filtering another 4%, subsequently, one can expect 12kW (13.7*.92) pure clean sinewave average power output (post analog filter) while the sun is shining (e.g. 3 to 5 hrs/day).

Definition of Terms

We define several terms to communicate our concept:

Solar Roof Controller -- Box with a microprocessor that manages solar array (e.g. 100m^2). This system communicates with all PCB's inside of solar material (e.g. 1 PCB for each 1x2m bank) and determines power output for each 1x2m bank (DC or AC, voltage, frequency).

HVAC Controller -- Box with a microprocessor that manages heating and air conditioning system (heat pump, fans/dampers in ducts).

Solar Cell -- This can be thought of as a battery that turns on when the sun shines. Its voltage varies from 0.6V to ~120V depending on how it is configured.

One Panel -- This is a 1x1meter (3x3ft) area of the solar array. In all our calculations, we assume it is 21% efficient and we assume light hits it at a 45 degree angle. The sun produces 900W per square meter, yet after being discounted by the efficiency and angle, we expect ~137W from this panel (900*(.21)*Sin (45)). These are nominals values used to characterize our simplified concept.

Switch -- On/off switch implemented by a MOSFET or TRIAC and controlled by the Solar Roof Controller.

Low Voltage Switch -- This is a ≤ 60V MOSFET switch. An example would be the NVMFS5C670 which sells for $0.34 in 1k quantity, has an ON resistance of 0.006 ohms, is good with up to 60V, and is 5x6x1mm in size. Four of these in parallel cost $1.40, provide 0.0015ohms of ON resistance, and waste 0.15W of energy given a 10Amp load (0.015Vdrop * 10A).

High Voltage Switch -- This is a > 70V TRIAC switch. An example would be the ACST210-8B which sells for $0.50, has an ON voltage drop of 1.5V, and is good for up to 800V. This part wastes 0.45W (1.5V*0.3A) of energy given a 0.3A load. TRIACS are good with high voltages, yet they waste 1.5V due to their voltage drops. MOSFETS don't have this yet MOSFET devices tend to work at lower voltages (e.g. ≤ 60V).

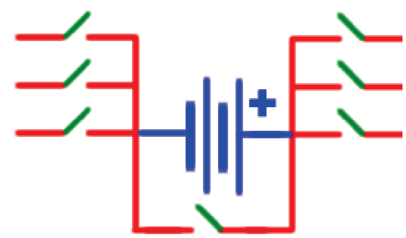
Low Voltage MOSFET Panel -- This is a 1 x 1m (3 x 3ft) region that contains low voltage MOSFET switches and solar cells that produce 0.6V, 1.2V, 2.4V, 4.8V and 9.6V, 19.2V, and 38.4V (for example). Each of these "batteries" can be turned on or off with an internal MOSFET switch. If one combines these in series to produce 1.3A at 76V (100W total), and does this with fourteen 0.006 ohm MOSFETS (two for each cell), then the MOSFETS would waste 0.15W (14*0.006*(1.3^2)); or 0.15% of the total (0.15W/100W). In this example, current is a constant 1.3A and the 0.6V cell produces 0.8W (1.3*.6), the 1.2V cell produces 1.6W (1.3*1.2), etc. Low voltage panels work with ≤ 60V voltages with respect to earth ground.

High Voltage TRIAC Panel -- This is a 1 x 1m (3 x 3ft) region on the roof that contains high voltage TRIAC switches and solar cells that produce 76.8V. There are actually 4 of these cells, each with their own TRIAC control. If each 1x1m panel outputs 137W then each of the 4 cells would produce 34W (0.45A * 76V). The 1.5V drop across each TRIAC waste 0.5W (1.5*.3A); or 2.0% of the total (0.5W/25W).

One Bank -- This is 1x2meters (3x6ft) in size and consists of *two* 1x1m panels and one control/switching PCB. One bank is capable of producing an 110VAC sinewave at 1.3Amps.

We differentiate between high voltage and low voltage panels because MOSFETS only operate at the lower voltages. TRIACS support higher voltages yet are less efficient. So we use TRIACs for higher and MOSFETS for lower.

MOSFETS, Money and Math



We combine multiple batteries in many different ways to produce AC and DC electricity at different voltage levels, different currents, and different frequencies; and we bypass faulty solar cell elements. If the solar cell are combined in series, their voltages add and if they are combined in parallel, their currents add. In our simplified 100m^2 example, we obtain significant control over 13kW of power with 3 switches connected to each cell terminal, and one bypass switch connected to both cell terminals. This works out to $420 for MOSFETS ($0.34*7*2*3.5*25) and $1050 for TRIACS ($0.50*4*2*3.5*75) for 100m^2. Cost per square meter works out to $15.70/m^2 using our example 110VAC 1x2meter bank (i.e. ($420+$1050)/100).

Variable HVAC

The traditional HVAC system turns on/off in a 30 to 60min cycle causing occupants to feel a little too cold or a little too hot. Subsequently, we prefer variable speed air-conditioning.

An AC motor wants to spin at a multiple of the rate of the 110/220VAC power, which is 50 or 60Hz. It is not easy to change this rate since the 50/60Hz frequency is fixed. There are techniques for creating variable speed motors; yet they are complex, involve inefficiencies, and are costly. Subsequently, we prefer the solar panel to support variable speed motor control via the following techniques:

- drive DC motor with variable voltage and/or variable current
- drive DC motor with variable switching speed
- drive AC motor with variable frequency created by switching MOSFETS

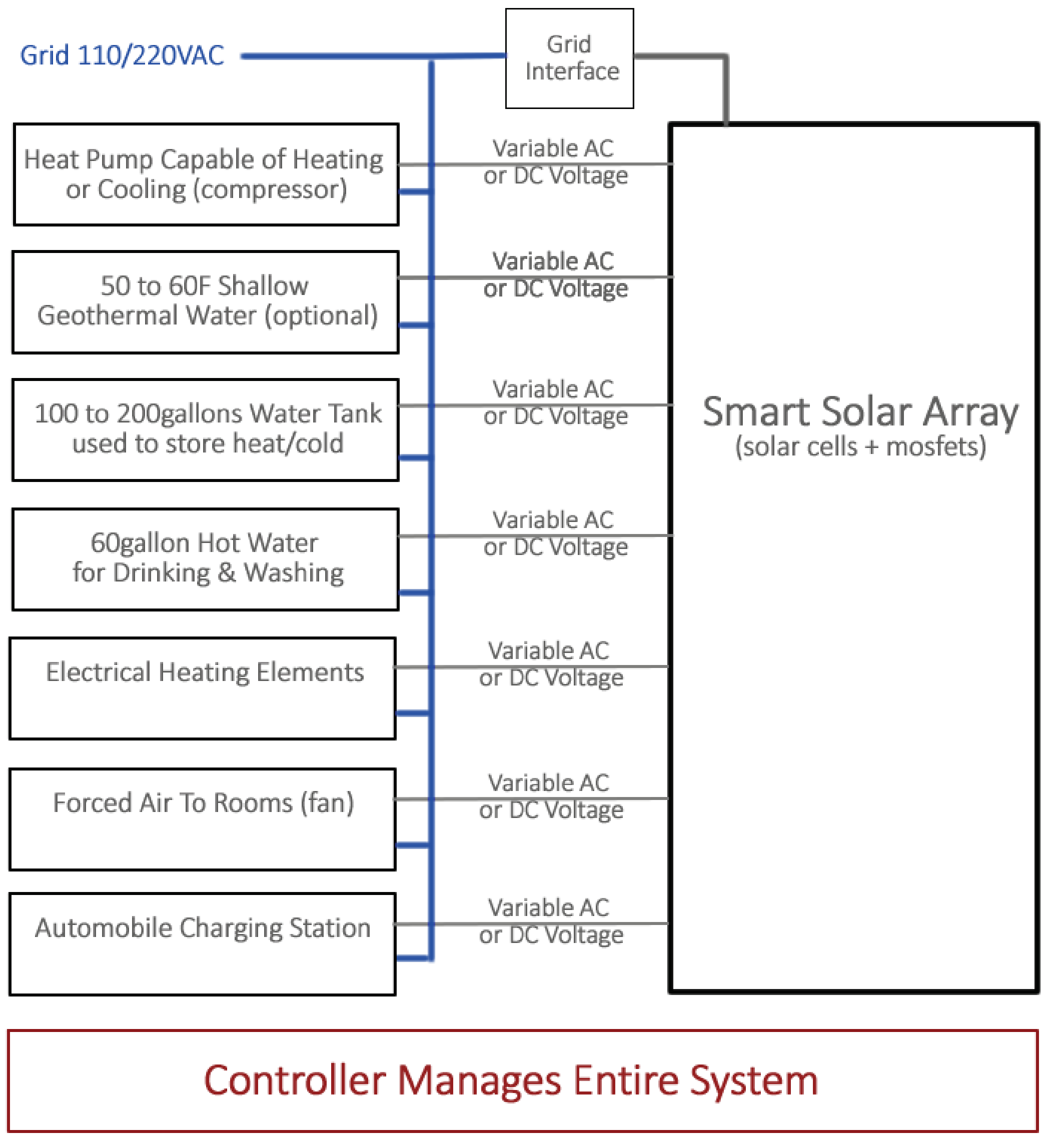
Advantages of Internal Switches

Solar material with many built-in switches provide the following advantages:

- Avoid external DC to AC converter to save cost, increase efficiency and improve reliability
- Easily build pure sinewaves, and not "modified sinewaves" which are closer to square waves
- Bypass faulty solar cell elements
- Support direct connection to heater, air conditioner, 110/220Vac power; and balance loads as needed
- Create variable AC voltages, variable DC voltages, and variable AC frequencies to help drive variable speed motors

System Design

A building's system design varies depending on building size, energy requirements, outside temperatures, and feasibility for geothermal assistance. The following illustration shows a system design for a typical one family USA home with a 13kW roof and shallow geothermal water at 12°C (55°F) from pipes 4m (13ft) underground.



The solar array provides variable AC and DC voltages to each major component, as needed, and as determined by a Solar Roof Controller. A heat pump does whatever needs to be done at the time. Shallow geothermal water at 50 to 60°F may be available; and if so, may be helpful at any given time. A water tank is used to store heat or cold; and is capable of creating ice to help store cold (without mechanically expanding against walls). This tank might include vacuum insulation to help reduce energy loss through its wall.

Storing Energy with Water

An important system component is a device which stores energy. A low cost way to do this is a tank of water, possibly with vacuum insulation. A small vacuum pump occasionally operates to maintain vacuum. In addition to energy storage, the water tank can be used to optimize system functions. For example, if one produces a sinewave and has a residual (1-|sine|) wave, this residual can be used to drive a heat pump motor and create heat or cold that is used at a later time. For details on water storage, click here.

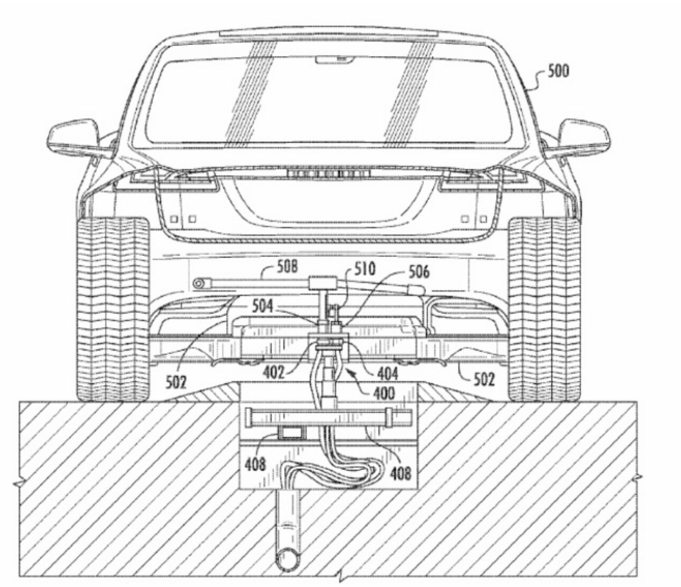
New Construction vs Existing Construction

New construction has the freedom to easily implement newer technologies, whereas existing construction is more limited. For example, new construction can easily set up a locked equipment room in the basement with outside double-door access to allow for large water tank delivery and installation; whereas existing construction typically has less basement access.

Chapter 9) Create Big Battery Borrowing System for Cars and Buildings

Create Big Battery Foundation (BBF)

Manhattan 2 forms the Big Battery Foundation (BBF) to help coordinate multiple companies involved with large batteries. BBF headquarters is near Tesla, Inc. and Stanford University in Palo Alto, CA. This organization consists of experienced engineers on sabbatical from industry (e.g. Tesla, GM, Nissan, BMW, Volkswagen, Renault, Honda, and Toyota), postdoctoral fellows, and PhD students.

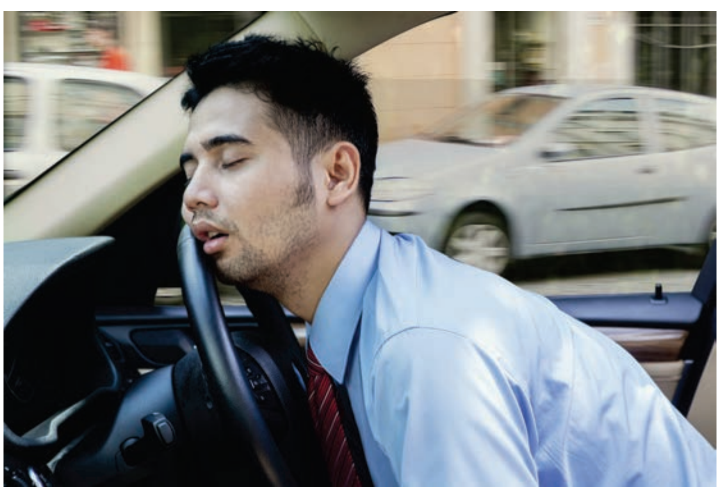


BBF designs Big Big-Battery Borrowing System (BBBBS)

BBF designs a worldwide system to coordinate the borrowing and handling of large batteries.

This system supports replacing a depleted battery w/ a fresh battery in less than 1 minute via a mechanism built into cement under car.

When gas runs out (which it will), gas stations will disappear. Instead, car owners will drive into a ~~gas~~ Battery Swap Station (BSS), swap out battery, and be charged for electricity consumed *and* wear on deposited battery.



This solves the electric car range anxiety problem, where drivers worry about running out of electricity while on long trips.

This solves the problem of falling asleep while waiting to charge, or falling asleep while waiting to wait to charge.

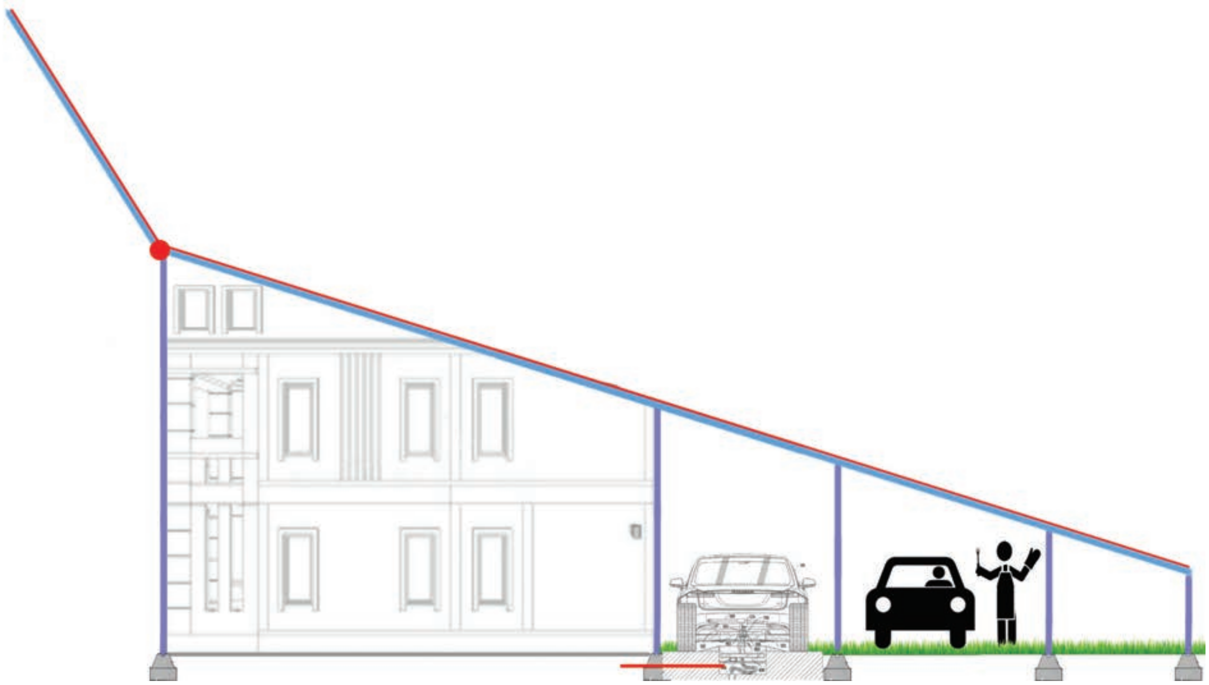
This reduces cost since low cost/low range batteries become more feasible.

Homeowners install swap mechanism in their driveway, own multiple batteries, and connect batteries to house power system. This means solar panels on roof can charge batteries, and batteries can power house when the sun is not shinning. The later point solves the solar energy storage problem (i.e. homeowner has electricity when sun is not shinning). And keeping big batteries out of house keeps Jim happy.

Tesla might consider a proprietary borrowing system with their own batteries, yet the density of electric stations is higher if Tesla collaborates with others (e.g. Toyota, Honda, and GM). The auto companies often do not talk; therefore a trusted 3rd party might be required to help coordinate.

Battery Powers House at Night, and Solar Panels Charge Battery during Day

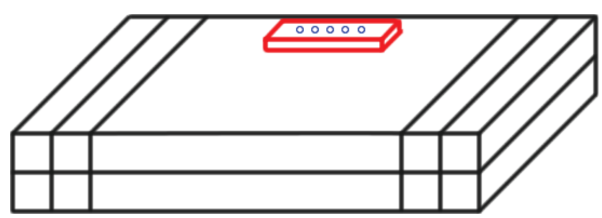
BBF is responsible for moving energy between batteries and buildings, in both directions. For example, solar array on top of house charges battery during the day, and battery powers house at night. BBF mechanical engineers design multiple concepts for cavities in concrete under car; where mechanism in cavity manages multiple batteries. Batteries are heavy (e.g. 1KLBs each) and therefore are best handled with steel frame on concrete, low to the ground. PVC pipe (e.g. 4" diameter) between basement and battery cavity helps to route power and data cables between two positions.



Also, BBF engineers design mechanisms for handling batteries in battery swap stations which store dozens or hundreds of batteries at one time. This includes cavities in concrete, steel rails, and racks.

Flexible Battery Mechanical Standard

BBF defines several standard widths, several standard thicknesses and several standard lengths; as illustrated below. For example; widths might be 0.6m, 1.0m, 1.6m; thicknesses might be 7cm, 10cm, 20cm, 30cm; and lengths might be 0.6m, 1.0m, 1.6m, and 2.5m. These support everything from a large truck to a small hybrid gas car.



Notice in our example we have 3x4x4=48 different permutations. This does not mean that BFF advocates placing all of these into production. Instead BBF leaves the decision about specific popular battery sizes for mass production to other people at other times (who control production). Perhaps only a few sizes are mass produced.

BBF designs a power connector standards that support small, medium and large.

BBF designs mechanical sockets that grip batteries of different sizes. For example, suppose Toyota makes 1m x 1m x 10cm batteries in their factory, yet 1m x 1.6m x 10cm are popular at battery swap stations; subsequently, Toyota engineers utilize a socket under car that accepts multiple battery sizes.

Challenges for BBF Engineers

In order for this system to work, BBF engineers must develop: low cost under-the-car mechanism, battery mechanical standard, power connector standard, communication standard, battery wear price model, smartphone user interface, car user interface, payment interface, and secure system to resist fraud.

BBF engineers design and build multiple prototype systems, yet are not responsible for adoption. Their job is to design, prototype and demonstrate. If they do a good job and the market desires this kind of system, their work might evolve into a worldwide standard.

All work done by BBF is made available to the public on a free gnu license basis.

Initially BBF is controlled by MA2, yet is eventually spun off as an independent foundation.

BBF Encourages Worldwide Industry Participation

BBF sponsors decision making seminars and pays travel expenses for key industry executives (e.g. one person from each of 8 auto companies). For example, BBF sponsors a week where *prototype* mechanical dimensions are decided (e.g. length, height and width). BBF prototypes are more likely to be well received if industry participates in making key decisions. This is not asking auto companies for a commitment. The decisions made at seminars only effect BBF prototypes and standards proposed by BBF to industry.

BBF engineers love all worldwide car manufacturers equally. They invite them to visit and invite industry engineers to take a 1 to 24 month sabbatical at BBF headquarters. BBF works closely with companies and organizations involved in energy storage; including Tesla, Honda, Toyota, Nissan, GM, Ford, NTSB, and NEC.

BBF covers salary and offers sabbaticals a room in a beautiful mansion to encourage participation by top engineers. Perhaps 3751 El Centro would work? If this is what it takes to attract highly productive outstanding engineers, it is worth the money.

Battery Swap Math

A $12K Tesla 75kWh (310mile range) battery wears out after ~200K miles and therefore has a wear cost of about $0.06 per mile ($12K/200K) or $18 per 300 miles, on average. However, this is an approximate number. A more accurate value depends on battery temperature and the percent charged while using. The customer is charged the more accurate value.

Electricity cost from the grid for 300 miles might be another $11 (i.e. $0.15/kWh*75kWh, 75kWh battery).

The gas, excuse me, electric station might charge another $20 for maintaining the battery borrowing and recharging system. Total cost works out to ~$49 (18+11+20), which is similar to gas (e.g. Toyota Camry gas cost ~$50 for 300 miles, Jan 2019 USA). This calculation varies depending on where and when you are. For example, in Jan 2019 Germany, gas prices are two times higher and electricity prices are *also* two times higher than USA.

Cover Neighborhood Electric Station with Solar, Not

If a neighborhood electric battery-swap service station lot size is 0.2 acres, for example, (750 m^2, 8K sq ft, 80x100sqft) and is completely covered with solar then it would produce 102kW when the sun is shining (13.6*750/100, 45° angle). With 4 hours of sun each day, one could fully charge five Tesla 3 batteries (75kWh each, 4*102/75) with that array. Not too exciting. What to do? If houses in the neighborhood have big solar arrays and export electricity, it could travel down the block, around the corner, and into a battery swap station.

> *Fundamentally, 300 miles of transportation involves a lot of energy.*

Chapter 10) Charge Electric Cars with Smart Solar

Cars contain DC batteries (AC batteries do not exist), therefore it is more efficient to charge them with DC electricity than to convert DC solar to AC and then back to DC. In other words, we want our smart solar cell array to connect directly to batteries and supply the correct DC voltage and current. Exactly how this is done is beyond the scope of this document.

Driving typically involves long distance *or* short trips on a daily basis (e.g. commuting and errands). The solar array can keep up with short range, yet long range is more challenging.

With 50% of a 100 square meter 13kW system, one could charge a 220mile range Tesla 3 automobile for *1 hour* a day and drive 45km (28miles) each day (220miles*.50*13kW/50kWh). Most areas of the USA receive 4 to 6 hours of sun each day; therefore 45miles/day is easily supported by a 13kW array.

If a family has two electric cars and one commutes to work each day, they can alternate cars such that one stays home to charge and one travels each day.

The typical two car garage has a surface area of 53 square meters (7x7m, 24x24ft, 576sq. ft.), and this can be covered with solar direct to plywood. Alternatively, one can have a carport that is covered with corrugated steel solar. Either way, this can produce 7kW of energy (13kW * 576/1070sqft) for 4 to 6 hours each day.

The average lot (yard + house) in the USA is 0.2 acres (8.7K sq ft, 800m^2). If one devotes 25% of this to solar (e.g. corrugated steel solar over house, patio, and cars), they could devote 26kW (13*2) to home and transportation.

Parking Lot Covered with Solar

Let's look at what one can do with a 1 acre parking lot that supports 150 cars. This is 45K square feet (242 x 180ft) and is 42 times larger than our 13kW example. If we covered this with corrugated steel panels and dedicated 100% to automobile charging, we could fully charge 43 Tesla 3 vehicles in 4 hours ((4*13kW*42)/50kWh); or fully charge 10 of these vehicles in 1 hour (50/5). Solar material (parts + processing) *above* the corrugated steel is estimated to cost $470K (42*$11.2K) and provide $101K/yr in electricity ($2.4K*42, 45% angle to sun, 4hr/day sun). This does not include the cost of foundation, framing, and corrugated steel. For more details, click here.

Highway Median Strip Covered in Solar

Let's assume cars driving on a countryside highway periodically stop at an electric station to swap out depleted batteries for fully charged. Let's assume this is done by a fast automated mechanism and also assume the station charges 1000 batteries each day. This corresponds to a solar system 22 times larger than our 150 car parking lot, described above (1000/42). Cost of materials above corrugated steel are estimated at $10M ($470K*22). If we devoted 60ft of median strip between lanes to solar, we would need an array 3 miles long for each electric station ((22*242*180/60)/5280). Alternatively, one could devote 1000ft x 1000ft of land (20m * 5Km = 60ft x 3miles = 300m x 300m = 100,000 m^2= 25acres).

The US Interstate highway system is 48K miles long. If one places stations every 8 miles, for example, we would have a total of 6K stations (48K/8). Cost of material above corrugated steel works out to $60B ($10M*6K). This is reasonable, especially if spent over 10 years. The total interstate highway system cost $500B. The $60B number does not include cost of framing, foundations, and installation. One would need significant automation installation to keep those costs down.

This system would support 1.3billion vehicle miles travel (VMT) each day ((220*6000*1000)/1e9) and 0.5Trillion miles each year (1.3*365) if fully utilized. The rural interstate VMT is 0.3T/yr and the urban interstate VMT is 0.5T/yr; therefore 6K stations is a reasonable estimate to assist in long distance travel.

If capital cost of material *below* smart solar fabric is the *same* as above (i.e. $105/m^2) and one charges 200 batteries each day at each station over 20 years, then capital cost would be $14 per battery per charge, which is reasonable (i.e. $120e9/(20yrs*365days*200charges*6000stations)).

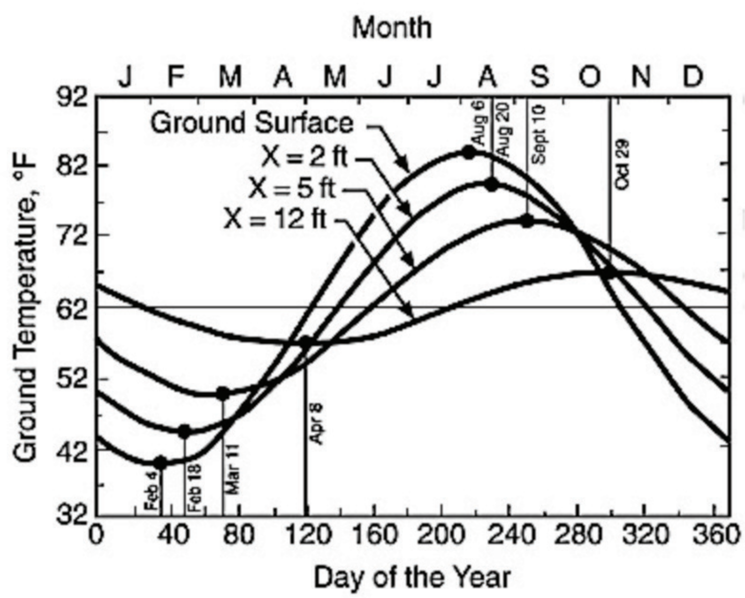
This system works best in regions that receive more sun per day, less snow per year, & have more gov't land nearby. Obvious, many areas of the country are *not* suited for this kind of system, especially urban areas, which is where most cars are driven. Eventually, safer nuclear might be required to support urban areas.

Chapter 11) Ground Source Heat Pump (GSHP) Background

Electricity from solar panels is great, yet not great enough. We need to do more. If you take a shovel and dig down 4 to 5 meters (~13ft) in the typical backyard, you will hit a temperature between 10°C and 15°C (50°F and 60°F), almost all year around. One can run water through a plastic pipe in this environment, feed this chilled water into a heat pump, and reduce energy required for air conditioning *and* heating. This technique is much more efficient at creating heat than burning natural gas or running electricity into a heating element. Also, it is much more efficient in the summer at air-conditioning then an outdoor compressor attached to an air heat exchanger. This technique is referred to as shallow Geothermal Heat Pump, and is popular throughout the world. However, it has a high up front cost (e.g. $16K), and installation typically creates a mess like you have never seen. For more details, please view this video or click here.

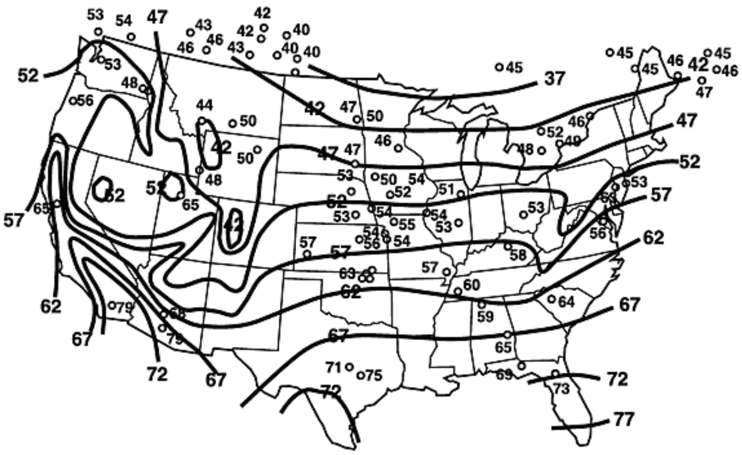
California Temperature

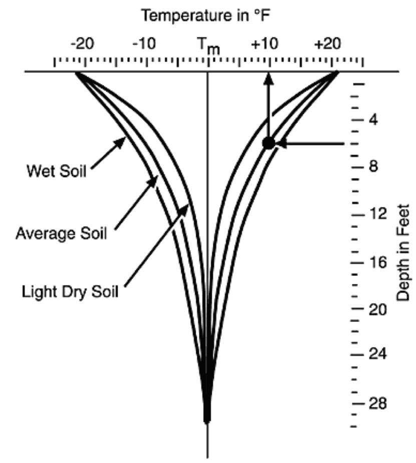
The graph to the right shows typical California temperature, underground, over the course of a year. At a depth of 4m (13ft), it is coldest (58°F) in April due to the ground slowly cooling during the winter, and it is warmest (66°F) in Oct due to warming during the summer (8°F swing). At a depth of 1.5m (5ft), we see temperature ranging from 52°F to 72°F over the year (20°F swing).



United States Underground Temperatures

The graph on the left shows average °F temperature (T_m) over one year at 8m (26ft) depths throughout the United States. The graph on the right shows how much this varies throughout the year, at different depths (warmer in summer, cooler in winter). The average average is 12°C (54°F). When air conditioning, one wants the underground temperature to be as low as possible, and when heating one wants this to be as warm as possible. Geothermal heat pumps make more sense in some regions than others.





Example Ground Source Heat Pump Home

Let's assume you want your New York home to be 24°C (75°F) throughout the year. At the beginning of the *summer* you use geothermal chilled water directly to cool the house without the use of a compressor. A water pump circulates geothermal water through the ground and into a heat exchanger inside your central air primary duct. For two months you dump heat from house into that water and the ground, and heat it up. When your back yard (yes, you are heating up your yard) reaches ~75°F you turn on your heat pump to move heat from home into the ground to make the ground warmer (e.g. ~85°F) and the house cooler. This uses much less electricity than heating an outside heat exchanger to ~110°F and cooling it with ~90°F air (which is traditional air conditioning).

During the first two *winter* months one uses the warmth in the ground (e.g. ~85°F) to heat the home, and then, when that water cools, one uses the heat pump and geothermal water in reverse. After pulling heat out of the ground all winter long the ground will become colder than normal (~45°F) and be ready to chill your house at the beginning of the summer.

When heating, the heat pump plus geothermal water uses approximately one third as much electricity as pushing electricity directly into a heating element. In other words, you get more heat if you push electricity into a compressor motor than a heating element.

In summary, shallow geothermal water significantly reduces the cost of both heating and cooling; yet installing the underground pipe is expensive and messy.

Traditional Pipe Installation Techniques

The typical way to install shallow geothermal pipe is to remove dirt, lay pipe, and replace dirt -- as shown to the right. This is expensive and typically represents 60% of the total cost of a geothermal system.

Chain Trench Digger

There exists a class of machine called a "chain trench digger" that digs a trench with cutting teeth attached to chain, similar to chain saw.

Chapter 12) Reduce HVAC Energy w/ Low Cost Geothermal Heat Pump

Underground Chilly Water Can Change Climate

If the typical house can obtain chilled water for $4K instead of $16K, the world's energy situation changes dramatically. This is because one can then trade the outside AC heat exchanger for chilled water at similar cost; yet chilly consumes ~37% less energy. Approximately 50% of energy consumed by a home is for heating and cooling. If you drop that by 37% you can reduce total home energy consumption by 18%. The Paris Climate Accord's USA goal is to drop $C0_2$ emissions 26% by 2025. Subsequently, this 18% would satisfy ¾ of that goal for USA households, without spending much money, *if* we can install geothermal piping more cheaply. Chilly H_20 can change climate.

We Need a new Class of Vehicle

Current trench digging machines supports depths of 1.5 to 2meter depths and are seldom capable of public road driving. We need a new generation of shallow geothermal pipe laying machine that can:

- Significantly reduce cost of installing pipe (e.g. $16K to $4K)
- If dirt is removed from trench, put it back into place
- Drive on a public road including interstate highway
- Place pipe at ≥ 4m (13ft) depths, which is twice that of today's typical system
- Install pipe in a small ~11x11m (36x36ft) yard
- Installation performed by one person to dramatically reduce cost
- Avoid rocks, tree roots and other obstacles
- Control thermal conductivity around pipe

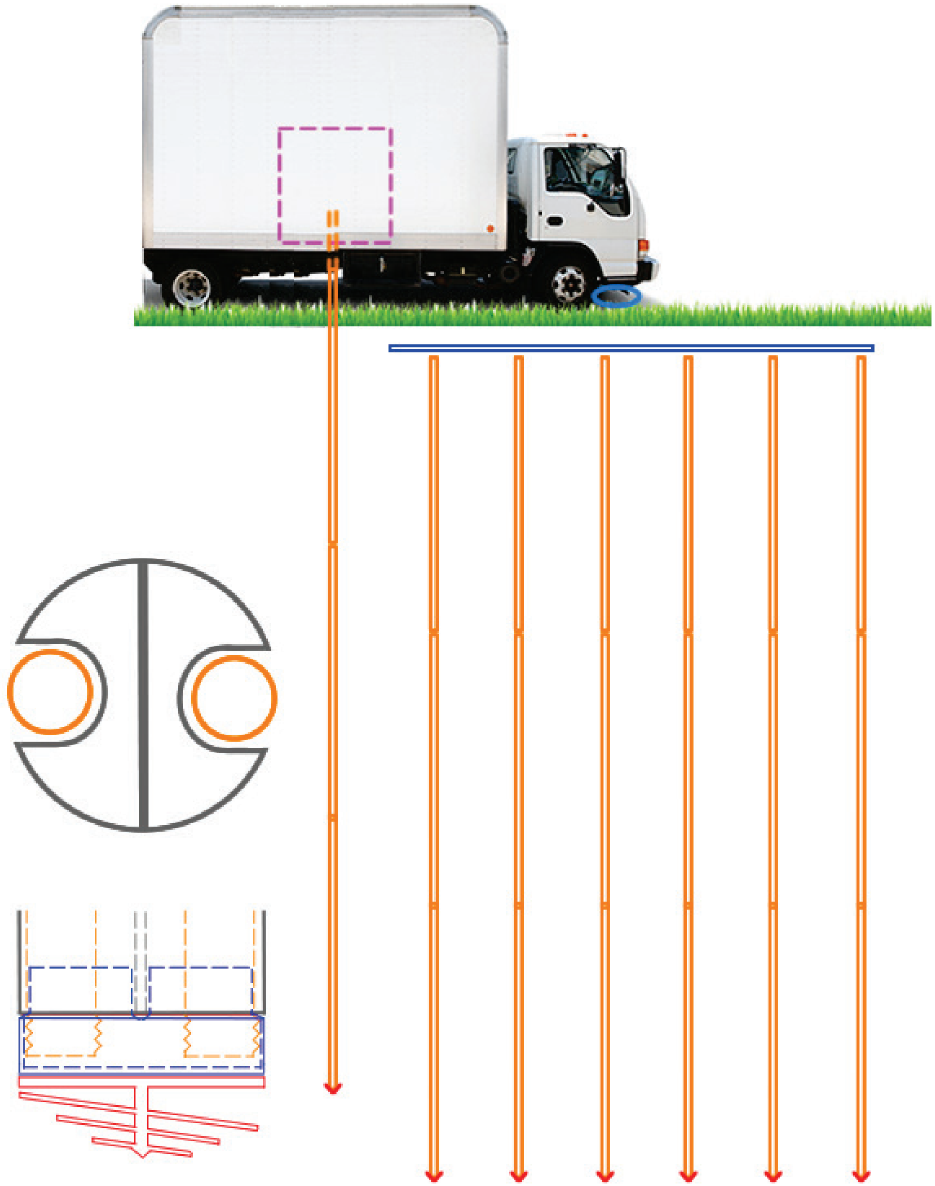
Engineers Evaluate Multiple Approaches and Prototype those that are Feasible

MA2 engineers explore *multiple* concepts that install pipe at lower cost. For each concept they create drawings, simulate, and produce cost models. If a design is feasible, prototype is built. Concepts include, yet are not limited to, Chain Trenching, Directional Boring, Underground Robotic Gimbal Chamber, and Vertical Boring.

Concept 1: Automated $4K Vertical Boring Machine

Currently, we have [expensive services that drill](#) two ~50m (150ft) deep 1ft diameter vertical bore holes to provide chilled water to a house.

Can we create a machine that drills 4 to 10 holes that are 50 to 120 ft deep in several hours, with one operator, at 4 to 6 times less cost?



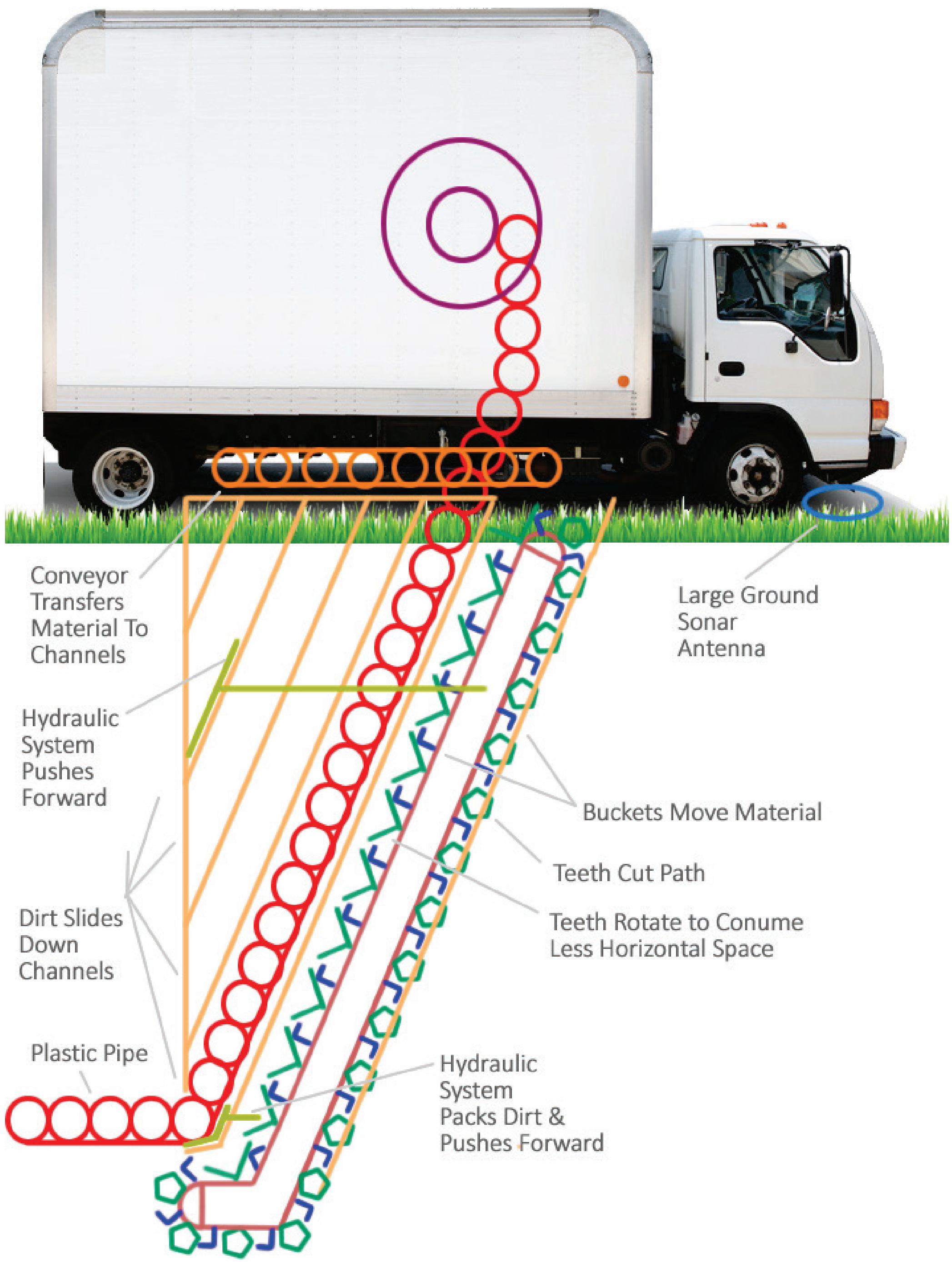
One method of pushing 3" to 4" diameter HDPE plastic pipe is to place removable steel pipes to left and right while drilling, as shown in orange. This helps in the following ways:

- Guide HDPE pipe in a straight line.
- Reduce stress on HDPE plastic while pushing pipe, enabling one to insert further.
- Reduce stress on lower HDPE due to weight of above HDPE.
- Move soil between head and surface via water in these pipes.
- Power rotating drill head via water/air in metal pipes.
- After drilling, metal pipes deliver grout to hole as pipes are withdrawn and low cost head is left in place. If grout is under relatively low pressure when it interfaces with soil, permitting by gov't is easier.

The bottom of the pipe might interface with a molded PVC (or aluminum) housing (blue), as shown in the drawing. A rotating bit (red) is driven by water via steel pipes. When drilling is complete, pipes unscrew from housing, extract from hole while inserting grout, and are then reused. The HDPE is extruded and therefore can have a complex geometry. In this concept, we have two internal cavities, one for geothermal water that is pushed into the pipe and the other for return. Obviously, the two HDPE cavities are connected at the bottom.

For more details, click [here](#).

Concept 2: Automated Chain Trenching



The below theoretical concept shows a chain digger that lowers from inside a box truck, cuts its way into the ground, and then moves forward. As it moves, it deposits a plastic water-carrying geothermal pipe 4m (13ft) underground, and covers it with dirt.

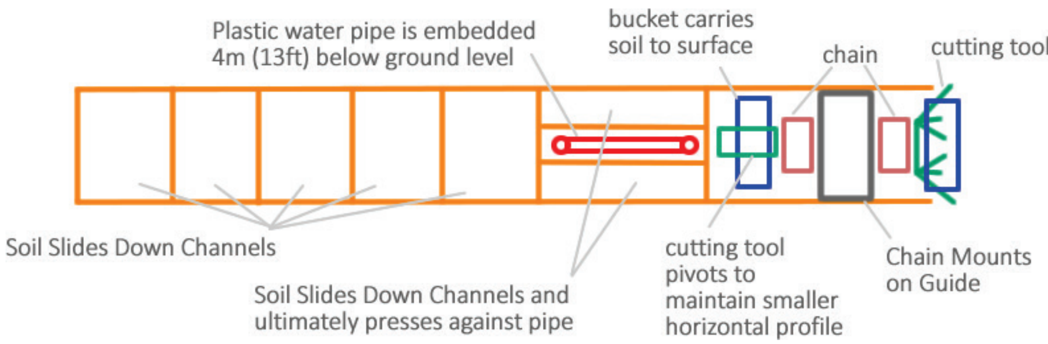
The illustration is a cut-away view. This is what you would see if the mechanism is cut down the middle and it's internally surface is exposed.

The mechanism looks large compared to the truck, and in a sense it is, yet it also fits into a 0.3m (1ft) wide trench and is therefore small in the forward looking dimension.

A chain system cuts soil in a manner similar to a chain saw cutting wood. Teeth at the front cut, and then twist 90° to avoid structure while on their return path.

Buckets on chain move material to top surface, where they mate with a conveyor system that returns material back into the trench, behind our cutting mechanism. This is a closed loop system where dirt is (mostly) not placed onto the ground while the truck moves forward.

Below is a view looking down from the top, under the conveyor.



Channels for soil provide paths for dirt to fall back into the trench, behind our mechanism.

Plastic geothermal pipe travels to the 4m depth via its own channel; and is surrounded by two channels that carry dirt that ultimately press against the pipe.

One of the most important components is difficult to notice. It is at the bottom of the mechanism and is shown in gold color. It is a hydraulic system that presses dirt into the trench at the 4m (13ft) depth, around the geothermal water pipe. This also helps to propel the system forward as teeth cut into the front of the trench.

We also have another hydraulic system above the lower hydraulic system, also shown in gold. Both help to propel the machine forward. Without hydraulic propulsion, tires would spin against the ground. Traditional farm tractors are heavy and have large back tires to support a strong forward force. Yet those vehicles do not drive on the interstate highway at 100kph (70mph). Hydraulics give us tractor like forward force yet with highway capable driving.

Wheel-to-ground traction by itself is not great enough to steer; therefore the two underground hydraulic systems interface to a pivoting surface in order to nudge the trencher a little to the left or to the right. Hydraulics are a key component to this concept vehicle.

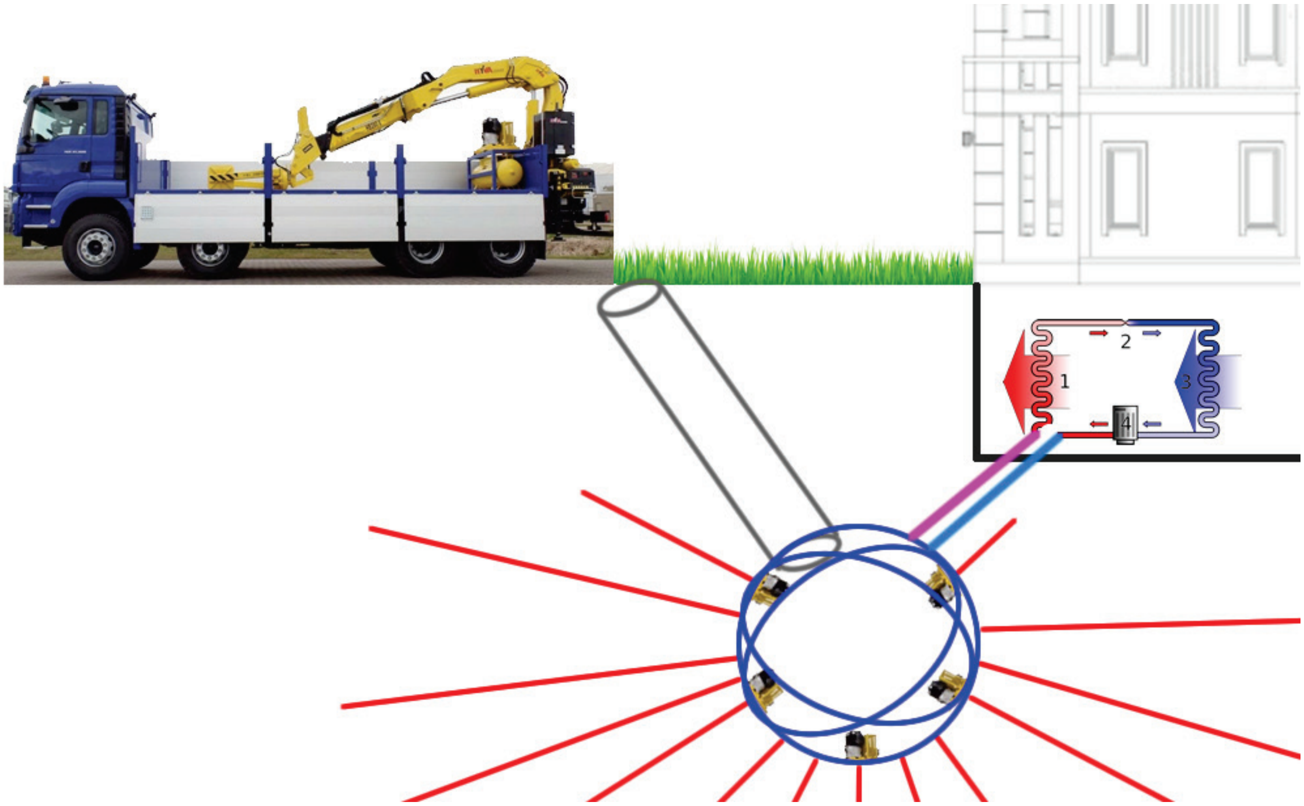
If truck slides and rotates slightly while on a hill then chain mechanism might jam against trench wall. If this happens, can vertical rail pull mechanism out of trench? Engineers need to think about this scenario.

MA2 engineers explore variations of this concept. For example, is it possible to create a mechanism that supports cutting a narrower 0.1m (4inch) wide trench? What are advantages and disadvantages of various widths?

For further considerations on this advance chain trencher concept, click here.

Concept 3: Underground Robotic Gimbal Chamber

MA2 Engineers explore creating a permanent chamber ~6m (20ft) underground that is accessible via a ~0.3m (12") diameter permanent pipe, as illustrated below.



A flatbed truck with robotic arm attaches end of arm to various robots who help to build and maintain system. Truck is parked on street, next to house, and accesses port from that location to reduce damage to yard and sidewalk. One robot drills a 0.3m (1ft) diameter hole with an auger to a depth of 6m (20ft). This permanent hole is lined with plastic (e.g. PVC). Another robot inserts a tool into hole that digs out a 2m to 4m diameter spherical chamber, and installs support structure as needed. High pressure water is sprayed at end of pipe in all directions while dirty water circulates through filtration tank above ground. Subsequently, one builds underground chamber of programmable shape. Gimbal is constructed underground to avoid cave in and to support multiple drilling robots that affix to gimbal. Robots drill holes simultaneously for water pipes that emanated from chamber. When finished installing pipes, they are connected together and routed to house.

If a drilling robot capital cost is $3k and one has five running simultaneously from inside the chamber, then capital cost of $15K is reasonable. If drilling robots are mass produced, their cost becomes very low.

Drilling 300m (1000ft) involves great pressures; however, drilling shorter distances is much easier. A typical house might install 10 pipes, each 20m (60ft) long.

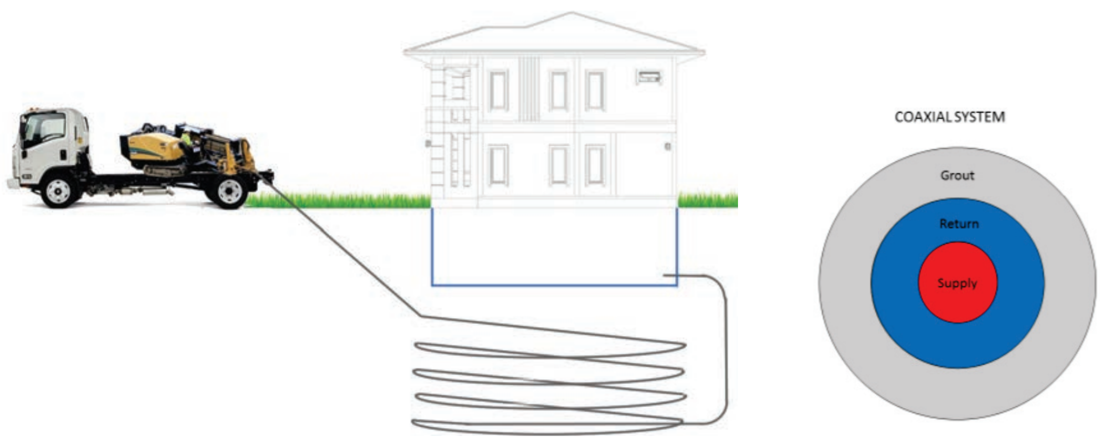
This approach might work well in urban areas where truck on street accesses soil via flower bed at front of building.

For further considerations on the robotic gimbal concept, click here.

Concept 4: Directional Boring

Directional Boring is a common practice that involves drilling with fluids that control direction. The current boring products (no pun intended) do not meet our exact needs, yet it is possible this type of technology could be adapted.

In the below theoretically illustration, we drill, connect geothermal plastic pipe to drill head at its destination position (e.g. basement) and then pull geothermal pipe into hole when drill is retracted. Pipe contains Pipe within pipe (i.e. coaxial design) where the inner part carries outgoing water and the outer carries return. Grout material (e.g. concrete) is pumped into hole to maintain thermal contact between pipe and soil. Truck on street inserts at front yard and makes use of space under yard and house.



Why has this not been done? Several issues including minimum turning radius, drilling fluids under significant pressure might leak into neighbor's property, high volume of fluids, many workers, complicated environmental permits, and significant cost. If a brilliant designer can resolve these issues and others, this technique might have merit.

To learn more about current directional boring techniques, click here.

Chapter 13) Extend Computer Network to Windows, Doors, Ducts & Ceiling

The Last 10 meters

In networking there is a term called "The Last Mile". This refers to how one routes internet data between communications buildings in a city and individual homes. The answer is typically cable modem or fiber. Yet what is more interesting is how one routes power and data to the last 10 meters -- out to things like doors, windows, occupancy detectors in ceiling, fans in ducts and dampers in ducts.

Also, one wants to do this inexpensively and reliably.

The world needs a standard way to physically interconnect devices in a building; primarily for purposes of reducing energy consumption to help justify the cost of the devices. There are several existing networking standards, yet they have cost and reliability issues when working with the last 10 meters; and therefore adoption rates are low.

A PCB with a microprocessor connected to a network is sometimes referred to as one "Node".

The world needs a new networking standard that routes power and data to tiny $1 nodes with 99.999% reliability; and then place these nodes into every window, every door, and every light switch, etc.

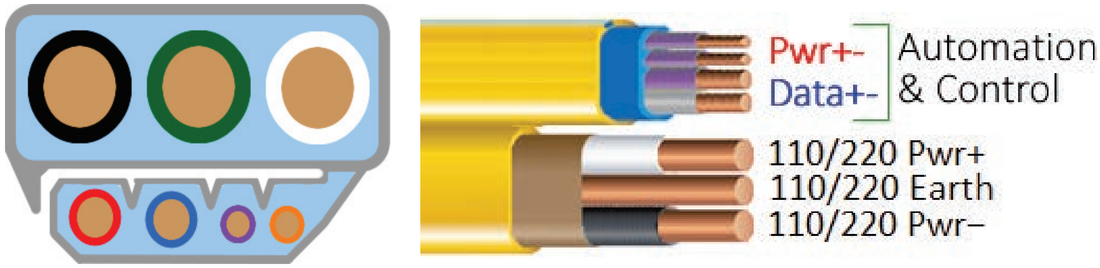
Two Simple Design Goals -- Reliability and Low Cost

Let's begin with several design goals. In order for building automation to be popular, we must meet two requirements: <u>cost</u> and <u>reliability</u>. If the cost per node is low (e.g. $1), then it is easy to justify. This is built into infrastructure that might last 100 years (i.e. a building), and if a building "has issues", it loses value. Therefore the system must be reliable, otherwise building owners will avoid it.

Free Wire to Device

Let's assume devices are in every door, window, switch, light, solar system, water tank, damper in duct and fan in duct. And let's assume we want to route wires to these without spending money. There is only one option, which is to bundle energy management wires with power wires; and pull those from the master fuse box location to each electrical box. Below illustrates the wire. The 3 power wires are 12 gauge (20 Amps), the Automation power wires are 18 gauge (50% diameter of 12 gauge) and the Automation communications *data* wires are 22 gauge (33% diameter of 12 gauge).

New 7 Wire Cable (3 power, 4 automation & control)



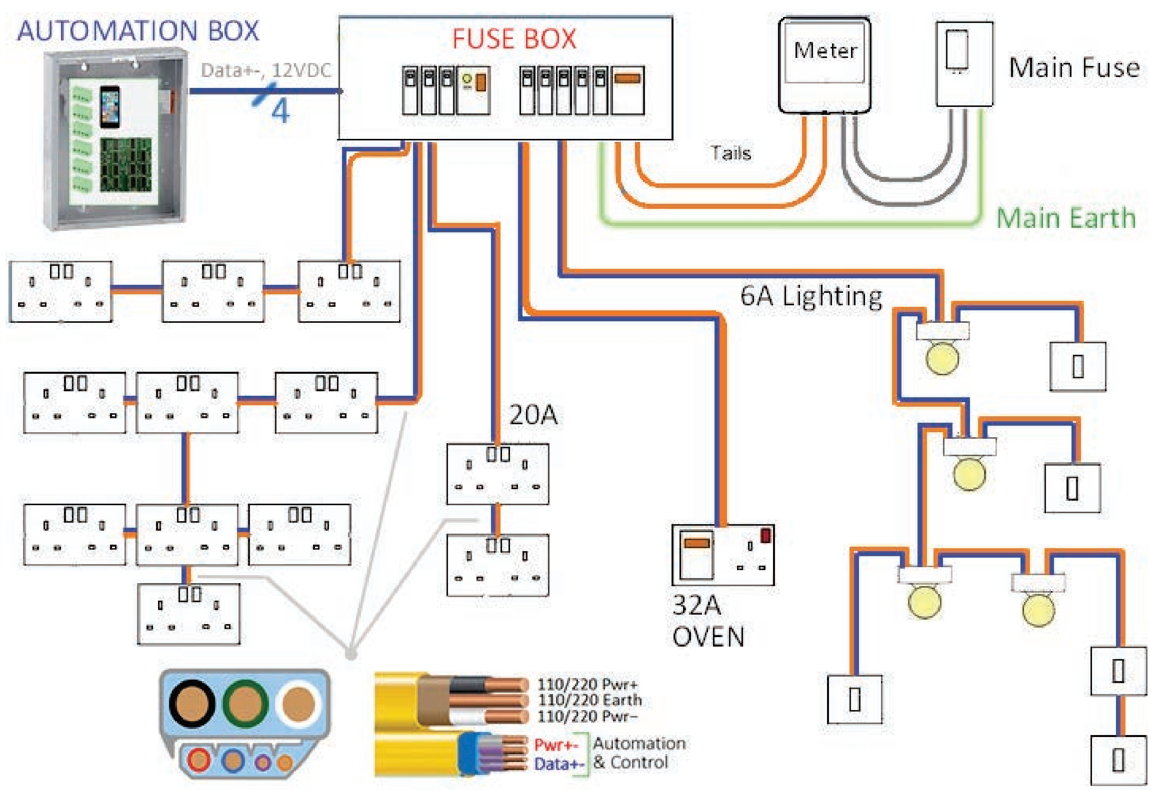
The illustrations to the right show wire diameters drawn to relative scale. The electrician can separate the two bundles with a knife. The 4 Automation wires are in a standard sized jacket with grooves in-between wires so that a hand tool can strip, add connector, or add splice in one squeeze. 1000ft of 4 conductor 18 gauge cable is $200 (e.g. Grainger #6CGV4), therefore adding the additional wires to the bundle is not expensive. Adding a 4 wire cable from all doors and windows to their closest electrical box; and splicing 4 wires at electrical box to facilitate routing involves labor, and this is your primary cost. Cost to wire a new house for automation and control might be $750 to $2000. This is just the wiring, parts and labor, and not the low cost modules that attach to them (i.e. $1 tiny PCB w/ processor).

New Bus

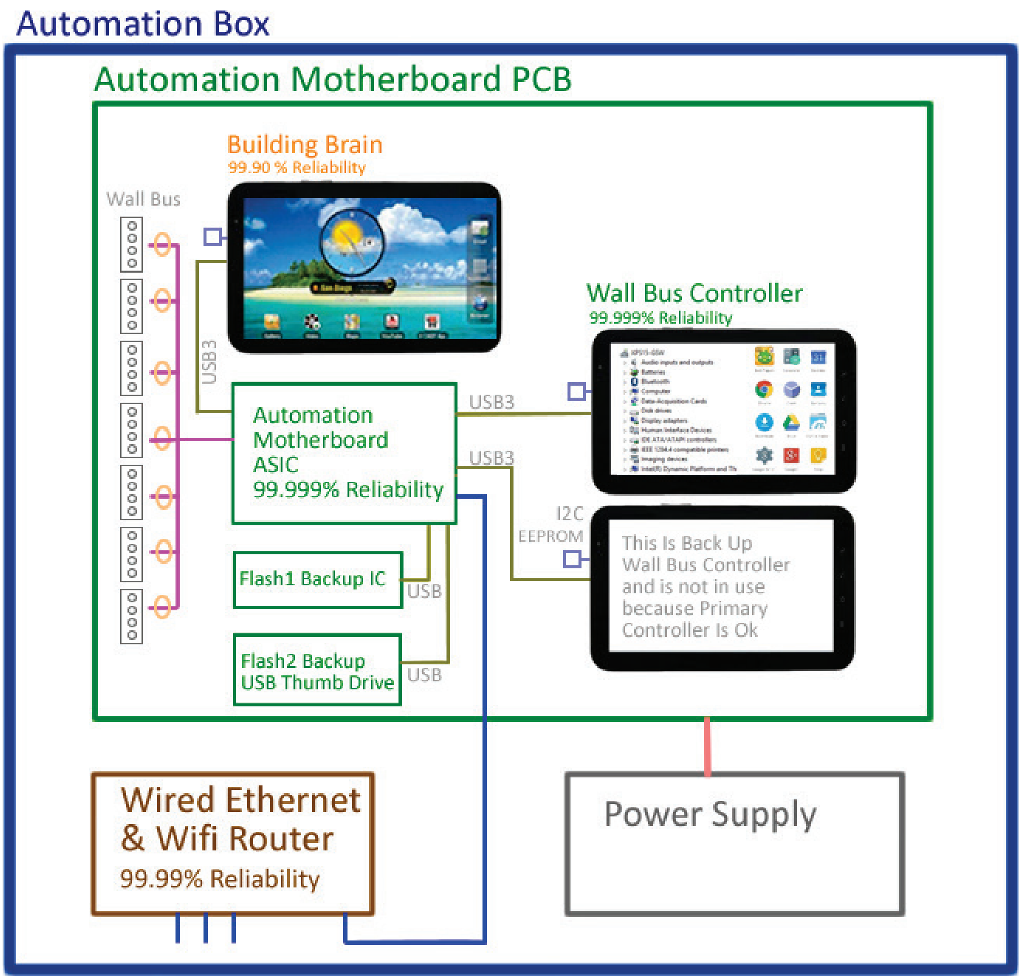
What do we call this new bus? It is in the wall, it is slow like a walrus, and it is a bus. "Wall Bus"?

Typical Building Wiring

In the typical building, power wires route to a fuse box at a central location. Below is an example wiring diagram.



Building Automation Box



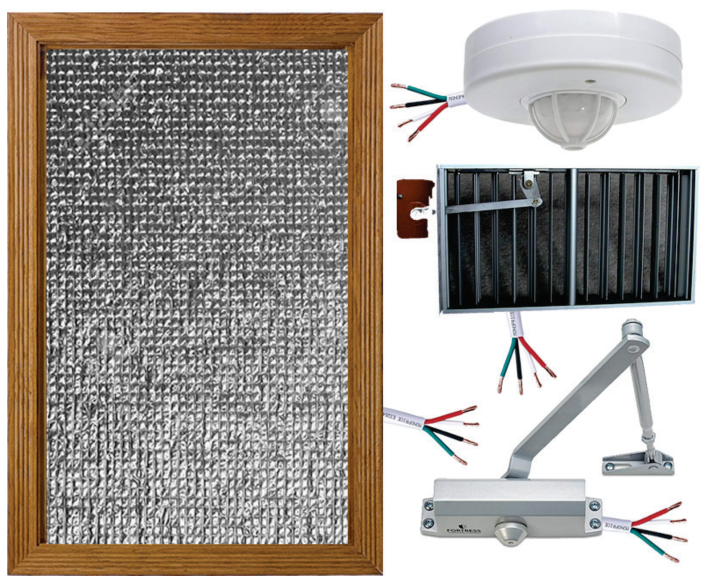
If we bundle our four network wires (Pwr+-, Data+-) with 110/220VAC power wires, for the above reasons, they end up at a central location, in the fuse box. We separate the 4 network wires from the 3 power wires, route the power wires into the fuse box and route the network wires into the "Automation Box". The automation box provides power to our $1 nodes ("devices"), interconnections between devices, a connection to the local area network, and internet access.

Wall Bus Controller

The Automation Box contains a controller (e.g. $50 tablet or smartphone) that manages the 4wire network, interfaces to the local area network (e.g. Wi-Fi, wired Ethernet), and interfaces to the Internet. Tablets contain tremendous System on Chips (SOC) which provide terrific computing power at very low cost; therefore, leveraging off of this technology is reasonable. This device, which we call the "Wall Bus Controller", is stripped down (only install required software) in order to provide reliability, as discussed in further articles.

Brain Thermally Shuts Down Rooms Not In Use

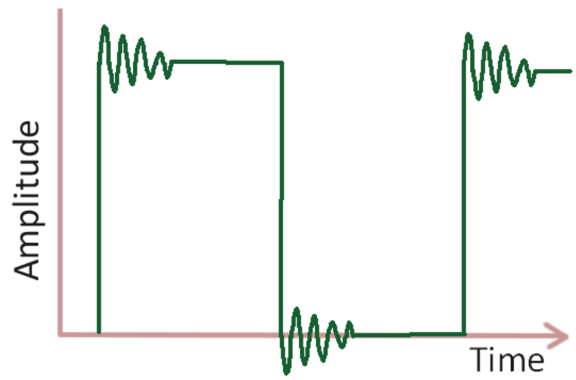
This bus allows us to gain thermal control over every room by closing and opening doors, adjusting window coverings, and adjusting fans and dampers inside ducts. Rooms not in use are thermally shut down. This involves Big energy savings.



Chapter 14) Wall Bus Signaling

The World Needs a New Electrical Interface Standard

We currently have RS-485 which supports communication between 1 to 32 modules that are wired in parallel with 2 wires. For high reliability (99.999%) and low cost ($1/node), we need something similar, yet different.



In many physical networks, wires are modeled as a transmission line and impedance of the line is tightly controlled. This often involves point to point topology or multiple nodes along a line. We are dealing with a Tree topology. Notice in the above home power wiring diagram we see nodes configured more like tree branches than nodes along a line (e.g. daisy-chain). With a Tree, there is no way to control impedances with termination. If one drives a Tree with a fast rise time (e.g. 15nSec), then reflections will cause the signal to ring, which leads to bit errors, as illustrated below.

We Need Slow (for a change)

If one slows down the logic 0 to logic 1 rise time and logic 1 to logic 0 fall time to 1/30th of the time for the electricity to get from one end of the network to the other, then this goes away. In other words, if your rise time is 15uSec instead of 15nSec (1000 times slower), you get a reliable and very low cost connection. Doing this has several side effects:

- Data transmission is limited to 10Kbits/second. This is acceptable since we are focused on simple operations such as reading a sensor or opening a window; not typical data networking. If one wants more, they can add Wi-Fi and use Wall Bus for power.
- Cost is reduced. A typical Rs485 driver pumps 150mA to support driving a cable quickly (e.g. #ST485CDR). This involves tremendous energy which translates into a large chunk of silicon which drives cost. If we slow the rise time 1000-fold, we reduce this dramatically.
- Since the transmitter is slowed down (i.e. bandwidth limited), we can also slow down the receiver (i.e. low pass filter). This is exciting because it means it is harder for noise to couple into the signal wires. Noise often involves switching, which couples a spike onto a nearby wire, yet we reject these with a low pass filter on the receiver.
- If we have a 1M ohm series resistor in front of our receiver we can protect the receiver from being shorted to 220VAC. Normally this does not work because the 1M resistor forms a low pass filter with pin capacitance (e.g. 10pF) and this excessively slows down the signal, yet since we are already slow, this is ok.

In some cases, slow is beautiful. What do we call it? Snail Bus? It is similar to RS-485. Perhaps RS-496?

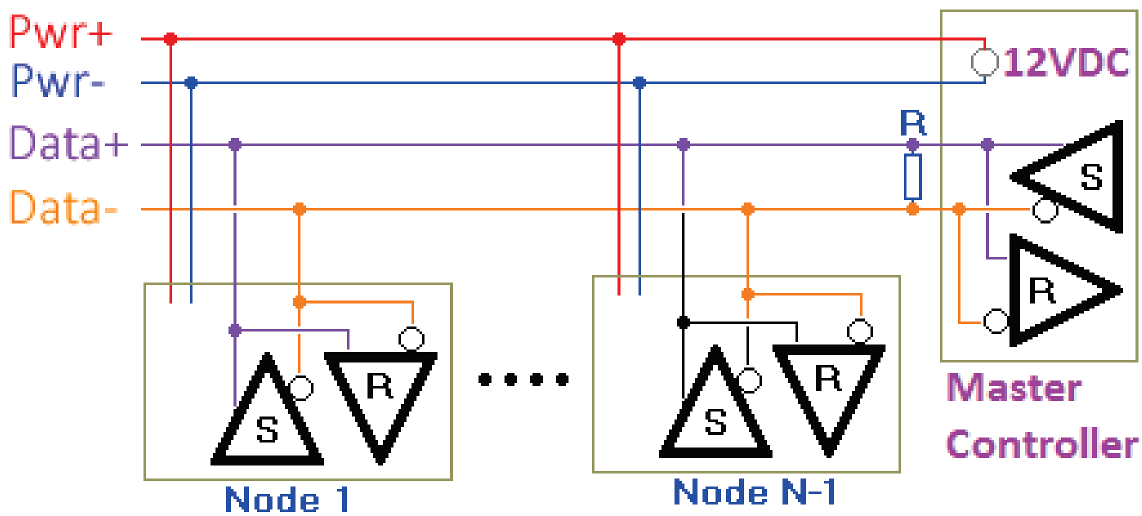
Wall Bus Differential Data Signaling

The 110/220VAC negative wire includes noise (with respect to earth ground) from switching power supplies, dimmers, and microwave ovens (currents on wire involve voltage drop). And these are not controlled. We don't know where they are, how many there are, and how they are changing over time. And we want 99.999%

reliability at low cost. The only option is the above mentioned Rs485-like interface. It transmits 0's and 1's differentially on two wires (data- and data+). A logic 1 is +1.5V between data- and data+, and a logic 0 is -1.5V. For example, 1V on data- and 2.5V on data+ with respect to earth ground is a logic 1; and 1V on data+ and 2.5V on data- is a logic 0. The Pwr- wire will bounce up and down with respect to data- and data+, yet the difference between data- and data+ will be somewhat clean, and it is this difference that is measured by the receiver.

The below figure shows how one communicates with one pair of differential digital wires. This supports communication between any two nodes, or one node broadcasts to multiple nodes. This system can support a maximum cable length of ~100 meters. One resistor at the master controller between data+ and data- reduces activity when the bus is not driven.

One 12VDC power supply in Automation Box routes power to modules via 18 gauge wire, as illustrated below. For more considerations, click here.



Chapter 15) $1 Tiny Nodes on 4Wire Cable

Low Cost Node

In order for the system to be adopted by many consumers, parts cost per node needs to be low and it needs to fit neatly onto a tiny PCB (e.g. 1 x 1 inch). If low cost and tiny, one can afford to have a node in each door, each window, each electrical box, and each ceiling light. Here is an example $1 circuit:

- 12Vdc to 3.3Vdc power supply can be built with two 4.7uF capacitors (50V, 0805, $0.03/each) and one voltage regulator (Az1117EH-3.3TRG1, $0.07/1k); for a total cost of $0.13. If node draws 30mA max and one wants a 2KHz low pass filter on power with a 0.6V maximum voltage drop, then a 20 ohm series resistor ($0.002) before the voltage regulator can help.
- The 16pin Infineon XMC1100 sells for $.58 and provides: ram memory, flash memory, six 12bit A/D channels, digital I/O channels, SPI, counter/timers. ST Micro has a similar part.

The 16pin XMC1100 cannot connect to external ram/flash IC's and it cannot do floating point operations; yet is sufficient for many applications that involve measurement and control. It's A/D is probably accurate to ~ +-0.5%, provided one measures the voltage reference in the factory and stores that in the flash memory in the IC along with a serial number (e.g. IEEE 64-bit Global Identifier).

Wall Bus Devices Communicate via DIF Software Standard

Wall Bus devices use the DIF software standard to communicate, as described here.

Wall Bus in Every Electrical Box

The WEBE (Wall Bus Electrical Box) is manufactured with a Wall Bus PCB embedded in the back wall, as illustrated to the right.

The top surface this PCB facilitates the wiring of the 4 wire bus throughout the building with 4 position screw terminal blocks. The bottom surface of this PCB supports a $1 node processor that can do the following:

- Measure Wall Bus power voltages to look for over-voltage or under-voltage.
- Measure 110/220 power currents
- Measure temperature to detect fire, assist with temperature control, and look for freezing temperatures that might burst water pipes.

The parts cost on the PCB shown in the below photo is approximately $2, therefore a typical home with 30 boxes would have a total cost of $60, which is reasonable. The top of each physical glass window, bottom of each glass window, and top of each internal door one would have at least two 4 position screw terminal blocks to facilitate bussing Wall Bus at those locations.

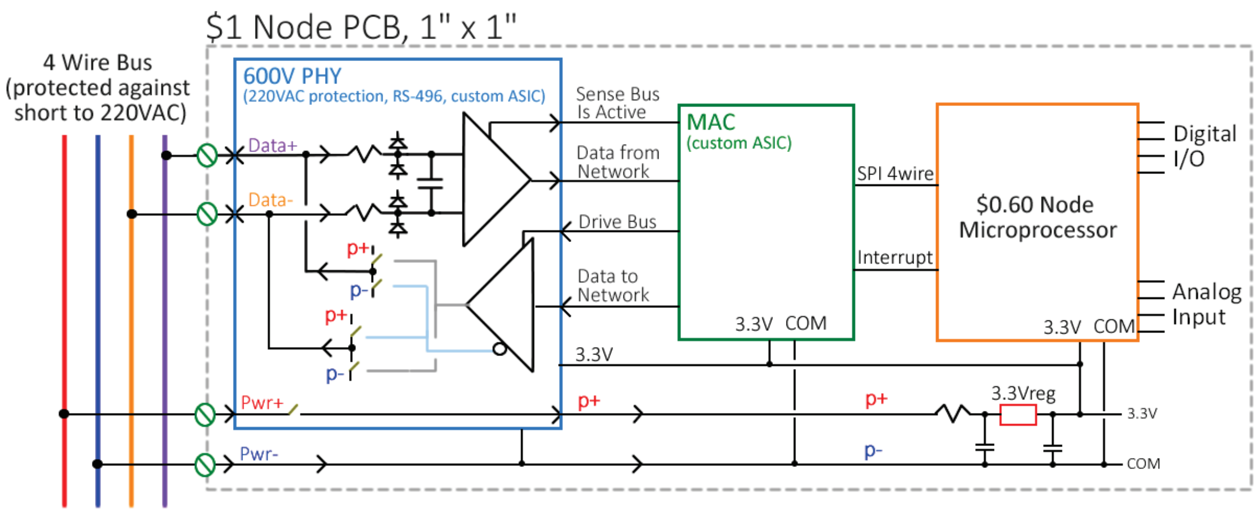
Measure Power Consumed at Each Outlet

The electrical component cost to measure 20Amps at 110VAC is $3. This entails $1 for Wall Bus node support and $2 for an Allegro Current Sensor IC. $150 to measure power consumed at each outlet is not unreasonable in a 50 outlet home. This node could be built into the plastic housing within each outlet and could help homeowner make decisions about what to power on and when.

Custom Protection Device

If one touches a 110/220VAC wire to an automation Pwr+- or Data+- wire for just a moment, it could permanently damage electronics *if* not protected against such a scenario. Subsequently, a low cost custom semiconductor protection device is required. Without this, Jim Pauley would have a fit.

The below illustration shows what this might look like. Our custom 600V PHY (blue) contains five 600V transistor switches; four for driving data and one for protecting Pwr+. One can short any of the 4 bus wires, shown at the left, to 110/220VAC without damage (600V switch opens and disconnects). P+ is protect Pwr+ on node PCB. P- is protected Pwr-, which is attached to Earth Gnd at the Automation Motherboard (and possibly other places).



1M ohm series resistors and protection diodes protect the data input circuit. Normally this is not helpful because ~10pF at the input * 1M ohms forms a 16KHz low pass filter, which is too slow for most applications, yet ok for what we are doing. 300V across 1M ohm is 90mW, which is ok.

Data+- out signals are driven with high voltage transistors and protection diodes that open in the event of over-voltage, preventing damage. In the above illustration, these are represented as switches. The disadvantage of this technique is it forms voltage drops across transistors and diodes; yet we have enough power voltage to afford these drops.

Protecting Pwr+- inputs is more challenging. One can look at this two different ways. One is where the node draws a small amount of power (e.g. 60mA max) and the other is where the node draws much power.

An example low power case would be a 60mA node where protection dissipates 60mW (0.6V for 600V transistor and 0.4V for diode). Most of the time nodes are sleeping and drawing ~0.03W, therefore wasting energy on protection is reasonable.

There could be a version of this custom semiconductor where the power transistor plus diode are external and are driven by this semiconductor. Subsequently, one selects external devices that match the application (e.g. 2W/2A/600V transistor for 0.5Amp motor that opens window).

When Pwr+ is shorted to 110/220VAC, the Pwr+ protection transistor opens and does not draw power. The transistor is only on when we are *not* shorted to 110/220 (normal case). A series diode protects Pwr+ when 110/220 voltage is negative.

The Automation Motherboard PCB supplies ~2Amp power to each 4 wire bus, and one could have many (e.g. 10) of these busses emanating from this PCB (one for each fuse in fuse box). If this PCB disconnects from Pwr+ in the event of short to 110/220, it needs a 2Watt transistor to do this. Or perhaps it shorts 110/220 to GND and blows a fuse?

Pwr- can possibly be "protected" by attaching it directly to earth ground at the master control box, and blowing a 6 to 20A fuse when shorted to 110/220VAC power. The 18AWG Pwr- and connectors need to be able to support 20Amps for short durations w/o damage, which is easy to do.

For more considerations, click here.

In summary, protection against damage, and its costs, deserves additional consideration.

Custom PHY and MAC ASIC's

The above illustration shows a typical node PCB with a low cost microprocessor that communicates with the 4wire bus via a custom 600V PHY ASIC and a custom MAC ASIC. Since the bus is slow, the MAC does not need much memory to store frame data, and the main microprocessor can handle some of the MAC processing with software. For example, a 128bytes buffer would take 128mSec to fill, and an interrupt can be handled in much less time. In summary, the MAC could be simple and easily attach to many different processors. Also, some processor companies might place this MAC inside several of their Microprocessor IC's and charge another ~$0.07 (and use internal RAM for frame buffer). The voltage regulator shown in the diagram could be placed into the PHY IC to save cost. One might be able to combine the 600V PHY and the MAC into one ASIC, yet it is not clear if the 600V silicon would easily support the MAC requirements.

Chapter 16) Create Standards that Automate Windows and Doors

In the past parents told their children "Turn off the *Lights* when you leave the room". In today's multi-zone home parents tell their children, "Turn off the *Room* when you leave the room". In the future, they will tell them nothing, because the room will know what to do. This involves:

- occupancy sensors in ceiling of all rooms
- motor driven thermal blankets on all windows
- motor control over internal doors
- control over numerous dampers and fans inside ducts
- thermal control over each room

New Terms: "Automation Capable Window" and "Automated Window"

We do *not* want window manufacturers to sell both automated and non-automated windows. Instead, we want for *all* windows to be "Automation Capable". Capable means one can add modules from the room to make an Automation Capable Window an "Automated Window". This can be done at any time (e.g. at window factory, building contractor with new construction, building owner at any time).

In order for all windows to be automation capable, the additional cost over a non-capable window needs to be very low (e.g. < $3 parts cost per window). This money goes to screw terminals, wire embedded in the window, and cavities to hold modules. If the cost is too high for "capable", then manufacturers will then offer only a few windows that are automation ready. And these might be in only a few styles. The adoption of automated windows will grow much faster if more models are available to architects and home builders.

New Term: "Window Module"

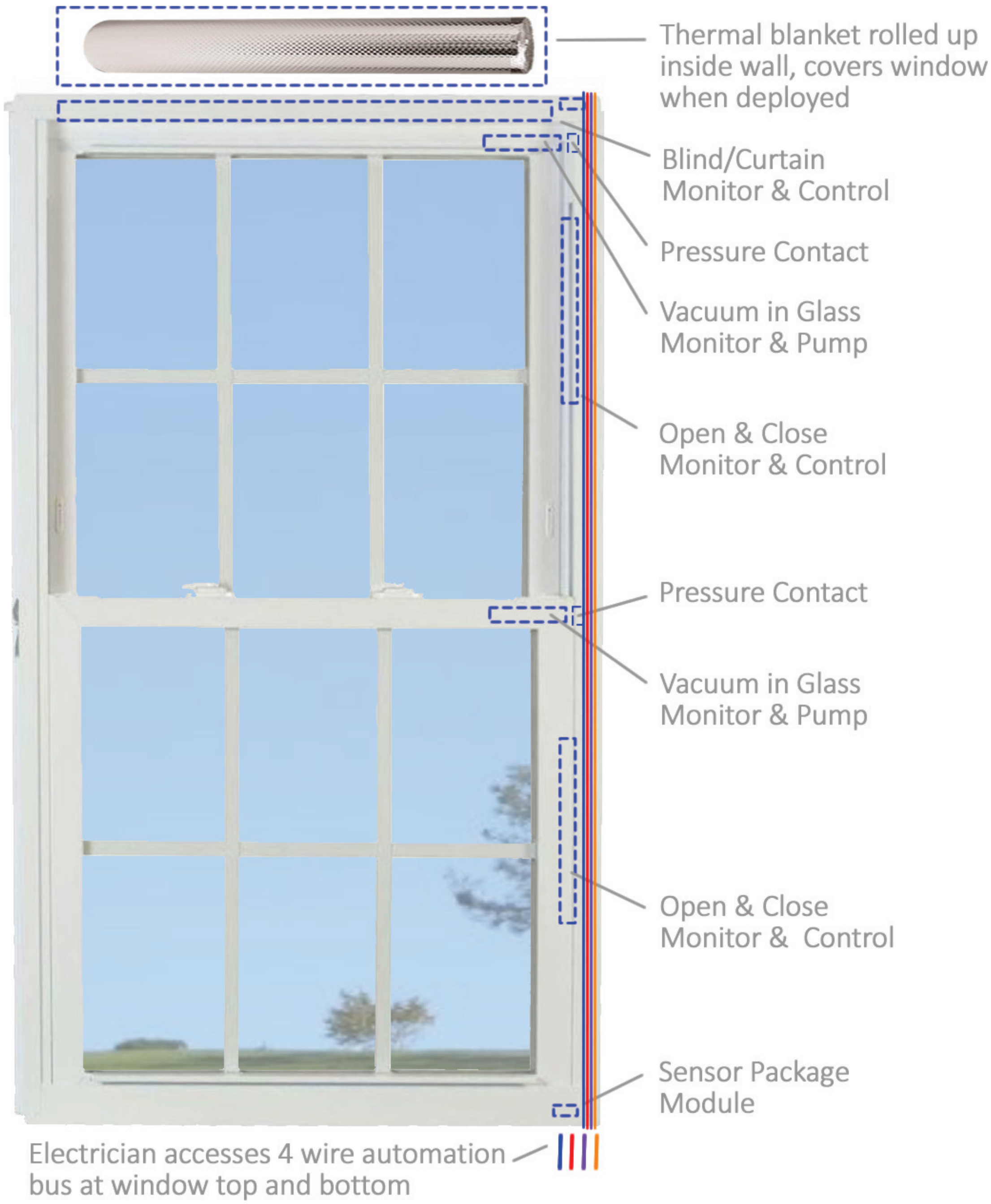
Standard window modules are physical devices that plug into cavities inside an Automation Capable window. Cavities include a cover that matches the surrounding window, therefore the automation capable window looks like a normal window. A connector in the cavity connects the module to our 4wire bus, described in our previous article. The connector includes a 5th wire that routes to base sensor module that helps to identify the position of each cavity.

Plug and Walk

After inserting a module one walks away without further adjustment, setup or software installation. If building Brain is set up to automatically thermally shut down rooms not in use, then it makes use of modules as best possible. If one wants to over-ride this automation, they can do this with manual control from a computer, tablet or phone.

Create Standard for Modules and Cavities in Windows

Imagine a living room with three windows near an electrical box and the 4wire bus is already at that box. To support window automation, the new construction electrician adds an additional 4wire cable between the electrical box and the bottom of each window. The window manufacturer internally further routes this cable to multiple standard sized cavities within the window. As illustrated below, modules that fit those cavities perform various functions such open/close window, measure temperature/pressure, open/close curtain, monitor/control vacuum in-between glass panes, and raise/lower thermal blanket. From an energy perspective, the last item is most interesting.

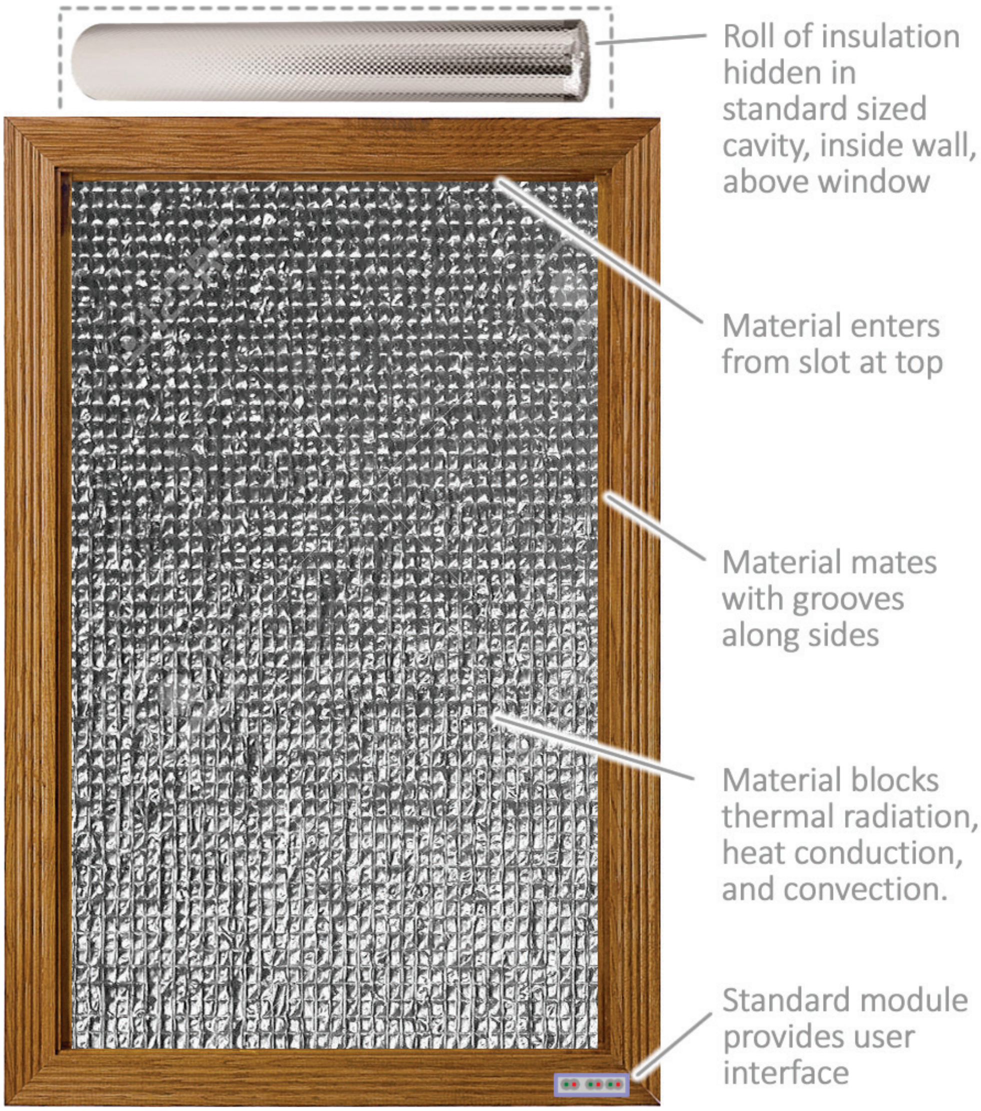


Every window on the Planet should have an Automated Thermal Blanket

Let's be honest. From a thermal point of view, windows are terrible. They involve massive energy loss both from thermal conductivity and from radiated heat; and most of the time they are not in use due to zero occupancy. Heat travels through glass easily and then has much surface area with which to traverse air in-between panes. And warm body radiation easily zips through glass.

The automation capable window maintains a roll of thermal and heat radiation insulation above the window, in the wall, as illustrated below. When lowered via motor in standard cavity, energy loss is dramatically reduced. Grooves in window sides mate with blanket to reduce air convection currents. This significantly reflects sun light, reflects thermal radiation, reduces airflow & reduces heat conduction.

The problem w/ many window coverings, such as curtains, is they look nice at the expense of thermodynamics. For example, aluminum foil is excellent at blocking thermal heat radiation, yet is not visually appealing. The thermal blanket optimizes thermal properties and assumes the home owner is not in the room to complain. However the norms of our society might change and people might become more appreciative of the below ascetics. Beautiful, isn't it?



Smart Blinds

There are two things one can do to optimize the energy performance of a horizontal or vertical window blind system.

1. Coat one surface with a shiny metal material that reflects thermal heat radiation; and point this toward room while heating and away while cooling.

2. Control blind angle with motor such that sun ray is parallel to blind surface when you want sunlight to heat your room (i.e. slowly change blind angle to track the sun).

900Watts per Square Meter with ~100% efficiency

Sunlight entering a room delivers 900 watts per square meter (normal to sun ray) of heat with ~100% efficiency. A typical window, normal to sun ray, is approximately 1m^2, and is therefore similar to a 900W (9Amps at 110VAC) space heater. This is a fantastic amount of heat, and this is just one window. Solar cells are ~20% efficient; therefore you need ~5 times more square meters on the roof to get the same energy (sort of). Subsequently, if it is cold outside, we want sun facing windows to be uncovered.

The Building Brain harnesses this energy as much as possible, and routes it as needed via doors and in-line duct fans (e.g. room to room). Many homes cover windows for privacy; and keep them closed throughout the day due to not wanting to move many coverings twice a day. If desired, the building brain provides both privacy and energy harvest by only covering windows when a room is occupied.

Vacuum-Insulated Window

Vacuum-Insulated Windows (VIW) evacuate air in-between two panes of glass. For example, a 0.01 ATM vacuum reduces conduction energy loss 100 to 1 through the air gap. This is the same concept used by a thermos to keep your coffee warm all day. An example product is the Guardian IG which separates glass by 0.25mm (0.01") and places tiny, almost invisible "pillars" in-between panes to keep them from sucking together and touching. Pillars are required due to 15 Lbs. per square inch of pressure against glass with full vacuum. The narrow gap is good because it means our pillars, if cubes, are viewed as 0.25 x 0.25mm (0.01" x 0.01") squares and are therefore hard to see, especially at a distance. Another advantage of a narrow gap is the glass will not crack if pillars fail, and instead will just touch. The disadvantage of a narrow gap is you get more thermal conduction (e.g. 10x more with 1.0mm gap vs 0.1mm gap)

Our vacuum monitor/control module resides inside a moveable window and interfaces to the window frame via a 4 position slide contact connector. The pump is helpful since a cavity under vacuum slowly loses pressure due to air that permeates through glass. Initially, this module helps the window industry to develop and monitor prototype vacuum in glass products.

See Also: Vacuum Insulation Group

Manhattan 2 Supports Doors

Doors are similar to windows. We connect the 4wire power/data bus to all doors, and internal cavities within doors accept standard modules. Modules do things like open/close door and detect door state. When one thermally shuts down a room not in use, the door must be closed. When one wants to quickly turn on a room, the door is opened and air outside mixes w/ air inside. Door control is required when automatically shutting down rooms not in use.

Manhattan 2 engineers develop a worldwide standard for modules that fit into new construction doors, as well as existing construction doors. New construction involves doors that are manufactured with internal cavities for standard modules; whereas existing construction relies on a chasses installed into the wall by an electrician, above each door.

The existing construction approach is similar to that which is done with existing construction windows: the chassis supports multiple plug-in modules, it is available 1 to 4 wide, it supports 4wire Wall Bus plug-in modules, it is recessed into the internal part of the wall and flush with external wall surface. The chassis supports modules that open and close physical door. If system did not have access to door bolt it might close door slowly until bolt touches plate, yet does not engage. In some cases, an electrician routes wall bus cable from top of hollow door to door bolt, installs motorized door bolt assembly, and interfaces plug-in module to door top.

Motorized Door Bolt Standard

Manhattan 2 engineers develop a standard technique to control motorizing door bolts with Wall Bus. This probably entails routing an internal 4wire cable from the top of the door to the door bolt location. This is easy to do with hollow doors and more challenging with solid. Four electrical contacts at door top connect Wall Bus to door frame, facilitating bolt withdrawal upon command.

Module Design Group (MDG)

Companies will probably not design Wall Bus module products described in this document due to a lack of interconnection standards. To move this forward, MA2 supports ~45 engineers who design, simulate, prototype, test, and write software; all of which is made available on a free gnu license basis.

This includes designing door and windows, their cavities, and devices that insert into those cavities. Also, this includes designing ducts with easy access, and fan/damper products that insert into those ducts. All inserts interface to the 4wire bus and are fully functional. An example is a low parts cost window open/close insert (i.e. motor, gears, and electronics). In year #3 MDG focuses on creating standards with valid reference designs and working software. In years #4 and #5 MDG improves modules, improves software, and solidifies standards.

One could create teams where each consist of one electrical engineer, one computer programmer, and one mechanical engineer. Initially, they could each be assigned an area:

Window

- Base Window Design
- Window Vacuum Monitor/Control Insert (pump, measure vacuum)
- Window Thermal Blanket System
- Window Open/Close Motor Control
- Window Curtain/Blinds Control

- Window Sensor Module

Door

- Base Door Design
- Door Bolt Control
- Door Open/Close Motor Control

Duct Fan/Damper

- Duct w/ Standard Access Panel System (1 mechanical engineer)
- Fan insert w/ on/off control, or variable speed control
- Damper insert

Ceiling Module

- evaluate different occupancy sensors and build prototype ceiling modules

Vacuum Pump

- 3 mechanical engineers and 1 electrical engineer design tiny vacuum pump that consumes 9 to 15V at 100mA.

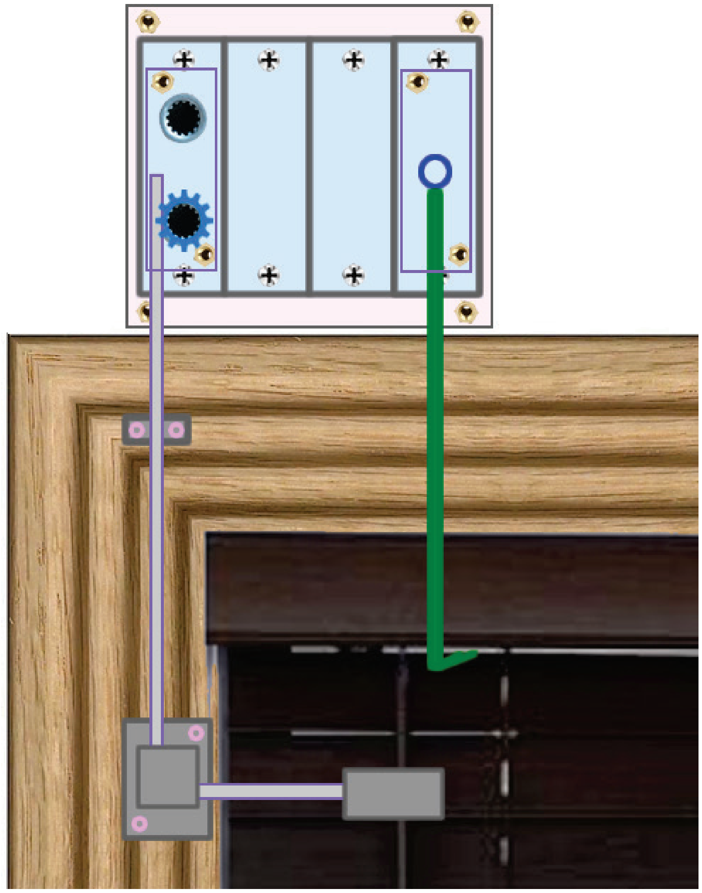
Vacuum Insulated Windows Design and Interface (valve, pressure measurement)

- 3 mechanical engineers and 1 electrical engineer design and prototype vacuum insulated windows with built in interface that monitors vacuum and provides connection to pump.

If properly directed by outstanding engineers, they could do great things with $46M funding spread out over 5 years (5yrs*$0.2M/yr*((12*3)+1+4+4)).

What do we get for $46M? Open reference designs and standards that help create a market for products that provide full thermal control over each room and dramatically reduce energy loss from windows.

Chapter 17) Create Modular Chassis Standard for Windows/Doors/Ceiling

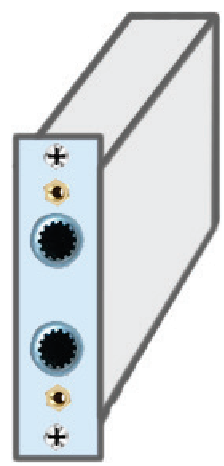


Obviously, one does not have the luxury of internal cavities built into a window when working with existing construction. Subsequently, MA2 creates standard chassis system that is installed into wall above window, into wall above door, into ceiling, or onto outer wall surface.

Standard modules plug into chassis to support things like occupancy detection, fire detection, [Wi-Fi range extension](), curtain/blind motor control, door open/close control, and Alexa/Google Home microphone/speaker (via Wi-Fi).

Each standard plug-in module is ~2.5cm (1") wide, ~7cm (3") tall and ~7cm (3") deep. These plug into a chassis that is 1, 2, 3, or 4 modules wide (1" to 4" wide).

For example, to control all windows, one might place a chassis above each window, possibly installed by an electrician, as illustrated to the right. Installations steps: electrician cuts a rectangular hole in the wall above each window and door, pulls 4wire Wall Bus cable from attic or basement to this location, inserts chassis into hole, affixes chassis to wood stud inside wall, attaches 4wire cable to chassis backplane PCB, and then installs cover plates over each port. A five pin connector at back of module connects module to Wall Bus. The 5th pin helps Wall Bus controller on backplane PCB interact with module in socket.

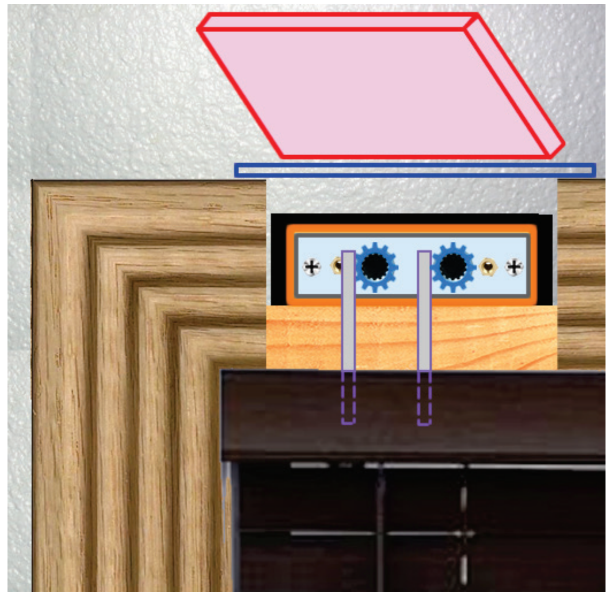


The motor control module, pictured above and to the right, provides a standard rotating shaft that mates with a variety of linkages that interface to existing curtains and blinds. A market for different linkages enables one to connect the standard gear to many different existing window coverings. The example here contains one motor yet two drive shafts for control over two different mechanisms at different times (e.g. curtains, blinds). Threaded mounting holes (gold) support external linkages. Chassis corners, shown above, provide threaded mounting holes as well for chassis cover that hide modules from view.

Also shown above is a standardized connector (blue) for a sunlight detector (green). A variety of sun sensor linkages interface with this connector to support different windows. Knowing when the sun is shining, and angle, is required if one wants to be smart at their windows (e.g. uncover during winter to absorb sun energy). Sun is affected by weather, obstructing trees and buildings, time of day, and latitude.

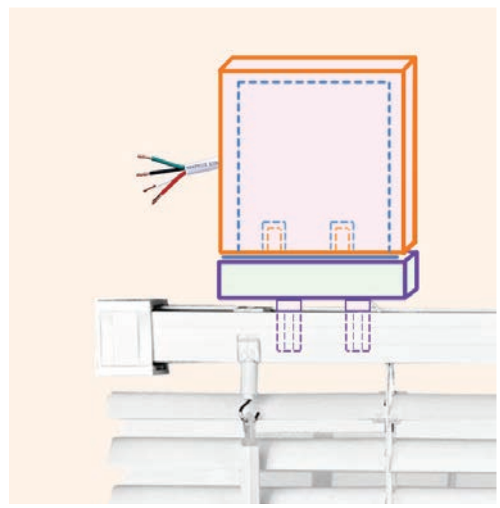
Zigbee manufacturers are invited to participate by producing modules that are powered from Wall Bus 12V wire, yet use zigbee Bluetooth to communicate. Subsequently, homeowners can rely on Bluetooth if they don't want Wall Bus.

Google Home and Alexa are Wi-Fi based, therefore Wall Bus is helpful at providing power and waking a sleeping Wi-Fi processor.



Shown to the right is a module plugged into a 1 slot card cage (orange) which is affixed to the top of the window frame inside the wall. The window installer cuts the molding to build a removable cover (red) that is the same material as the molding; subsequently, the module is typically out view. Linkage (purple) attach module to mechanism. Cover (red) pivots via hinge (blue) which is attached to both molding frame top and cover top.

The illustration below shows a slightly different chassis concept. Here, modules are held in place with thumbscrews instead of Philips head and front panel aspect ratio is 1 to 5 instead of 1 to 3. The right-most unit has a plastic cover which pivots to expose internal modules. It is shown as semi-clear yet actual covers would be opaque.



The above-right illustration shows a 1 slot chassis (orange) that is attached directly to the outer wall surface. Mechanism specific linkages (purple) connect module to curtains and blinds. An electrician routes an internal 4wire cable to module location.

Modules that fit into a chassis are electrically and functionally similar to modules that install into cavities within automated windows (i.e. new construction).

MA2 engineers develop modules, linkages, and chassis products; and install them into test buildings. Eventually, MA2 gathers the "good stuff" and submits it to a standards body organization as a proposed mechanical, electrical, and communications standard for modules that interact with doors, windows, and ceilings.

Chapter 18) Create Standards to Gain Thermal Control over All Rooms

HVAC systems often do not deliver a precise amount of energy to each room, as needed. This is mostly due to one central fan that feeds a tree of ducts that does not cater to precise control over each room. Dampers and in-line fans, shown below, can help, yet routing their wires to a central HVAC system is labor intensive and motor repair is costly due to difficult access.

HVAC engineers have struggled for decades to gain more control over each room without significantly increasing costs. How can our 4wire monitor/control bus help them? Let's begin by reviewing their requirements:

1) After construction, service person easily accesses dampers in ducts and motors in ducts
2) Worldwide physical, electrical and communications standard that defines how to interconnect fan, damper, and control products
3) Dampers and fans receive power and control from nearby electrical box, instead of routing wire from each fan/damper to a central location
4) Low cost for parts, installation labor and repair labor

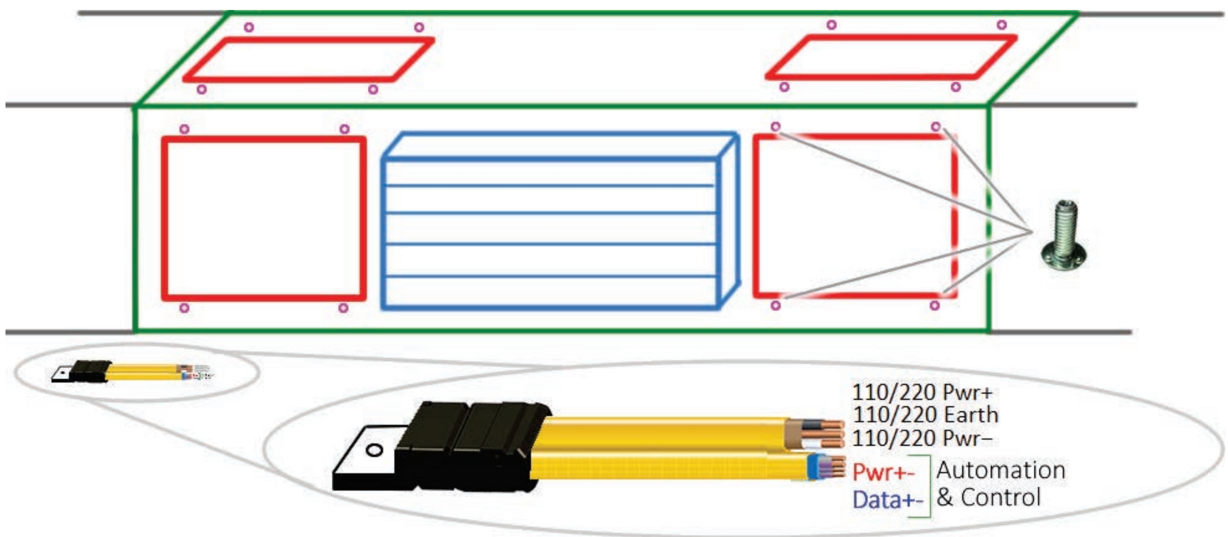
New Terms: "Full Zone", "Full Zone Capable" and "Full Zone Installed"

"Full Zone" refers to full temperature control over all rooms, "Full Zone Installed" refers to a system that is Full Zone and it is fully functional, and "Full Zone Capable" refers to new construction that has the ability to later be upgraded to "Installed". Capable is so cheap all builders do it and defer later decisions/costs to others; or they provide a fully zoned finished product.

Create worldwide standard for Ducts, Dampers and Fans

Manhattan 2 engineers develop a worldwide standard for dampers in ducts, and for fans in ducts that allow easy access and automation. MA2 engineers evaluate many different schemes, including the one presented here. Industry engineers are encourage to participate in the design and prototyping process.

Below is a duct (green) that contains eight standard-sized rectangular access ports (red). These metal plates are held in place with bolts welded to duct wall. Compatible fan and damper products are installed and serviced from the room via the vent (blue), and not from an open wall. In other words, service person can perform all service and installation tasks after construction.



During new construction, electrician routes a power/control stub cable, shown above in yellow, from electrical box to this location and secures end of stub to wall with nail. When one installs a fan, for example, the 4wire bus routes to a tiny $2 PCB inside fan housing that is capable of turning on/off a 110/220VAC device (e.g. via optically isolated Triac #MOC3021S $0.20 and #ACST210 $0.60).

Here is a typical fan or damper installation: remove vent, remove two standard sized port plates, locate 4+3 cable, cut off stub (while power is turn off), pull end of cable into room, attach to fan/damper, replace two metal port plates with custom port plates that interface with fan/damper, push fan/damper through vent and bolt to custom plates.

This is done with both rectangular and circular ducts at important vent locations, and at duct locations where service person has direct access via basement ceiling or attic floor. In the case where access is external (e.g. attic, basement ceiling), one would remove plates via sheet metal screws, insert device, secure to duct wall, replace plates.

A worldwide mechanical standard for access plates, bolts and screws allows HVAC people to finally get the control they need.

Why has this not occurred in the past? No Wall Bus.

Variable Speed Fan Motor

A Variable speed fan motor is easy to implement with a small Wall Bus PCB that attaches to the back of a fan motor. This contains the 4wire Wall Bus interface circuit ($1), a motor controller (e.g. $3 #imc101T), and a motor driver (e.g. $7 #IRSM505-065). For an example circuit, click here and here.

Damper

An inline damper is similar to an inline fan, in that it can direct energy throughout a building. These can be controlled with a stepper motor, which can interface to a $1 node processor (e.g. CMX1100 counter/timer) and a $0.60 solid state relay (e.g. #CPC1002).

New Term: "Ceiling Module"

A standardized ceiling module in every room measures temperature, detects smoke or excess heat, and detects occupancy. This is implemented with Wall Bus power/data wires routed to an electrical box in the middle of each room, and a standard "ceiling module" mounted in that box. Measuring temperature allows the system to intelligently control a variable speed ceiling fan to harvest energy right off the ceiling. Detecting occupancy helps system decide where to focus energy.

Example Scenario 100 Years from Now

While at work you shut down your entire HVAC system with two requirements: (a) keep all water pipes above freezing temperatures to avoid damage and (b) make sure home is comfortable upon return. Also, your bedroom maintains temperature between nighttime 9pm to morning 9am if at home; and the non-bedroom zone is thermally shut down between getting in bed and one hour before waking. While at work the sun is shining and your solar system heats/chills water, pumps energy into the electrical grid, and/or makes hydrogen for transportation. Safer nuclear energy assists as need. Hydro and wind assist as well, yet is only available in relatively few regions. It is very possible this is where we are headed. It is very possible this is the end game.

Living closer to the Edge

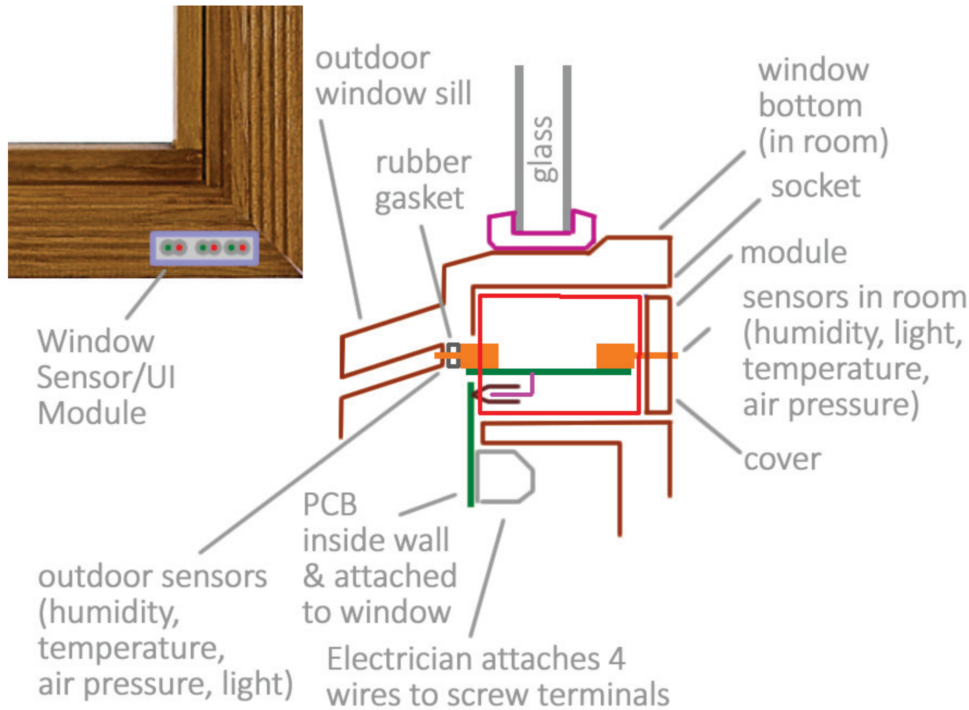
Detecting the temperature near a water pipe inside a wall is helpful for two reasons, one obvious and the other less so. The obvious is we don't want the pipe to freeze; and need to pump in heat if approaching freezing temperatures. The less obvious entails human behavior when setting building temperatures in the winter. If one does not know the relationship between the coldest water pipe temperature and the thermostat set point temperature, they may use a 25°F safety margin to insure pipes don't freeze. In other words, they might set the thermostat to 57°F (32+25) when a zone is not in use. If the temperature inside the walls is better understood, then one can lower that safety margin. Also, with better control over heat going into each room, and monitoring of temperature inside walls, an automated building might pump heat into specific rooms to maintain a reasonable safety margin, instead of the same set point temperature for all rooms. Getting closer to the edge with respect to keeping pipes from freezing can save a tremendous amount of energy, since energy is proportional to difference between outside temperature and inside temperature. For example, an outdoor/indoor temperature of 55 to 25°F (30°F differential) is maintained with twice as much energy as 40 to 25°F (15°F differential). Wall Bus routed to every electrical box means we have a temperature sensor in each electrical box, to help understand internal wall temperatures. If one loses power for 3 days, and pipes are close to the edge, they could freeze. This scenario needs to be taken into account when developing temperature control strategy.

Wealthy Demand Comfort

We have talked about cost and reliability, yet *comfort* is also key. Key because a home owner will switch from automatic to manual if automatic is irritating. Subsequently, the system must quickly prepare uncomfortable rooms when end user is moving toward them. A large tank of hot or cold water used for energy storage can be helpful since it is faster to transport *existing* energy than it is to create energy from scratch and *then* transport. This also means one needs strong fans that move energy quickly to keep occupants from muttering "automation sucks" and clicking "manual". After they click it, they might not click back; therefore, we need to be very sensitive to comfort.

Automatic might save $15 of energy per day ($5K/yr). Many Americans are somewhat wealthy and consider $15 small; therefore, MA2 Engineers must provide comfort or they will click. In effect, wealthy demand comfort.

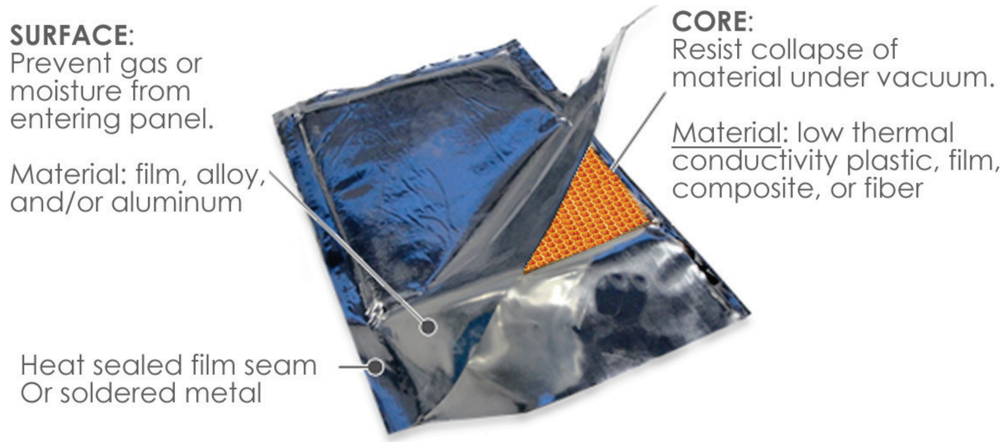
Sensor & User Interface Window Module



In the lower right corner of each automation capable window (new construction) is a 2x5cm (1"x2") cavity for a standardized module that includes sensors and/or a user interface (e.g. buttons to open/close window). This monitors temperature, pressure, light and humidity both inside and outside the building. On a cold winter sunny day, the thermal blanket might lift to allow sun to shine on the floor. If outside humidity is acceptable and outside temperature is more desirable than inside, the window might open to allow air to flow. Or will it? If wind is pressing against window, then air will flow in if opened; otherwise, it might flow out. The pressure sensor helps to determine this. Sensors help to determine how to optimize each window.

Chapter 19) Develop Vacuum Insulation That Relies on Pumps (VIG)

Create Vacuum Insulation Group (VIG)



MA2 forms the Vacuum Insulation Group (VIG) to study different techniques for making vacuum insulation panels (VIP) and tiny vacuum pumps. VIG consists of mechanical engineers, manufacturing engineers, and material scientists. This involves sabbaticals from industry, postdoctoral fellows, and PhD students.

VIG designs vacuum insulation panels that are low cost, mass produced, cut to length in factory under programmable control, and are supported by a small vacuum pump to mitigate permeation issues. This is different from current VIP products that assume a pump is not available, involve higher costs, and degrade over time due to permeation.

Typical panel is heavy aluminum foil w/ internal light honey comb material to keep panel from collapsing when under vacuum pressure. Picture of roll shows what panel might look like while still in factory, before being cut to size & having edges sealed.

VIP Requirements

We ultimately would like a 0.5 to 5.0mm thick flexible material made up of optional permeation barrier, metal foil (probably), insulation spacer inside vacuum to resist collapsing (e.g. light honey comb on ~1cm grid), metal foil, and optional permeation barrier. This is mass produced in a continuous process (e.g. resulting in a roll), cut to size and prepared with features at factory, all under program control. After cutting, factory seals via process that supports vacuum (e.g. crimp, lay down bead of solder/flux at joint, flow solder w/ IR lamp).

Applications include building walls, water storage tanks, refrigerators, and ovens.

Prototype and Test

Many concepts are studied via mechanical simulation. This includes thermal analysis, mechanical properties, permeation (how often one needs to pump), longevity, and cost. Designs with merit are prototyped and tested.

Prototype VIP Factory Production line

Prototypes that perform well are considered for support by a prototype factory production line. MA2 is committed to spending millions of dollars to see if we can mass produce VIP inexpensively. The prototype line supports multiple widths, multiple lengths, and different material designs.

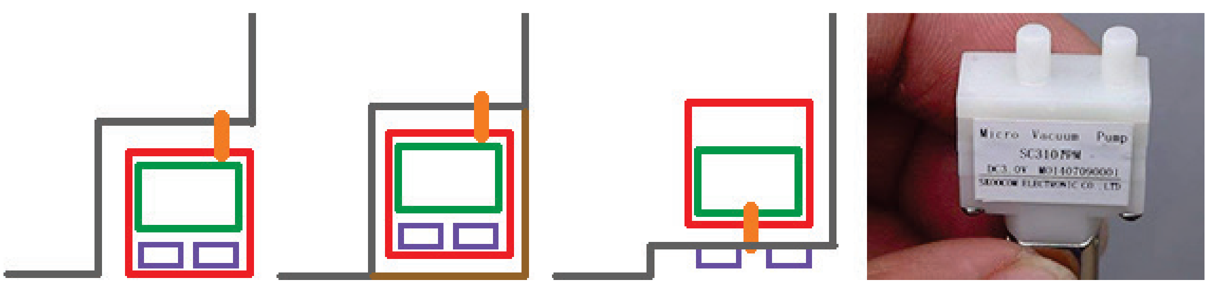
Explore Placement of Vacuum Pump

VIG explores having one vacuum pump support multiple panels via a network of pipes and valves. Engineers do designs and cost models to get a sense as to what is economically feasible. When is it better to have one good pump with pipes and multiple valves, instead of one pump per panel?

Also, VIG looks at placing a tiny pump inside the insulation panel (inside the vacuum), instead of placing pump outside panel with a pipe connecting the two. What are the advantages and disadvantages of the various options? If one places pump inside the vacuum, how much does its materials contribute to outgassing? Does pump in vacuum last longer with less oxidation of internal components?

If pump is small, it can be placed between sheetrock internal surface and room; and wall bus wiring can be placed under horizontal molding. Avoiding internal wall saves cost.

In the below illustration we show several options. The left concept shows a pump detachable from insulation, the middle concept shows a pump outside vacuum yet permanently included with VIP, and the right concept shows pump inside the vacuum itself. Wall Bus node PCB is shown in red, 4wire Wall Bus connectors are shown in purple, and pump is shown in green. The existing pump product shown to the right is similar to our desired pump, yet only pumps to 0.3ATM. The VIG group invests considerable resources to designing tiny vacuum pumps that meet our needs. We have plenty of time to pump, therefore flow rates can be low.



Design Small Vacuum Pumps

VIG engineers design a tiny low cost vacuum pump that periodically pumps vacuum, senses vacuum pressure, and controls on/off valve. Also, they oversee [contests] that design such devices. This is an extremely important initiative since these devices can have a dramatic effect on world energy consumption, especially in regions that are very cold or very hot. One method of roughly measuring vacuum is to place a strain gage on a bending metal plate, where bend relates to vacuum. Four strain gauge wires could then be routed to external wall bus module that measures strain.

Prototype Vacuum Pump Factory Production line

MA2 considers spending millions of dollars to see if we can mass produce vacuum pumps inexpensively. The prototype pump production line supports multiple pump designs.

Design and prototype wall bus module that controls vacuum pump and measures vacuum

VIG and Module Design Group (MDG) engineers design a small wall bus module that controls a vacuum pump, controls valves, and measures vacuum.

NXP and TE both make pressure sensors IC's. An example is the $1.20 MS5607-02BA03 which has a 0.01 to 1.0 ATM range and is accurate to 0.002 ATM.

For a very rough measure of vacuum, one could place a strain gauge on VIP or pump outer wall to detect bending from vacuum, attach strain gage to $0.50 ad8293g160 amplifier IC and to 3.3Vpwr, and measure amplifier output with A/D inside low cost $0.60 processor.

Hot Water Tank

VIG engineers design and prototype at least one hot water tank with VIP and vacuum pump. For example hot water tank wall has thin layer of glass to protect steel (e.g. wesglass), steel wall for strength, zinc plate, 5mm gap for VIP, thin steel outer shell with shiny internal surface to stop heat radiation and protect VIP, 10cm (2.5") of traditional back-up insulation, and painted steel outer wall. For more details, click here or here.

Design Vacuum-Insulated Window

VIG engineers design vacuum-insulated window (VIW) products that place a vacuum in-between panes of glass and are supported by a vacuum pump, as described here.

Design Cooking Oven with Vacuum Insulation

VIG engineers design/prototype cooking oven that utilizes VIP supported by a vacuum pump.

Design Hot Water Pipes with Vacuum Insulation

VIG engineers design/prototype hot water pipes (e.g. 5/8" diameter copper) that utilize VIP supported by a vacuum pump.

Design Air Duct with Vacuum Insulation

VIG engineers design/prototype air duct that utilize VIP supported by a vacuum pump.

VIP References

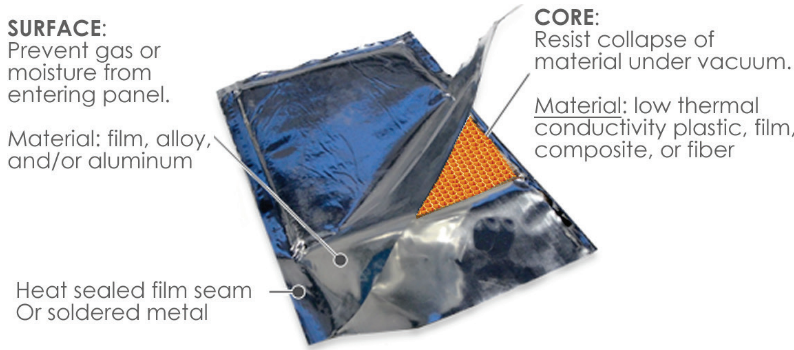
- Standard Specification for Vacuum Insulation Panels
- Lamination and Cladding
- Advanced Wall Group (AWG)
- Electronically Connect Architects to Pre-Fabrication Factories
- Comparing Optimum Barrier Variables of Aluminum and PET foil ...
- Permeation of water vapor through high performance laminates for VIPs...
- Small 0.5ATM pump and Small 0.3ATM pump

Chapter 20) Add Low Cost Vacuum Insulation to Walls (AWG)

Create Advanced Wall Group (AWG)

MA2 forms the Advanced Wall Group (AWG) to study advanced techniques for improving wall insulation. AWG consists of mechanical, manufacturing and computer engineers who focus on adding vacuum insulation to walls. This group consists of experienced engineers on sabbatical from industry, postdoctoral fellows, and PhD students.

A different MA2 group, called "Vacuum Insulation Group" (VIG), creates a factory production line to mass produce low cost vacuum insulation panels (VIP) and low cost pumps that are connected to Wall Bus. The Advanced Wall Group (AWG) focuses on installing these products into walls.



Typical panel is heavy aluminum foil w/ internal light honey comb material to keep panel from collapsing when under vacuum pressure.

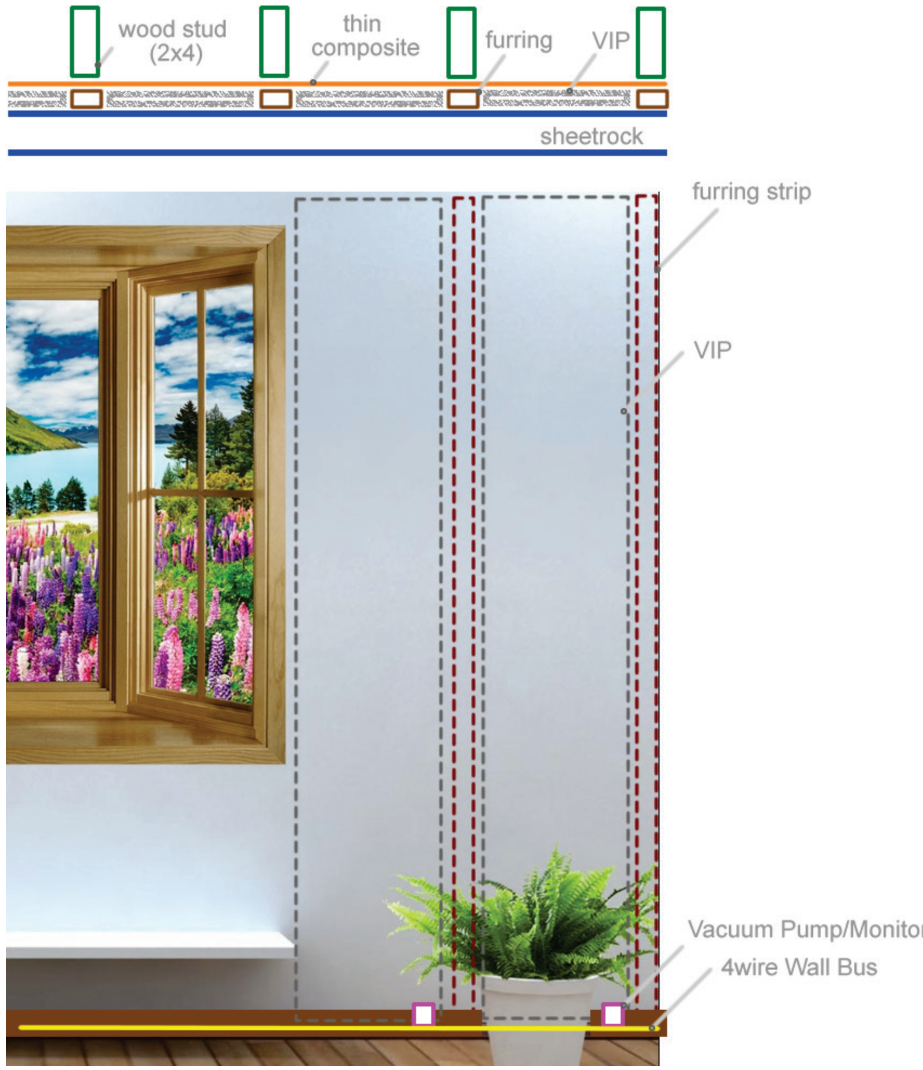
Adding VIP to Smart Solar Material Makes Sense

The additional cost of VIP added to smart solar material within one layer is low, as noted here.

VIP in Internal Walls?

Vacuum insulation between internal *rooms* reduces heat flow and reduces sound conduction. Sound does not travel well through a vacuum; therefore internal VIP can be used instead of, or in conjunction with, acoustical insulation board. Also, internal thermal insulation is helpful at maintaining different temperature zones.

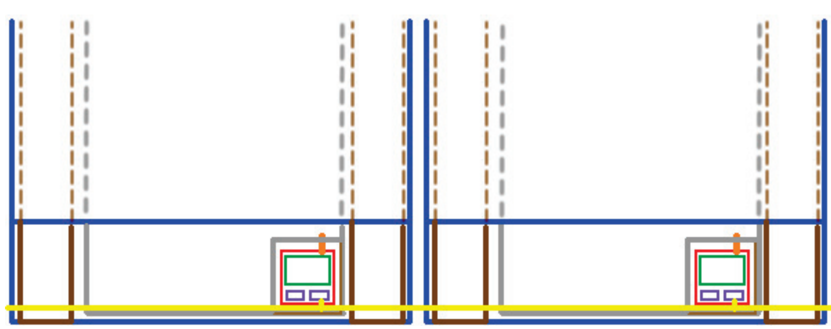
Add Low Cost Vacuum Insulation to Internal Walls



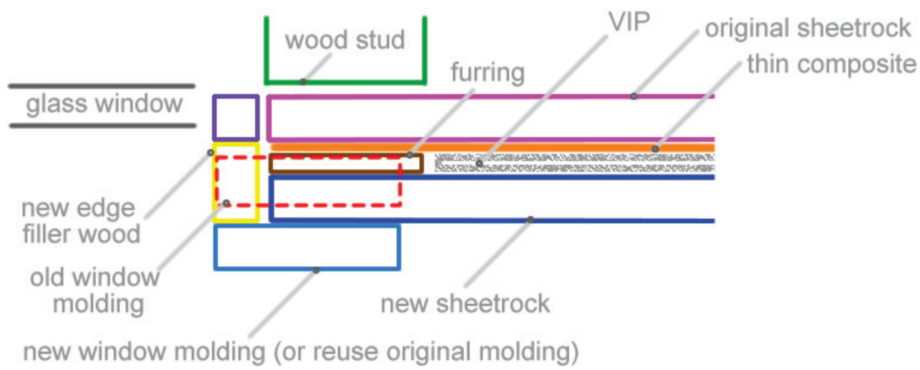
What can we do with Wall Bus everywhere and small $3 vacuum pumps? AWG engineers study multiple concepts, including the one shown to the right which places 14.0" x 0.3" vacuum insulation panels ([VIP](#)) at 16" intervals directly behind sheetrock.

Flexible [vacuum insulated panels](#) (VIP) slide into cavities from a base location, under the lower (or upper) horizontal molding. Molding is held in place with screws; therefore one can easily install, maintain, and replace panels, pumps and wiring. Evacuated VIP panel (gray) can be thin (e.g. 0.25 to 4mm) with small internal insulated spacers on ~1x1cm grid to reduce collapsing. VIP wall might be a heavy aluminum foil, possibly coated with a plastic film. [Furring](#) strips (brown) are made of a strong thermal insulator (e.g. plastic honey comb w/ internal air pockets). Small vacuum pumps (green) are connected together with Wall Bus 4 wire cable (yellow). A thin 1 to 3mm composite material (orange) protects VIP back surface.

The below illustration shows the bottom of two vacuum insulation panels after removing the lower horizontal molding. The VIP's (gray) are attached to small vacuum pumps (green), which are connected together with a 4wire Wall Bus cable (yellow).



Add Vacuum Insulation to Existing Construction



What can we do with existing construction and how much does it cost? AWG engineers study multiple concepts, including the one shown to the right. This is a top view of the window to wall interface. In this scheme, molding around window (dotted red) is removed, new sheetrock + VIP + thin composite wall is added (blue), edge treatment molding (yellow) is added, and new molding (light blue) is placed around window (or reuse original).

Advanced Wall Group (AWG) creates drawings, calculates thermal models, calculates cost models, and publishes their work for this scheme and many others.

The AWG consist of a small group of mechanical and manufacturing engineers. Cost models produced by manufacturing engineers are extremely important. If a scheme is deemed costly, researchers either mitigate cost drivers (rework things that are expensive) or move along to a different scheme.

What does it take to Make Wall VIP a Reality?

To implement wall VIP at low cost, we need:

- Factory creates prefabricated custom walls (sheet rock + VIP + furring + thin composite wall), as defined by architectural drawings. Each VIP panel is made to a programmable length and each wall panel is prepared with custom cut-outs and geometry. AWG engineers do a preliminary design of a Prefabricated Active Wall (PAF) factory, and create cost models for material produced, and cost of the factory production line.
- MA2 Architectural Data Foundation (ADF) creates standards that connect architectural software to prefabrication factories.
- MA2 Vacuum Insulation Group (VIG) designs VIP material manufactured in a continuous process (e.g. resulting in 14" x 100meters per roll), cut to size and prepared with features at factory, all under program control from architectural drawings. After cutting, factory seals via process that supports vacuum (e.g. crimp, lay down bead of solder/flux at joint, flow solder w/ IR lamp).
- MA2 Vacuum Insulation Group (VIG) designs tiny low cost vacuum pumps.

Prototype Radar to Laptop to Wall Projection System

AWG engineers connect laptop to radar, connect laptop to projected display, and project internal features onto wall. This system is demonstrated and software is made available to the public. Cost models are created for a factory manufactured version of this system, and labor savings are measured. What can be projected onto the wall? Can this be used to drive a robot? AWG explores these questions.

Create Wall Characterization Description Standard ("Wall String")

AWG engineers create a proposed standard, called "Wall String", which characterizes insulation and walls based on values that are represented in one string of text. Each material is given a 2 letter material code (e.g. "sr" refers sheetrock) followed by a thickness value in 0.1mm units. For example, "my3al1sr190" refers to Mylar 0.3mm + aluminum 0.1mm + sheetrock 1.9cm. Or "my3al1ai10sr190" refers to the same thing, yet 1mm air gap ("ai") between aluminum and sheetrock. One combines component strings to form one long string that represents entire wall.

AWG engineers create website that accepts these strings and displays webpage with cross sectional drawing and thermal properties. For example, "www.building.org/wall?q=my3al1ai10sr190", displays drawing of above described wall. One enters temperatures on both sides to assist in calculating W/m^2. Webpage shows temperature at each layer and illustrates path of radiated and conducted heat so viewer can better understand system. Webpage provides suggestions. For example, "click to add 1mm air gap between hot shiny aluminum and cold surface". System plots W/m^2 vs time to show how materials age over time.

Currently there exists terrific professional software that provides many capabilities described here; however, the public often bases decisions on one number, R-value. The proposed system enables the public to obtain thermal wall modeling capabilities previously only available to professional architects.

AWG engineers produce software that manages material string codes, creates thermal models for components and for systems (i.e. multiple components in layers). This software is made available to the public on a free gnu license basis in C#, C++ and Python.

Industry and gov't leaders consider requiring manufactures of wall products to display these codes on their product literature website. Customer can then obtain W/m^2 data when product is installed into a specific wall. Also manufacturers are required to open "www.building.org/wall?q=DescriptionString" when one clicks on that code.

Wall String also supports other building products such as vacuum insulation panel (e.g. metal foil, vacuum, metal foil), water tank wall (e.g. zinc, steel, glass coating), and glass windows (e.g. glass, argon, glass). If buyer knows product lasts much longer for small additional money, they will often spend. Yet how do they know? Wall string descriptors and reputable websites that interpret them assist in making good choices; and spending money to reduce energy consumption.

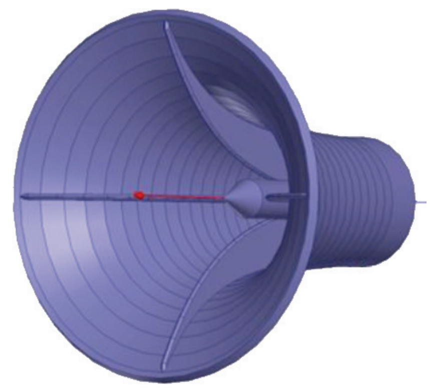
To assist in evaluating products, one of the codes (e.g. "pr") is a product code that is followed by a number that specifies the type of product that is being described (e.g. vacuum insulated panel, vacuum insulated glass, hot water tank internal wall, etc.). For example, if VIP product code is 24 then "pr24al2va20al2" refers to VIP with 0.2mm thick aluminum wall, 2mm vacuum, 0.2mm AL.

Chapter 21) Create Advanced Microwave Electronics Group (AMEG)

MA2 forms the Advanced Microwave Electronics Group (AMEG) to study advanced techniques for seeing into walls and into the ground. This group is based at a university that is strong in this area and consists of several electrical engineers who are familiar with microwave electronics. Current products that peer into walls and into the ground are deficient and need to be improved in order for a variety of Manhattan 2 robotic installation techniques to be cost effective (robots need to see well). AMEG engineers experiment with different antennas, antenna arrays, and frequencies.

Applications include: Ground penetrating radar to help with automated installation of solar on steel panels and framing (e.g. see into ground before drilling with auger); ground penetrating radar to see rocks and roots before installing geothermal GSHP pipe, internal wall radar to view wall before gut rehab (e.g. does sheet rock need to be removed to replace damaged insulation or is internal wall sound?), external wall radar (e.g. box attached to RCV arm) to look for damaged insulation (e.g. due to water in wall).

Portable System with Large Unfolding Antenna



Explored are portable antennas that unfold to create larger structures that support lower frequencies (i.e. that penetrate better). An example might be a 1.5meter diameter (5ft diameter, 50MHz) horn, pictured below, that unfolds like an umbrella (possibly inside a room) to form a metalized film surface accurate to 0.5 cm. Skin depth of 50MHz onto aluminum is 11um, which is comparable to plating thickness.

Perhaps a network of internal plastic braces control shape? Perhaps this moves on rail mounted on tripod while radar is scanning?

Large System Controlled By Robotic Construction Vehicle (RCV)

A Robotic Construction Vehicle (RCV) mechanical arm supports multiple attachments. What can one do with a large 8 x 8 x 5ft box that contains an array of multiple quad ridge feeds (above illustration) that are each 1.5m in diameter? Or one antenna inside large box on rotating platform that supports multiple positions? What is a low cost method to manufacture large metal horn? Metal spinning?

Signal Processing Research

Engineers explore various techniques for recording, including removing echo of close objects via analog subtraction (not digital) to better see further objects. This entails digitizing first with 16bit A/D to determine echo of close objects, and then recreating that with a 16bit D/A. The *analog difference* of the 16bit D/A signal (echo of close objects) and the antenna feed signal is then fed into a 16bit A/D -- to improve dynamic range of *further* objects. Can one continue to drill down? If so, how much?

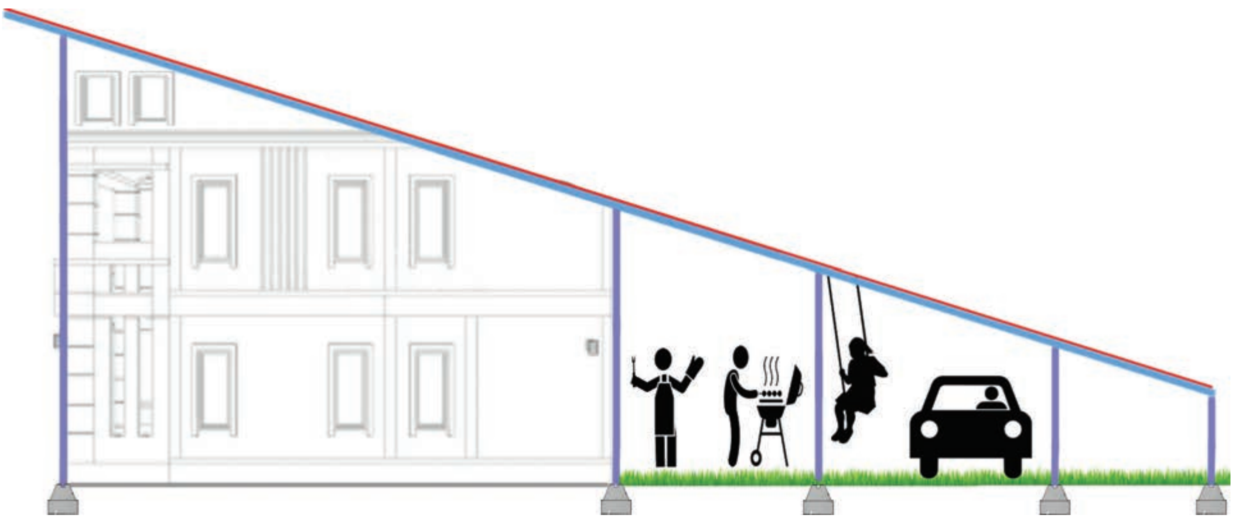
Electrical Engineers Accept Help from Brilliant Mechanical Engineer

The AMEG group contains at least one outstanding postdoctoral fellow in mechanical engineering to help design low cost and easy to manufacture innovative antennas. It is possible the mechanical engineer is the most important member of the MA2 microwave electronics group.

Chapter 22) Create Zero Energy (ZE) Architectural PhD Program

MA2 encourages architecture schools to set up a PhD discipline called "Zero Energy" (ZE). This entails designing and prototyping buildings that use (close to) zero energy; with less emphasis on aesthetics. Pumping energy into the grid and then pulling it out later is allowed. ZE students utilize PV solar, geothermal heat pump, tanks of water for energy storage, vacuum (VIP) insulation, and thermal control over doors/windows/rooms. Manhattan 2 encourages at least one school in a hot climate (e.g. Rice University, UT @ Austin) to participate in order to test ZE concepts in hot climates.

An example hot ZE concept, illustrated below, would be a 200 square meter (e.g. 13x16m, 40x50ft) 28kW solar array made of corrugated steel panels on aluminum framing. This replaces traditional wood roof and sits above house, cars, and patio. Can one do this at less cost than traditional wood roof via automation?



Comprehensive Solar Metal Panel Coverage (CSMPC) Is Exciting

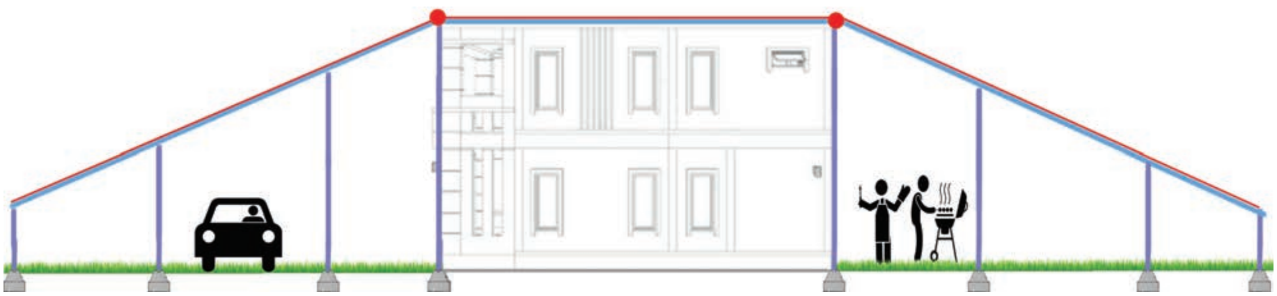
Many things happens when rain water and direct sunlight does not interact with walls and top floor. Wall external surfaces become less costly, walls look new 10 years after construction, paint/caulking lasts longer, there is less insulation damage due to water, less money is spent on paint and shingle roofing, and one can coat external surface with beautiful varnished wood without significant UV degradation. ZE architects figure out how to reduce cost of Comprehensive Solar Metal Panel Coverage (CSMPC) in an effort to drive its adoption.

Pivoting Solar Panel

In the below concept, we have two pivot points (red dot) for wing panels to pivot upward to better capture sun at an angle. Wing panels are very light since we want them cheap. Subsequently, they only deploy when it is not windy. If one loses power and wind presses sufficiently on deployed panel, it goes all the way to the left or right, and locks down (instead of flapping in wind and ultimately ending up in neighbor's bushes). Wings consist of 30x8ft corrugated steel (or AL) panels that install via crane from stack on back of flatbed truck.

Wind force directly against a 30x8ft panel is 4K LBs with 60mph wind and 16K LBs with 120 mph window; therefore engineers design structure to never directly fight big wind via multiple techniques.

If we have 30ft to left, 30ft to right, and a 30ft x 30ft house roof; then total solar surface is 2700 sq ft (250m^2, (30+30+30)*30). This produces 31kW (13*250/100) assuming sun is at a 45degree angle, on average. Questions for ZE student: "Can you figure out how to implement this pivot business at low cost and have it last 50 years? If this produces $6k of electricity each year, what is payback period?"



The Marriage of ZE and MA2

MA2 engineers initially create gadgets in prototype form; and make these available free of charge to ZE architects who deploy them in test cases (e.g. test wall or test house).

ZE students work closely with MA2 engineers. They need each other. The ZE student needs the new gadgets to do great things, and the MA2 engineer needs to know what the architect is willing to deploy.

Many ZE Architects are Engineers

ZE schools are encouraged to have at least ⅓ of program students with a degree in computer science and at least ⅓ to have a degree in mechanical, civil, structural, or manufacturing engineering. Designs are more feasible if students are familiar with, and get involved in, programming and engineering. For example, a mechanical engineer is needed to insure the above pivot mechanism is both low cost and long lasting. Also, a computer scientist could create a web page calculator that models the above concept given several input parameters; to help architects worldwide better understand this approach.

ZE Street

Architectural schools with a ZE program are encouraged to set up a ZE Street nearby that features exclusively low cost and long lasting ZE structures. We don’t want to encourage society to waste money or build non-durable structures; therefore concepts are fully simulated and cost modeled before being built. A strict review process ensures that only quality structures are presented. Local architects, trades people, and general contractors are invited to view ZE Street. Also, this is designated as a "Full Disclosure" zone since we want decision makers to have accurate information.

The Holy Grail of ZE

If manufacturing and robotics engineers sufficiently reduce the cost of corrugated steel solar on metal framing, then people with land will have incentive to deploy. Imagine the impact this would have on the world. The key to making this happen is [Automated Solar Installation](). And ZE students with a computer science or engineering degree are in an excellent position to assist in the development of this technology.

Chapter 23) Electronically Connect Architect, City, Suppliers, and Robots

Create Architectural Data Foundation (ADF)

Manhattan 2 forms the Architectural Data Foundation (ADF) to electronically coordinate architects, cities (who hold data files), pre-fabrication factories, general contractors, and robots.

This group consists of computer scientists, manufacturing engineers, robotics engineers and a few architects. Less than 10% of this group are architects since this group focuses on software development. Their headquarters is placed near a school strong in these areas and all work is made available to the public on a free gnu public license basis. ADF consists of industry engineers on sabbatical, postdoctoral fellows, and PhD students.

Electronically Connect Architects to Pre-Fabrication Factories

Architects are less inclined to work with pre-fabrication due to less flexibility and less control over final product. ADF engineers study ways of converting architectural files from different software programs to a standard language that drives factory machines who create wall sections built to order. Also, ADF engineers create a standard language by which factories provide feedback to architectural software. Factories typically produce a slightly different product, and their changes must be represented on original architect's computer screen in order for them to be comfortable with process. For example, a popup menu on architect's computer selects specific factory, and for each, sees modifications in red color, price and thermal properties. This enables architect to select factory, control details, and reduce cost.

In order for this to work ADF engineers create a standard language for "discussing" features within an architectural data file (e.g. DXF).

For example, command is sent from architect computer to pre-fabrication computer that says "what is cost to pre-fab wall W53 give DXF file of entire building, and what changes would you make?" Pre-fabrication computer isolates W53, redesigns based on machines, and transmits back to architect.

Electronically Connect Architect to Smart Solar Factory and to Roof Robot

Smart solar strips are fabricated at a factory and then unrolled onto a plywood roof by Roof Robot (e.g. 7strips each 1x10m cover 6x10m roof). ADF writes interface software modules that connect original DXF file, solar fabrication factory, and roof robot (e.g. place strip #1 at location {x1,y1}). Software also calculates dimensions and features for each strip, given roof design.

Electronically Connect Owner to City and to Suppliers

ADF Engineers create a test bed that includes city website that maintains original architectural drawings, building contractor tablet, architect's software, robots, and pre-fabrication factories. Each of these components involves a software interface module that is written by ADF engineers. Data is passed back and forth in a standardized and secure manner as defined by ADF. ADF works with common file formats such as DXF, DWG and DWF; and supports conversion between these standards.

Here's an example application: Home owners physically appears in city deeds office to receive a paper that contains access code to their drawings. With this code, home owner visits supplier websites to automatically obtain prices for various parts and services based on drawings. For example, homeowner visits homedepot.com, submits DXF access code, views existing house, clicks on one house window, and views list of replacement windows that fit existing frame.

ADF Writes Software That Describes Modifications To Existing Construction

ADF programmers write software that creates drawings that describe how to improve existing construction. For example, software creates drawings for electrician that show how to install wall bus wall bus chassis above windows.

ADF Focuses on the Public's Interests

ADF engineers write software and create test beds; and are not responsible for adoption. However, if well received, their work eventually evolves into a standard.

ADF makes all material available via free gnu license.

MA2 is interested in ADF because ADF reduces installation costs of energy saving technologies via automation, and ADF improves decisions that reduce energy consumption.

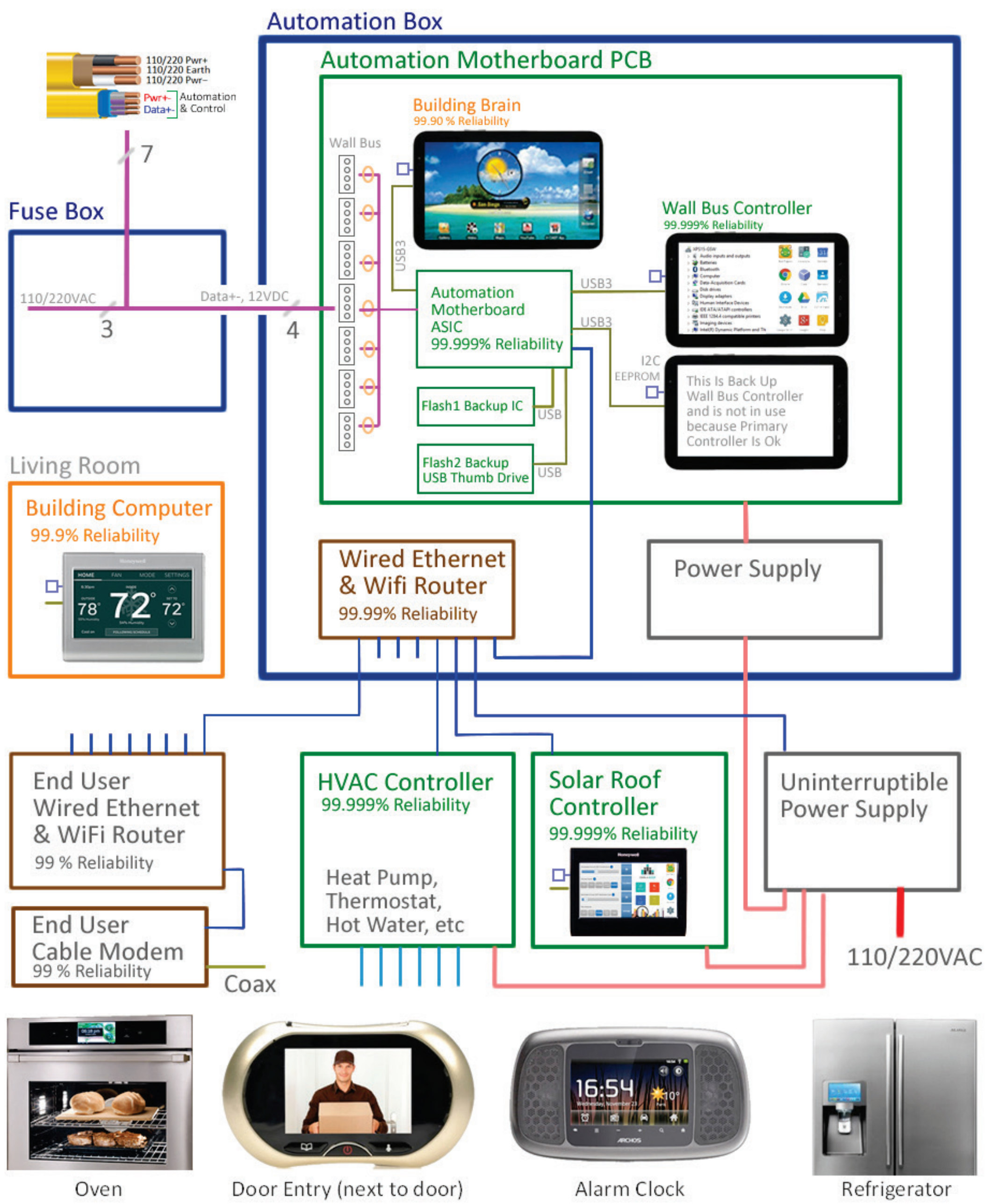
ADF is initially run by Manhattan 2 yet is eventually spun off as a privately funded non-profit foundation who's board of directors are university professors from various countries who are charged with promoting the public good. These professors are talented engineers and a maximum of 25% are architects. If we want robots and software to work well we need engineers.

Chapter 24) Standardize Building System Design

Devices inside a building do not coordinate well due to industries and companies who focus on their own needs. Devices from Amazon, Apple, Samsung, Microsoft and Google often don't talk and HVAC controllers are typically proprietary. This all begs the question, "How can we unify the various building components to save cost, increase convenience, and reduce energy consumption?" It turns out there are many things, several of which we discuss here.

System Design

Below is a possible System Design for our standard 13KW solar house described [previously](#).



The equipment room in the basement includes the HVAC System, Fuse Box, Automation Box, and Solar Roof Controller. These are mostly independent systems, yet do talk via wired Ethernet, which is more reliable than Wi-Fi.

HVAC Strategy

HVAC devices (e.g. heat pump, air conditioner, heater, fans and dampers) are typically controlled by a proprietary HVAC Controller. An example is the Honeywell TrueZone. Communication and control between HVAC devices is often 99.999% reliable and introducing an unknown entity would break this, therefore the HVAC controller is here to stay. However, it is reasonable to interact with HVAC components via the HVAC controller Ethernet port, as illustrated above.

New Term: "Wall Bus Controller"

The 4wire Wall Bus connects the Automation Box (e.g. in basement next to fuse box) to many $1 Wall Bus nodes throughout the building (windows, doors, occupancy sensor, fans, and dampers). These nodes are controlled with a computer called the "Wall Bus Controller" (WBC). This controller connects all nodes to high level software running on other computers, and keeps track of different devices on the Wall Bus network. In order to provide 99.999% reliability, it is kept simple and does not allow other software to be added. In other words, the WBC moves simple data (e.g. "open house window" command) between $1 Wall Bus nodes and Ethernet, very reliably, and does little else.

New Term: "Solar Roof Controller"

The smart solar fabric, described previously, involves many solar cells, switches, and consumers of electricity (e.g. heat pump, AC wall outlets). The Solar Roof Controller (SRC) must coordinate these by performing mission critical calculations every 16mSec. It is possible one would want two of these, so that if one fails the other steps in. This computer is physically located near the solar roof and connects to roof components via Ethernet.

New Term: "Building Brain"

The Building Brain Computer resides inside the Automation Box (e.g. in basement, next to fuse box) and is the master controller for the building. One can think of this as the building's Brain. It makes decisions every ~100mSec based on the specific situation at hand. For example, if one walks into a room, the brain might turn on the lights, raise thermal covers over windows, and crack one window to let in some fresh air. It communicates with building sub-systems via Ethernet. If all system components are working, the building's brain will either make a decision, or manage the making of a decision (i.e. tell specific light to respond to specific switch without involving brain). If a system component is not working, we resort to limp along mode, which involves reduced functionality, yet with some capability.

New Term: "Building Computer"

The Building Computer is affixed to a wall in a central location within a building (e.g. home living room). It takes the place of the thermostat (sorry Nest) and provides a user interface to the building. The user interface for an automated building must be excellent; otherwise, the end user will become frustrated and learn one button well: "Manual".

Back up Flash Ram on Automation Motherboard PCB

Flash RAM memory on the Automation Motherboard PCB is set up as a file system with multiple volumes for the various elements. Large companies involved with automation and devices (e.g. Google, Apple, Amazon) are given space (150mb each * 60 companies =9GB total) and each of the computers shown in the above System Diagram is given space. This memory is helpful when supporting Plug & Walk. Imagine the end user replaces one of the computers in the System Diagram. The new computer needs data in flash to determine how to recover without end user interaction.

This back up data exists on the motherboard flash ram IC and also on a USB thumb drive. The thumb drive is helpful in cases where the entire Automation box is replaced and one needs to move state to the new box.

Fault Tolerance via Redundancy

If the mean time between failures (MTF) with a computer is 2 years, for example, and you want to increase this, you can have two of them. In order to do this, the primary must periodically back up state data to external flash ram, and an external watch dog circuit needs to monitor the health of the computers. Redundancy is justified if computers are cheap and small, and the pain from computer failure is big.

Reliability

In our System Diagram, a controller with a simple operating system and one software program is considered to be 99.999% reliable (shown in green). Computers that accept new software (e.g. Apps that have never been tested w/ the combined system) are considered to be 99.9% reliable (shown in orange). The more important 99.999% tasks involve less innovation and less versatility yet do a few things very well.

Uninterruptible Power Supply

An uninterruptible power supply helps resolve power fluctuations and outages. Software is not well tested for power interruptions, and is therefore often buggy when this occurs. The UPS hardware's Ethernet port signals to all components that a forced shutdown is immanent. Permanent damage can easily occur when a computer is in the middle of doing something and power is withdrawn. Therefore, the UPS is a very important system component at maintaining reliability requirements.

Ethernet

Ethernet is shown in the above System Diagram in blue color. Rugged Ethernet cables connect Automation motherboard to HVAC Controller, Solar Roof Controller, and UPS. O-ring seals and threads are required to keep cables working well over long periods of time. This cost money, yet continuously buggy gear cost more.

The IP rating system specifies physical protection. For example, IP67 is resistant to dust and *temporary* immersion in water;

whereas IP68 is resistant to *continuous* emersion in water. The world needs to agree on Ethernet connectors to be used in the above system diagram to maintain a reliable connection over time and at low cost.

Above is an example of a ruggedized Ethernet connector.

USB 3.2 Dual-Role (DR) and On-The-Go (OTG)

USB, shown above in gold color, is a high speed interface that is available at the base of most smartphones and tablets. DR and OTG are a type of USB that allows the roles of Slave and Master to reverse, as noted in their specification document. Big SOC Integrated Circuits have multiple USB ports built right into the chip. IC's that convert USB to Ethernet range in price between $2 and $7 (e.g. LAN78xx, LAN95xx, and CYUSB3610).

Chapter 25) Create Standard Android-like Computer Device Platform (GAP)

MA2 Places Modified Version of Android Everywhere

In the below illustration we have Android physically located next to the bed (alarm clock), at the front door (entry control), in the oven, in the refrigerator, and on the living room wall (i.e. Building Computer). Google's Android and Apple's iOS are the two big standards for tablets. Android is more open and less costly, and is therefore the logical choice for devices made by 3rd parties spread throughout a building (no, I am not affiliated with Google).


Oven


Door Entry (next to door)

Alarm Clock


Refrigerator

Building Computer
(on living room wall)


Solar Roof Controller
(in attic near roof)


Wall Bus Controller
(in Automation Box)


Building Brain
(in Automation Box)

Create New Android-like Device Standard

MA2 creates a new device standard that builds on top of android, smartphone and tablet technology. This is called "Generic Android Platform" (GAP) and consists of an Android Wi-Fi Tablet or Smartphone with a standard mechanical design determined by MA2 and industry engineers. This means the enclosure height, width, length, and corners must be an exact shape; and the display in an exact position, for each mechanical class. There are five mechanical classes which are defined by the size of the display: 4", 5", 6", 7", and 9". And there are two USB options which are defined as USB ≥3.2 Type C connector or USB 2.0 Micro B connector. In both USB options, the connectors much reside at specific positions. Also, a 3.5mm Stereo jack is required at a specific position.

The 3.5mm jack functions normally, unless an I2C eeprom is attached causing the hardware to boot differently, as specified by data in the eeprom. The I2C eeprom (e.g. $0.30, 16KB, 1x1mm, M24128) clock input pin is attached to the 3.5mm jack microphone pin via a series resistor and the eeprom data pin is attached to the jack LeftSpeaker pin via a series resistor. The RightSpeaker pin can be wiggled by an external circuit in a specific way to tell the device to power on. ~3K ohm series resistors protect microphone/speaker accessories that might be attached instead of the eeprom IC. Also on the tablet PCB are pads for an unpopulated tiny 1x1mm eeprom that could be soldered down if one did not want to touch 3.5mm jack.

The GAP device runs 100% normally if the eeprom is not present, and looks like a normal tablet or smartphone. Subsequently, manufacturers do not need to give up much to support this standard.

Upon startup, the device wiggles the 3.5mm pins and listens for a response. If it sees one, it knows it needs to boot up differently. The boot data tells the hardware which Apps to show, which to hide, which software to run immediately upon startup, which picture to show in the background, which background tasks to run, etc.

The GAP device can be embedded into another product, as shown in above illustration with dotted red to indicate embedded device. Since these are manufactured in high volume and provide great value per dollar, the world can place Android everywhere (no, I am still not affiliated with Google).

In some cases, a buyer would not touch Android and would instead run their own software. This would apply if a buyer was looking for powerful hardware that is low cost due to high volume manufacturing. In many cases, piggy-backing on existing hardware cost less than developing one's own lower volume hardware. In order for this all to work, the buyer's software needs to be pre-loaded onto the standard Android device -- more about that later.

The GAP standard supports both Android tablets and smartphones because it focuses on the mechanical form, 3.5mm boot support, and pre-positioned customer files; and not the presence of a communications radio (i.e. primary difference between tablet and smartphone is often radio).

USA list price for these devices typically ranges between $30 and $150 depending on processor, RAM, and display. There might be a market for $15 to $30 devices (USA list price) with a black and white display, smaller processor, and less RAM. The 2018 Amazon Fire 7, for example, sells for $50.

How Does Android Everywhere Help Energy?

If the alarm clock knows when you will awaken, the rest of the home can be shut down while you sleep, with the kitchen and bathroom reaching set point temperature 1 minute before the alarm sounds.

If the oven timer is set for 45minutes, the solar roof might divert electricity from refrigerator to oven during that time and let refrigeration coast for 45 minutes.

One can adjust energy saving automation features from any location with an Android device.

GAP Kernel

GAP devices contain a kernel that boots before Android. It looks for a 3.5mm boot eeprom and if found, loads it in. If the eeprom does not exist, GAP Kernel drives a stake into its own heart and lets Android take over with a normal Android boot.

The boot eeprom specifies the role of the GAP kernel to be one of:

- GAP kernel shuts down, Android boots normally and supports Apps as specified in eeprom boot file
- GAP kernel is maintained underneath Android and provides services (e.g. VPN to other GAP devices)
- GAP kernel maintains multiple virtual machines, Android is placed into one VM, and other software is placed into other VM's (e.g. Honeywell 99.999% reliable HVAC controller runs Linux in one VM with 90% processor support, and runs 99.9% reliable Android in another VM with 10% processor).

Each GAP consumer (e.g. Honeywell) can control anywhere from 100% of the processor to very little; as determined by the tiny eeprom.

Why GAP Kernel? Because the big companies of the world are not going to embrace this standard if they don't have control over the processor. If they don't get control, they walk. In effect, they require GAP.

Chapter 26) Coordinate Gorillas

Let's take a break from computers for a moment and think about the world in which we live.

The world is full of big companies with great powers. Google, Apple, Microsoft, Samsung, and Amazon are examples of these. A company, big or small, will embrace a technology if it benefits them. Therefore, in order to motivate gorillas to move forward with the technologies discussed in the article, we need to make sure they benefit.

Gorilla 60 (G60)

Below is a list of the 60 largest companies involved with devices, commercial electronics, building automation, computer hardware and computer software. In diplomacy the terms "G7" and "G20" refer to specific groups of countries. Since"G60" is not being used, we will use it here. Below is the "G60".

Platform: Amazon, Apple, Google, Microsoft

Commercial Electronics: Arcelik, BSH, De Longhi, Daiken, Electrolux, Elica, Gree, Haier, Hisense, Hitachi, HP, Huawei, Lg, Lutron, Middleby, Midea, Miele, Nest, Panasonic, Philips, Rinnai, Robam , Siemens, Samsung, Sonos, Sony, Toshiba, Videocon, Voltas, Whirlpool

HVAC: ABB, Carrier, Honeywell, Trane

Tablet Manufacturers: Asus, BLU, DataWind, Dell, HTC, Lenovo, Motorola, Nokia, OPPO, Vivo, Xiaomi, ZTE

Communication Standards: Zigbee, Z-Wave, KNK, EnOcean, IFTTT

Communications: AT&T, T-Mobile, Verizon

Cable: Chart, Xfinity

Unification

In business there is a term called "unification". This refers to multiple companies working together and it often does not occur since they compete, they don't talk, and they don't coordinate. For these reasons, unification seldom occurs. Yet if they play well together in the world's buildings, we can reduce energy consumption. This begs the question, "How, Who and When unifies the G60?" We will first discuss How and then later talk Who and When.

How

Our approach is very simple.
We woo them with treats.

Chapter 27) Create Gorilla Coordination Foundation (GCF)

MA2 creates the Gorilla Coordination Foundation (GCF) to help coordinate the G60. This is initially controlled by Manhattan 2, yet is later spun off as an independent foundation. GCF consist of ~20 computer scientists and engineers. This involves industry engineers on sabbatical, postdoctoral fellows, and PhD students. Headquarters is placed near a university that is strong in computer science and close to G60 companies. The SF Peninsula might be a good location since one can fly direct to SFO from many Asian and USA cities; and it is near Google/Android HQ and Stanford.

20 engineers working for 5 years at $0.3M/yr average cost a total $30M. This is reasonable if it succeeds in motivating G60 devices to speak a common language and better manage the world's buildings.

The GCF team maintains GorillaCoordination.org, creates proposed specifications (eeprom, mechanical standards), writes pre-Android GAP Kernel software, writes security software (authentication, certificates, encryption), and creates test beds for the various components.

www.GorillaCoordination.org

Each of the G60 can begin participation at any time, free of charge. To participate, they designate one person who is their representative and this person is given login access to www.GorillaCoordination.org. It is through this website that a company participates. The first thing a participant does is specify their name in all lower case letters with no spaces and no special characters (e.g. "att", "zwave", "siemens", "huawei"). From that point on, software will use this label to identify them. To encourage participation, GCF woos G60 with treats:

Treat 1 - Backup Storage in Automation Box

The automation motherboard bolted to the basement wall provides two locations to backup data. One is a flash ram IC and the other is a standard USB thumb drive. In both places we have a directory called "oem" and in that directory are sub-directories for each participant (e.g. "oem/honeywell/..") to place files, free of charge. The amount of space they receive depends on what is available. For example, if we have a 16GB flash IC and 9GB is designated for gorillas, then each could receive a maximum of 150MB (9G/60). Participants store information in encrypted form to help them keep track of devices. Also, it helps them back up state information in the event a device breaks or is replaced. For example, an HVAC person installs and configures a Honeywell HVAC controller, it breaks eight years later, and the new one is installed without setup due to backup information stored in the Automation Box.

Treat 2 - Brain Access

The Building Brain, bolted to basement wall, makes decisions, and participants on the network are allowed to receive information from this computer (e.g. what is the current spot price of electricity?).

Treat 3 - Security

Security is a complex topic. In summary, the various system components need to be authenticated, and after authentication, they can communicate via encryption. GorillaCoordination.org is considered to be the trusted authority to help with this process. They maintain security certificates for each participant and they maintain cross certificates to trace each to a trusted authority. GorillaCoordination.org also maintains a pre-Android kernel that boots before Android and helps with security.

Treat 4 - GAP Devices are Pre-loaded with G60 Files

Software Participants upload signed .zip packages that contain files to be pre-positioned onto GAP devices (e.g. "oem/participantName/files.."). They also specify who can use their files. For example, a Software Participant might request all GAP Hardware devices to include a package, or they might specify that only a few be involved. Each GAP Hardware manufacturer controls what is pre-loaded onto their hardware, and can ignore requests from Software participants.

Example Scenario

Here is an example. Zigbee uploads their gateway software to GorillaCoordination.org and tells all tablet manufacturers they can pre-position this software onto their computers free of charge (since it helps zigbee devices talk to internet). The hardware companies ignore this request except for Samsung and Huawei who place it onto their 5" GAP tablets. Honeywell uploads their building control Android app to GorillaCoordination.org and tells hardware companies they can only install this if they have a signed agreement with Honeywell; and Samsung and Huawei sign. Then, Honeywell creates a "Honeywell Building Computer" hardware product with a simple PCB that contains an eeprom that mates with an internal embedded GAP tablet. This eeprom requests that only their pre-positioned Honeywell App run on Android and also request that the Zigbee gateway software run in its own virtual machine (VM) with ≤10% processor support. Honeywell is happy because they now have two suppliers of low cost mass produced hardware. Zigbee is happy because they have more support for their devices. Google is happy because they are selling more Android. Samsung and Huawei are happy because the can sell more hardware. All this works, provided the pre-positioned software does not take too much space relative to available flash memory on the device.

What would Apple, Amazon and Google do?

Apple would probably make use of free processor and storage on everyone's living room wall. What would they do with it? HomeKit gateway software could enable HomeKit devices to access the internet and other networks; and iPhone gateway software could help the iPhone access the building when off-site.

The Google Home speaker with internal gateway product talks to google devices, and the Amazon Alexa speaker with internal gateway product talks to Amazon devices. These companies might feel they are in good shape, provided their speaker product is powered on and installed and running. If this is not the case, they would probably want GAP gateway software to continue to support their devices.

Limitations

Obviously, one is limited by processor speed, flash storage space, dynamic ram space, and network bandwidth. For example, the Honeywell building computer might allow the zigbee gateway software to run, provided it did not use up too much flash memory, ram, processor time, and network bandwidth. The average Android App is 11MB; therefore 25 pre-positioned Apps consume 3GB. IC's with 16GB storage are currently selling for $5, so this is reasonable.

GorillaCoordination.org Publishes Who Does What Where

The GorillaCoordination.org website does the following

1. Provide a way for the G60 to interact with respect to pre-loaded files on GAP computers.
2. Provide a webpage for GAP hardware manufacturers to specify which files are being pre-loaded on their hardware. The website then creates a master .zip package of all files, along with an activity report.
3. Makes available to the public the above reports. Subsequently everyone can see which files are pre-loaded on each GAP device (e.g. "Honeywell Home User Interface App v5.523" on Samsung G10c).
4. Provides a webpage to help create eeprom boot xml files. One can specify that specific categories be included or excluded, specify that specific companies be included or excluded, or specify that specific software packages be included or excluded. The output is an xml file that one places into boot eeprom attached to 3.5mm plug.
5. Makes available to the public all boot files used in every manufactured socket (e.g. Honeywell Building Computer socket enables … software if it is pre-loaded on the GAP device that plugs into that socket)
6. A pubic webpage specifies which software runs given a specific GAP hardware device plugged into a specific Socket (attached to eeprom).
7. Products with embedded GAP devices publish the combination eeprom boot file and list of pre-loaded files; therefore, the public can see what software runs on each product with embedded GAP.

Treat 5 - Building Computer Access

The Building Computer (e.g. bolted to living room wall) provides a user interface from a central location. Participants can interact w/ this computer in a variety of ways, as determined by the company that sells this device.

Building Computer Game Theory

Lets assume Honeywell manufactures a Building Computer that says "Honeywell Building Computer", as shown to the right. The boot eeprom tells the device to only run the Honeywell and Weather Android Apps and it tells the device to run the Zigbee gateway software (support Zigbee Bluetooth devices) and the KNX gateway software (support HVAC interface).

This product does not sell well because Building Computer (BC) products sold by other companies have boot eeproms that enable more software. So Honeywell's next product enables almost everyone's software in order to compete.

Lets assume Amazon sells a BC that talks to Amazon Alexa devices yet not Google Home devices; and Google does the same thing yet in reverse. However, Honeywell's new product supports both and sells more; pushing Google and Amazon to be more supportive of each other in this market, and to compete in other ways.

In the end, the market heads toward Building Computer products that are supportive of everyone's software (without being buggy).

Does one company pay another in return for enabling software via boot eeprom? This issue is between respective parties and does not involve GorillaCoordination.org.

Brain Game Theory

The *Brain* Computer is bolted to the basement wall, it aggregates information, and it makes decisions every ~100mSec. The *Building* Computer is bolted to the living room wall and focuses on user interface. Both are similar in that both have embedded GAP hardware, both are controlled by boot eeprom which is controlled by product manufacturer, and both are under market pressure to be supportive of software from many companies (without being buggy).

The Little Guy

We talk about the G60, yet what about the entrepreneur working 16 hours/day in the garage whose wife left him because she was tired of all his crap? Or the 61st largest company who says "me, me!!"? It is probably best to begin with a small group, get the standard moving forward, and then evolve into supporting more participants. Note that without G60 support, there is no standard.

Rights and Privileges

Rights are provided free of charge to all participants. An example is flash storage space in the Automation Box bolted to basement wall.

Privileges are with respect to a specific hardware device and are controlled by the company that sells that device (and controls boot eeprom). For example, Honeywell makes and sells the Honeywell Building Computer and grants *privileges* to other software participants with respect to that one device via the 3.5mm boot eeprom.

GCF Is Eventually Spun Off by Manhattan 2

At some point in time, MA2 spins off GCF and sets it up as an independent foundation with a charter that focuses on encouraging companies to play well together in an effort to decrease energy consumption, decrease cost, and increase comfort. The board of directors might consist of ~7 university professors from regions throughout the world, since they tend to be less focused on corporate and country interests, and more focused on public interests.

Chapter 28) Create Universal Language for Building Devices to Communicate

Interoperability

The world needs a standard language with which different devices in a building can communicate. One might refer to this as, THE standard data model. If this existed, application software (e.g. on your smartphone) and high level artificial intelligence software (e.g. Building Brain) could talk to devices independent of their product family (e.g. Google Home, Amazon Alexa, Samsung SmartThings, Zigbee, etc.) to reduce energy consumption and increase comfort.

Exactly what is a "Data Model"?

We define the term "data model" as one class that contains a set of fields (e.g. variables) and functions in one file. This is directly supported as a class by object oriented programming languages such as C++, C#, Java, JavaScript and Python. In the non-object oriented C language a data model is represented with a struct and several functions that refer to that struct in one .c file.

Create Data Interoperability Foundation (DIF)

Manhattan 2 forms a new organization called the Data Interoperability Foundation (DIF). This non-profit organization creates and manages standard Data models (e.g. classes). If focuses on the public's interests and treats all countries, all companies, and all individuals equally.

Initially DIF is run by approximately ~30 Manhattan 2 computer scientists and electrical engineers, and later is spun off as an independent non-profit foundation where it is maintained by a handful of computer scientists & engineers. The term "dif" is an oxymoron since the organization is more about "same" than "difference", yet we will overlook that fact for now.

Manhattan 2 places DIF headquarters at a university that is strong in computer science and is interested in this kind of work.

DIF interacts with the world via their website, www.DIF.org.

All work done by DIF engineers is made public on a free gnu license basis.

DIF Supported Products

DIF supports common products, some of which are listed below.

Lighting: Light bulb, light socket, on/off wall switch, variable wall switch

Window: building window thermal blanket monitor/control, building window covering monitor/control (drapery motors, automatic shades, and blinds), building window open/close monitor/control, vacuum in glass monitor/control, building window base sensor wall bus module, window security monitor

Door: door position monitor/control (motor moves to any position to help thermally control each room), door bolt monitor/control

Door Entry: entry door video monitor, door lock monitor/control, external to entry door user interface (person outside requests entry), internal to entry door user interface (person inside controls entry)

Occupancy: occupancy sensor

HVAC: thermostat user interface, thermostat control, pump monitor/control, duct damper monitor/control, duct fan monitor/control, dehumidifier monitor/control

Power: electrical box wall bus wiring PCB, in-line on/off power switch, in-line variable Triac dimmer control, utility company electrical power meter

Sensor: module that optionally measures each of: VAC voltage, VDC voltage, AC current (AC Amps), DC current (DC Amps), temperature (C), differential pressure (Pa), absolute pressure (Pa), flow (m^3/s), relative humidity, absolute humidity, Carbon Dioxide (ppb, CO_2) , Carbon Monoxide (ppb, CO), Radon (pCi/L), wind speed (MPS), wind direction (azimuth degrees), people in room (# of people), Illuminance (lux).

Security: security user interface module, alarm siren

Kitchen Appliance: oven, hot plate, rice cooker, food blender, refrigerator, freezer

Cleaning: vacuum

Clock: wall clock, alarm clock

DIF Engineers Wrap Existing Popular Data Models (e.g. KNX, BACnet, zigbee)

~20 DIF engineers wrap commonly used devices that are controlled by established standards such as Zigbee, Amazon Alexa, Google Home, BACnet, and KNX. This enables high level software (e.g. Building Brain) to work with different types of devices (e.g. light bulb from zigbee, BACnet devices, Amazon Alexa, or Google Home).

For example, a Zigbee light bulb might not explain their specifications, therefore the DIF.org wrapper marks these fields as "unknown". However, the standard DIF TurnBulb_ON() function easily wraps the zigbee bulb control function. With this support, high level software can utilize one subroutine to control all bulbs. DIF engineers ensure that many Zigbee, Amazon, and Google devices can talk to high level DIF software via DIF wrappers.

Companies Compete by Adding To DIF Technology

If an existing company has a proprietary technology and they see DIF moving toward them they might get nervous and think they are having a heart attack. However, after they calm down and catch their breath they might ask themselves, "What can I do in this new world?" A typical response would be to build on top of generic reference designs produced by DIF and MA2 engineers, and improve them. For example, Honeywell engineers might be able to improve the very mediocre MA2 Building Computer reference design and propel Honeywell to the top of this market.

Website Based Data Model Editor

Each data model is a class that contains fields and functions. DIF.org provides a web form-like environment by which one can create and edit classes. A web page for a class with two fields might look like something like the following illustration:

```
Data Model Editor: SimpleLightBulb__PRODUCT_SPECIFICATIONS

class SimpleLightBulb__PRODUCT_SPECIFICATIONS
{
   [DELETE]  [EDIT]  int16      socketType;        // type of socket, e.g. E26
   [DELETE]  [EDIT]  int16      beamAngle_degrees;  // beam angle in degrees units (120 is wide)
}

[ADD FIELD]   [ADD METHOD]  [GENERATE SOURCE CODE]
```

One can edit an existing field by pressing the 'Edit' button, or add a field by pressing 'Add Field'. When one adds, they specify field name, data type (e.g. int16) and a comment.

DIF Code Generator

The DIF.org website maintains a description of each class and each field, and also generates code for each class in 6 different languages (C, C++, C#, Java, JavaScript and Python). The original data typically is not in a programming language, and is instead in a database managed by DIF.org.

A source code generator options panel enables one to specify which files are generated for a given data model. For example, if you are writing smartphone python code and want to control light bulbs you would set up the panel as shown below and click 'Generate Source Code'.

```
Generate Source Code: SimpleLightBulb__PRODUCT__client


    Server Code
        [ ] C,   [ ] C++,   [ ] C#,   [ ] Python,   [ ] Java,   [ ] JavaScript

    Client Code
        [ ] C,   [ ] C++,   [ ] C#,   [X] Python,   [ ] Java,   [ ] JavaScript

    [X] Include data models that are utilized by this data model (dependent resources)

    [X] Include Source Code for DIF Glue
```

DIF.org Imports Classes from Other Languages

DIF.org manages data models (class with fields and methods) and enums (list of constants). After creation, one can export source code to other languages and also DIF network glue can connect client and server versions of a data model. To establish a data model or enum at DIF.org, one has several options:

- Manual - Create from scratch using DIF.org website, described above.
- Import - Import an existing class or enum written in C, C#, C++, Java, or Python
- Lock - Upload original source code and DIF locks to this file (editing not allowed)

When one creates a class or enum manually, the data is created using a web form, described above.

When one Imports an existing file, it is scanned and copied into the DIF.org system. One can then edit it using the DIF.org editor (add, delete, change). After editing, it will not match the original.

When one Locks to an existing file, it is scanned and copied into the DIF.org system, and one cannot change the original data other than add documentation and mark fields as Permanent or Dynamic. To update the data, one must upload a newer version of the same file.

Standard and EndUser Data Models

There are two types of data models, Standard and EndUser. 'Standard' are controlled by DIF engineers and 'EndUser' are controlled by anyone who registers for free at www.DIF.org. A participant might be a 16yo student who is learning computer science or a microprocessor design engineer that sets up a class that relates to a set of registers inside a microprocessor for purposes of diagnostics and control. End users mark their own models as 'HiddenFromPublicView', 'VisibleToThePublic', or 'OnlyVisibleToDifOrgRegistrants'. In many cases end user models are ignored; however, in some cases, they are helpful to people connected to the author in some way. The DIF.org code generator creates files in multiple languages to help devices communicate over the network.

DifClassID is a fixed 16bit integer that refers to a standard class managed by DIF.org. In the case of end user classes, we use a 32bit number that refers to the EndUser registered at the DIF.org website (their serial#) and a 16bit number that they use to refer to their specific classes. DIF glue supports both Standard and EndUser since it does not know anything about content other than each field has an ID, byte size, and byte offset when packed.

EndUser to Standard Migration

Suppose a thermodynamics engineer creates data models that thermally describe each room, duct, fan, damper in a building (e.g. thermal inertia). And suppose several low and high level programmers are using these. If a DIF engineer is later asked to set up support for thermal modeling tools, he/she might consider copying existing work and converting it to a Standard model. After copied, it might be marked as "Proposed" for a year while the engineering community comments and improves. Ultimately it is frozen and marked "Final Version 1.0".

Public Access

Anyone can view 'Standard' data models and non-hidden 'EndUser' data models. Also, anyone can register at DIF.org, free of charge. After registering, one can create their own data models and generate source code for their own models and any other visible models. Also, they can comment on any field and any function of any visible model on the system. For example, a field that describes light bulb power might be a 16bit integer in units of 0.1Watts with a range of 0.1 to 3276W and someone might comment they prefer 0.2W and range of 6KW. Or someone comments that you should add a field for bulb color temperature (e.g. 2700°K warm white) in units of °K to better describe a bulb.

Rules

There are several rules that govern the DIF system:

- Anyone can register at DIF.org, free of charge, create data models, and decide if each is hidden or visible to the public.All public data models (seen by public) are available to the world on a free gnu license basis.
- Fields and subroutines are never deleted from a data model, only added.

- Fixed classID, fieldInClassID, and subroutineID never change.
- Only DIF engineers can create or change a standard model.
- Anyone registered at DIF.org can comment on a model, field, subroutine, etc.
- Participation in DIF.org is free.

Chapter 29) Working with DIF Data Models

DIF data models are controlled by the DIF website and help to coordinate multiple companies.

Permanent Fields

Fields that never change (even when one cycles power) are referred to as "Permanent" (e.g. light bulb specifications). These are marked as such so that high level code only needs to read them once from a device. For example, your smartphone might gather specifications of all devices in your house and cache them into a file on smartphone flash (i.e. as a set of standard data models) to improve system performance.

Dynamic Fields

Fields that change are referred to as "Dynamic". For example, a data model that describes a light bulb on/off state might have one field that indicates whether or not the bulb is on or off, and it might have two subroutines that changes that state (e.g. TurnBulb_OFF(), TurnBulb_ON()). Flags specify behavior as UpdatesOnceAtBootUpAndThenNeverChanges, UpdatesOnlyWhenPeriodicallyCalibrate, and UpdatesAtAnyTime.

DIF Supports fields of type Value, Enum, Array, Pointer, Flags, and Handle

The DIF.org website enables one to create fields within data models that are simple variables (e.g. int16, flt32), pointers (e.g. int16*), handles (e.g. int16**), handles to pointers (e.g. int16***), flags, arrays and/or enums.

DIF allows one to maintain enums using a webpage user interface where one defines each item (name, value, description), and then creates source code in multiple languages with the press of a button.

If a server data model has a pointer then the companion client data model would convert that to a ReadOnly variable (e.g. 'int16*' in server is represented as 'int16' in client).

Devices Expose their Data Models to Network

Each physical device exposes a handful of standard models to the outside world via the network. In the case of the light bulb, there might be one data model with permanent product specifications (e.g. what is written on package label at store); and one with a dynamic field (on/off state) and subroutines (to change state).

DIF_ClassID, DIF_FieldInClassID, DIF_SubroutineID

To support this interoperability, each class, each field within a class, and each subroutine is given a fixed integer ID code. Therefore, one can have a function SetFieldValue(DIF_ClassID, DIF_FieldID, newValue) in one language that sets a field in another language. SetFieldValue() is hard to read, therefore DIF.org generated wrappers make it easier, an example of which is shown below.

```
TurnBulb_ON() { SetFieldValue(DifClassID_Bulb, DufFieldID_Bulb_State, 1); }
```

The DIF.org website maintains fixed ID's and defines these when it generates code (e.g. in enum). During code generation, wrappers similar to what is shown above are also placed into source code files.

DIF Maintains LowLevel and HighLevel Versions of Data Models

In some cases a device will have a low cost processor with little ram, no floating point support and/or a low bandwidth connection to the bus master. In this case, the first powerful gateway with DIF support appends a HighLevel model to the LowLevel. LowLevel might avoid floating point numbers and might packs data as tightly as possible. HighLevel works with floating point numbers, Standard SI Units, and Microprocessors that are reasonably powerful. Below is an example DIF webpage for defining a field that has both a HighLevel and LowLevel version. LowLevel represents temperature with a 16bit integer in 0.01°C units and HighLevel represents temperature with a 32bit floating variable in °C units.

Define One Field

```
LowLevel Field
       Name:        [ temperature_10milliC_units_int16 ]
       Description  [ temperature in 0.01C units, -320C to +320C range ]
       Data Type:   [ int16 ]

HighLevel Field
       Name:        [ temperature_degree_C_units_flt32 ]
       Description  [ temperature in 1C units, flt32 ]
       Data Type:   [ flt32 ]

Mapping
       Heavy = (((double) Light) * [ 100.0 ]) + [ 0.0 ]
```

When one generates code, a file is created for LowLevel and a file is created for HighLevel, with conversion functions (or macros) that convert from one to the other. For example, a Wall Bus temperature sensor 1x1" PCB might return a 16 bit integer to the Wall Bus controller (powerful processor in Automation Box) who manages a HighLevel data model that is tacked onto the LowLevel data model; and exposes both to the network.

DIF Webpage Helps Wrap Existing Standards

DIF engineers wrap HighLevel data models, described above, around data models from existing standards. Existing standards include zigbee, BACnet, KNX, Amazon Alex, and Google Home. These are considered "LowLevel" within the DIF system. To do the wrap, DIF engineers perform the following steps:

1) Import into DIF website data models from common standards (e.g. zigbee, BACnet). DIF considers these LowLevel.
2) If a relevant DIF HighLevel data model exists (e.g. flt32 variables in SI units), DIF engineer writes code that appends HighLevel data model to Imported LowLevel model. If the Imported data model works with integers that are ultimately represented as flt32 in SI units, and a corresponding HighLevel data model does not exists, then DIF engineer creates it.

In some cases, wrapping is done by hand, yet in others one types a line of code in a web form that calculates the value of a high level field based on lower level fields, and let the code generator create the wrapper as shown below.

Wrapped High Level Field - HighLevel.MeasuredVAC

Zigbee wrap

```
[ HighLevel.measuredVac.value = zigbee.measuredVAC * zigbee.multipler / zigbee.divisor ]
```

BACnet wrap

```
[ HighLevel.measuredVac.value = BACnet.measured * 100.0 ]
```

KNX wrap

```
if (KNX.measuredVoltsVac <= KNX.minPossibleVac ) {
   HighLevel.measuredVac.error = Error_Too_Low;
   HighLevel.measuredVac.value = 0.0;
}
else {
   HighLevel.measuredVac.error = NoError;
   HighLevel.measuredVac.value = KNX.measuredVoltsVac;
}
```

DIF Engineers Wrap Tunnels to Existing Standards

Currently there exists multiple standards for devices in a building, an example of which is BACnet. To connect DIF to existing standards, the DIF engineer would do the following:

1) Upload enum and class files from existing standard (e.g. BACnet) into DIF.org and lock to original. After locking, one cannot change data, only upload new versions. DIF.org supports uploading multiple files in a .zip package; subsequently DIF engineers can quickly pull in an existing standard (e.g. upload 30 enum files from BACnet at one time).
2) Write "tunnel" code between client and existing standard. For example, write glue code between Building Brain server and BACnet "Binary Lighting Output", where BACnet device appears in the DIF system.
3) If a high lev
4) el data model exists for a device, write wrapper code around its tunnel. For example, DIF_Standard_Light_Bulb wraps around BACnet "Binary Lighting Output" and also wraps around zigbee light bulb; subsequently, the Smartphone App programmer can interact with both in the same way.

DIF Webpage Generates Code That Populates Permanent Fields

DIF maintains Data Models that define common products, several of which are listed here. A "product" is defined with several data models that are nested to form one master data model which we think of as a "product" with an SKU number and a price. Also, this product connects to the network in one way or another and is thought of as one "node".

DIF.org can generate source code for any product or any data model where the permanent fields are defined in a web form. The below illustration shows an example of this for a light bulb. The participant presses a button to create device source code with permanent fields loaded as defined here.

Create Device Source Code - Light Bulb

```
DIF_BASE_PRODUCT_DESCRIPTION_server
       vender_SKU_number:                [ SKU-23523   ]
       manufacturer_name:                [ Philips     ]
       manufacturer_URL:                 [ Philips.com ]

DIF_SimpleLightBulb__PRODUCT_SPECIFICATIONS_server
       socketType:                       [ E26  ]  type of socket, e.g. E26
       beamAngle_degrees:                [ 90   ]  beam angle  in degrees units (e.g. 120 is wide)
       power_Consumed_100mWattsUnits:    [ 7.0  ]  power consumed by product, +-32K value, 0 to 3000W, 0.1W
```

Manufacturing Companies Declare Specifications

Manufacturers who register at DIF.org can declare specification of their products (e.g. for each SKU #). DIF keeps these in a database for each product type, and it is unclear what happens next. Ultimately, this data could help consumers and designers select products. However, this is an invitation to massive fabrication. A 400LBs teen ager sitting on his bed in New Jersey could declare he is a manufacturer when he is really reselling out of his mother's basement, saving money for an MBA degree. Who insures the data is accurate? DIF does not have the energy to chase after ~~every teenager~~ everyone who registers and declares.

Data Model Fields Can Refer to Other Data Models

A field can point to another data model, therefore one can establish relationships between them. In the below example, we create a data model for a light bulb product that contains both a product specifications data model and an on/off control data model. The int32 offset parameters tell us where objects resides with respect to the beginning of the master DIF device data block.

```
struct DIF_LIGHT_BULB_PRODUCT_server
{
    DIF_BASE_PRODUCT_DESCRIPTION_server                *baseProductDescription;
    Int32                                               baseProductDescription_offset;

    DIF_SimpleLightBulb__PRODUCT_SPECIFICATIONS_server *productSpecifications;
    Int32                                               productSpecifications_offset;

    DIF_ON_OFF_CONTROL_server                          *onOffControl;
    Int32                                               onOffControl_offset;
}
```

Chapter 30) DIF Supports Server and Client Versions of Data Models

A device (e.g. C server) instantiates data models and exposes them to the network via DIF glue code. Subsequently, high level code (e.g. client C# or Python) creates a DIF session and gains access to the objects in the device.

The DIF.org code generator creates server classes, and also companion client classes. For the most part, client classes are the same as server yet without dynamic fields. Only permanent fields are shown in client classes.

In the above LIGHT_BULB_PRODUCT example, the DIF.org code generator creates companion client files in many different languages. For example, the C# client code would appear as follows:

Code generator creates C# *client* class that relates to entire product:

```
public class DIF_LIGHT_BULB_PRODUCT_client
{
   public DIF_BASE_PRODUCT_DESCRIPTION_client   baseProductDescription;
   public DIF_PRODUCT_SPECIFICATIONS_client     productSpecifications;
   public DIF_ON_OFF_CONTROL_client             onOffControl;
}
```

Code generator creates C# *client* data model with permanent fields:

```
public class DIF_PRODUCT_SPECIFICATIONS_client
{
   int16 socketType;          // type of socket, e.g. E26
   int16 beamAngle_degrees;   // beam angle in degrees units (e.g. 120 is wide)

    .. all permanent fields are shown here
}
```

Code generator creates C# client class with functions that enable control.

```
public class DIF_ON_OFF_CONTROL_client
{
    // the following dynamic fields are not included in client data model:
    //     onOrOff, state..

    public void TurnBulb_ON()  { .. glue .. }
    public void TurnBulb_OFF() { .. glue .. }
}
```

The server programmer (e.g. C code in device) instantiates LIGHT_BULB_PRODUCT, PRODUCT_SPECIFICATIONS, ON_OFF_CONTROL, BaseProductDescription; connects the latter three to the first; and then exposes these to the network via DIF glue code.

The client programmer (e.g. C# in smartphone) creates a DIF session to gain access to the LIGHT_BULB_PRODUCT_client class.

The beauty of this scheme is it makes it easy for the server programmer (e.g. C code in device) and the client programmer (e.g. C# or Python in smartphone).

DIF Creates Server/Client Testbed

DIF.org can generates a project that forms a testbed for each data model. This includes graphical console window and processor thread for each of device, gateway, and client (e.g. smartphone). One can then use the debugger to follow data movement through the system. Also, DIF.org can create multiple projects in different languages that communicate via a socket (e.g. Device C project talks to Client C# project).

Packed Data Matches C Structs

The DIF.org code generator packs/unpacks fields in each data model in the same way C stores fields in memory (same byte alignment and size for each field). Subsequently, one can directly move a C struct's data to/from a network payload buffer. DIF.org generates the C struct, pack/unpack code, and byte offset values; and therefore has control over alignment and position of each field. Also, the DIF generated source code contains functions that flip bytes to convert from one endian to the other as needed.

Here is an example case where a light bulb product description in a C struct is read by high level python code in a smartphone: the phone sends the DIF ReadAllFieldsInOneDataModel command to the device, the device packs those into a data block via DIF generated C source code (e.g. move C struct into payload buffer), the data block moves through the network, the DIF generated python source code unpacks and places data into a python class.

Subroutines

Parameters passed in subroutines are managed in the same way as a data model since they involve a set of fields of variable size that are packed into a ram buffer. In a sense, subroutines with parameters are like a subroutine with one parameter which is a pointer to a C struct, and the struct is represented as a data model in the DIF.org system.

Tables of Field Descriptions

All programming languages (e.g. C#, Python, Java) support a FieldDescription class that describes one field in a data model. It contains parameters like byte size, data type (e.g. int16, flt32), byte offset (when packed). An array of these is maintained in ram memory for each data model in use (i.e. similar to a table). For example, the four fields in the LAST_MEASURED_VALUE data model (described above) would involve an array of four FieldDescription structs, where each describes one field in that data model. Two arrays are maintained to convert indexInToArray to/from DIF_FieldInClassID (one level of indirection). Also, a 32k element array is maintained to convert DIF_ClassID to a pointer that leads to one of these tables.

DIF glue code maintains these tables and arrays, therefore the non DIF programmer can focus on calling higher level functions and let the DIF glue code do the right thing. When DIF glue is performing I/O it does not use dictionaries, does not scan each element of a list, and does not create a class or allocate memory (unless doing something for the first time).

When high level code first executes, it calls DIF CreateDifSession() to join the network and create data model description tables utilized by the network's devices. If one averages 5 fields per class, 5 parameters for each field, 150 data models per session; then tables that describe data model fields would consume 7500 bytes (5*5*2*150), which is a reasonable amount of memory. CreateDifSession() in many cases does not need to create these tables from scratch, since they already exist elsewhere and can be copied.

Devices (server) export data model descriptions to clients and higher level software so that one can work with fields and data models via ID codes without having knowledge of content.

Tables of Data Model Descriptions

The DIF glue also maintains a DataModelDescription struct that contains information about each exposed data model. An array of these is used to keep track of all device data models.

DIF Exposes Master Data Block to Network

Each device exposes a set of Data Models (e.g. C struct's) to the network via DIF glue. Client code (e.g. smartphone, Building Brain) interacts with exposed device data models by reading their fields, writing to their fields, and calling their functions.

All exposed data models reside in one contiguous block in ram memory. This is referred to as the device DIF master Data Block. Upon boot up the device creates this via DIF glue (DIF_StartCreation_DifDataBlock()), pushes multiple data models into this master data block via DIF generated code (DIF_Create_*DataBlockName*()), loads permanent fields, and then tells the DIF glue that the data block is complete (DIF_EndCreation_DifDataBlock()). The EndCreation function creates tables that describe the data models (i.e. DataModelDescription and FieldDescription arrays), and tacks those onto the end of the DIF Data Block. Ultimately all fields in all exposed data models, and all tables that manage fields and models are packed into one block of sequential ram. In a typical product, this might be 2KB. After EndCreation, the position and number of the items within the block are considered to be permanent. They do not change after boot up, and also after subsequent boots.

The reason we keep everything together with fields in fixed known positions is because this entire data block is copied by DIF glue to gateway devices, the Building Brain, the Building Computer, and end user devices (e.g. smartphone). DIF glue then allows these "clients" to access permanent fields without using the network. The network is slow since one needs to send a message and then wait for a response; therefore we prefer to avoid it if possible. If a building has 100 devices and each device DIF data block is 2KB, for example, then DIF glue might maintain a 200KB file in flash memory for all devices that interact with the network. If a device is powered off or disconnected, it will still be in that block, which is helpful because higher level code needs to manage devices that "blink" on and off. The connected status of each device is another issue and can be ascertained with DIF glue code. Also, if one tries to read a field from a disconnected device, for example, DIF glue will return error code DeviceNotConnectedToNetwork.

Fields within DIF Data Block

Client code can calculate the location of fields within the data block using tables that show location of data models and locations of fields. Subsequently, client code can tell devices to read or write a specific number of bytes starting at a specific offset within the DIF data block. This means client code can read and write fields to/from the device without involving end user code. This is a nice feature since it helps client programmers focus on their high level code and also helps device programmer focus on implementing the bodies of exposed subroutines.

Example Device

Shown below is the Standard Light Bulb product data model maintained by DIF.org. It contains a master product Data Model (LIGHT_BULB_PRODUCT_server) that includes three other data models.

```
struct DIF_LIGHT_BULB_PRODUCT_server
{
    DIF_BASE_PRODUCT_DESCRIPTION_server                 *baseProductDescription;
    DIF_SimpleLightBulb__PRODUCT_SPECIFICATIONS_server  *productSpecifications;
    DIF_ON_OFF_CONTROL_server                           *onOffControl;
}
```

When the device processor boots up (i.e. light bulb), it allocates memory for all exposed data models.

```
// start the creation of the DIF data models that will be exposed to the network
DifDataBlock *difDataBlock = DIF_StartCreation_DifDataBlock ();

// allocate memory for data model, place inside dif data block, DIF_Create..() was generated by DIF.org
BaseProductDescription_server *baseProductDescription = DIF_Create_BASE_PRODUCT_DESCRIPTION_server (difDataBlock);

// allocate memory for data model, place inside dif data block, DIF_Create..() was generated by DIF.org
SimpleLightBulb__PRODUCT_SPECIFICATIONS_server *productSpecifications = DIF_Create_SimpleLightBulb__PRODUCT_SPECIFICATIONS_server
(difDataBlock);

// allocate memory for data model, place inside dif data block, DIF_Create..() was generated by DIF.org
ON_OFF_CONTROL_server *onOffControl = DIF_Create_ON_OFF_CONTROL_server (difDataBlock);

// allocate memory for data model, place inside dif data block, DIF_Create..() was generated by DIF.or
LIGHT_BULB_PRODUCT_server *lightBulbProduct
    = DIF_Create_LIGHT_BULB_PRODUCT_server (difDataBlock, baseProductDescription, productSpecifications, onOffControl):
```

When finished, one finalizes the data block with DIF_EndCreation..().

```
// end the creation of DIF data models exposed to the network
DIF_EndCreation_DifDataBlock ( difDataBlock );
```

After instantiating data models one loads permanent fields, an example of which is below:

```
baseProductDescription->vender_SKU_Number = "HNY-32535";
```

Upon device discovery, the first client checks if it already has each device's master data block (via CRC and length). If it does not, it downloads it into its own internal memory and saves to disk (cache); otherwise it relies on an already cached version.

Discovery, Security, Clusters, Provisioning

This document does not cover important topics like discovery, security, provisioning and clusters due to trying to be as brief as possible. Zigbee does a decent job with these and is therefore a good resource.

Conclusion

Manhattan 2 forms an organization called The Data Interoperability Foundation (DIF) which creates a website that maintains universal data models for common devices in a building. These models can then be used by high level software to better manage a building and energy resources. All work performed by DIF engineers is made available to the public on a free gnu license basis, which encourages worldwide adoption.

Chapter 31) A Simple Light Bulb

The Data Interoperability Foundation (DIF) maintains comprehensive data models of common devices. We will take a deep dive into one device. At first glance this device might seem simple, yet after careful review, not so much. Our example screw in light bulb contains a built-in Bluetooth based microprocessor that provides on/off control from a smartphone. This is considered to be one "node" and one device on the network. The socket itself is not involved -- we are referring to the actual bulb. This kind of product actually exists and is sometimes referred to as a Smart Bulb.

Package Label Product Specifications

What does the world need to know about the bulb? In other words, what specifications are printed on the package label?

This information helps with product selection (BUYING!), product replacement (BUYING!), diagnostics (to reduce debugging time), identification of exceeded specifications, and end user education (smart phone can show these numbers).

An example standard data model for light bulb product specification is shown below. All of these parameters are ignored by (lazy) Zigbee, Z-Wave, Google Home, and Amazon Alexa.

```
//      ------------------------------------------------
//      Simple Light Bulb Product Specifications
//      All data is Permanent and never changes (always the same each time node powers up)

struct  DIF_SimpleLightBulb__PRODUCT_SPECIFICATIONS_server
{
        int16    product_capabilities_flags;          // bits in int16 specify capabilities: VDC_input , VAC_input,
                                                      // DimmerTypeX_compatible, DimmerTypeY_compatible,
                                                      // measure_temperature, measure_vdc, measure_vac, measure_side_light,
                                                      // directToZigBee, directToZwave, directToAmazonAlexa, directToGoogleHome,
                                                      // compilesWithTheStandardAutomationDataModel,
                                                      // containsBlueToothInterface, containsWiFiInterface

        int16    power_ConsumedWatts_100mWunits;      // power consumed by product, +-32K value, 0 to 3000W, 0.1W units
        int16    socketType;                          // type of socket, e.g. E26
        int16    beamAngle_degrees;                   // beam angle in degrees units (e.g. 120 is wide)
        int16    lifetime_10hrsUnits;                 // lifetime in units of 10hrs, e.g. '100' = 1K hrs, 0 to 32k value, 0 to 320K hrs
        int16    lightOutput_lumins;                  // amount of light put out by bulb, lumens (lm) units
        int16    colorTemperatureDegreesK;            // color temperature, degrees K units
        int16    product_VAC_min;                     // product VAC input voltage, 1 VAC units, min
        int16    product_VAC_max;                     // product VAC input voltage, 1 VAC units, max
        int16    product_VDC_min;                     // product VDC input voltage, 0 to 32K value, 0 to 320 VDC, 0.01V units, min
        int16    product_VDC_max;                     // product VDC input voltage, 0 to 32K value, 0 to 320 VDC, 0.01V units, max
    }
```

We work with 16bit integers because low cost processors often do not support hardware floating point, and 2bytes is less than 4bytes. Product specifications do not change, therefore this data is copied to flash memory throughout the network and associated with a unique 64bit product serial number.

What Information is needed to Purchase a Replacement?

The product_capabilities_flags parameters indicates which dimmers are compatible with this bulb. Example scenario: end user notices blown bulb in living room, smartphone lists replacement options, customer clicks to buy, installs, and does *not* hear hum. What does it take to provide a positive out of the box experience to the end user? It is the data model designer's responsibility to make that happen. In other words, now is the time to be comprehensive and fully characterize products with parameters.

Base Product Description

A base product description data model provides fields: GTIN_number, EAN_number, UPC_number, ASIN_number, vender_SKU_number, manufacturer_modelNumber, manufacturer_name, manufacturer_URL, product description (< 96 characters), product_documentation_URL (webpage that describes specific product), optional Bluetooth manufacturer companyID, optional BACnet manufacturer companyID.

Base Unique Unit Description

These fields apply to all product types and are unique to each manufactured unit: ieee_64bit_serial_number (EUI-64), manufactured_DateTime (int64 100nsec since Birth of Christ).

Product Physical Location

Building electrical wiring is designed in software by an electrical engineer that specializes in building power. Electrical boxes in the wall are identified with codes like EB1, EB2, etc. Also, the engineer specifies requirements such as being able to handle a maximum number of amps. The information from this software is available to the installation electrician who can load this into eeprom memory inside an electrical box's node PCB. Subsequently, higher level software can match each electrical box with the building electrical drawing diagram. If there is excessive heat in an electrical box, this helps to identify the location. Understanding cold electrical box temperatures is important too, since the building computer needs to know how low it can set a room thermostat without freezing a water pipe. Also, since electrical boxes can measure power current (Amps), this helps to identify where power is being drawn in a building. If homeowner sees a building diagram along with annual electricity cost for each outlet in each room they are more likely to turn off big consumers of energy.

Sensor Measurement Characterization

Each analog sensor (e.g. temperature, light, humidity, etc.) is connected to an A/D converter data acquisition system. This capability is either contained inside a microprocessor or is external. Each sensor plus A/D can be characterized with the following parameters. These never change and therefore can be cached elsewhere after being read once. On a side note, I designed my first of 30 when I was 15 and called it ADIOS. ADIOS is an acronym for Analog and Digital Input/output System. Zigbee, Z-Wave, Google Home, and Alexa all ignore the below parameters; yet they are required in many applications.

```
//    ------------------------------------------------
//    capability of sensor, this never changes (always the same each time node powers up)

struct DIF_SENSOR_MEASUREMENT_SPECIFICATIONS_server
{
    int32    sensor_MaxPossibleMeasuredValue;                 // sensor maximum possible measured value
    int32    sensor_MinPossibleMeasuredValue;                 // sensor minimum possible measured value
    int32    absolute_AccuracyOfMeasuredValue_noIntegration;  // absolute accuracy of measured value, 0 seconds integration
```

```
                                                                            // (measurement is accurate to +-ThisValue)
        int32       absolute_AccuracyOfMeasuredValue_1PlcIntegration;      // same as above, yet return average value over 1 power line cycle
        int32       differentialMeasurementAccuracy_noIntegration;         // assume sensor and sensor electronics temperature
                                                                            // has not changed, you do two measurements, subtract,
                                                                            // this is the accuracy of difference, no integration
        int32       differentialMeasurementAccuracy_1PlcIntegration;       // same as above, yet return average value over 1 power line cycle
        int32       noise_RMS_noIntegration;                               // rms amount of noise, 0 seconds integration
        int16       timeToDoMeasurement_noIntegration;                     // time to do measurement, 10uSec units, 0 to 0.32 seconds, no integration
        int16       timeToDoMeasurement_1PlcIntegration;                   // time to do measurement, 10uSec units, 0 to 0.32 seconds, no integration
        int32       recommended_min_time_between_measurements;             // recommended min time between measurements, 1uSec units, 0 to 2000sec
    }
```

Minimum/Maximum measured values refers to the possible range of measured values and are limited by the sensor or the A/D system. If one temperature sensor has a maximum range of 110F (43°C) and it measures 110°F then you know the actual measurement is >110°F, which is important information.

If one measures quickly, they can get data quickly, yet it also might not be as accurate as a measurement that occurs over 16mSec (one power line cycle). In many cases, accuracy improves 10-fold.

Typical measurement specifications are absolute accuracy, differential accuracy, and noise (e.g. variation in 100 different measurements).

Absolute accuracy is important. Many processors contain an internal temperature sensor that is not accurate (e.g. +-3°C); and is therefore good for fire detection or dangerous heat (dog knocks over lamp and then covers it with blanket). Yet this is not good for thermostatic control over a room, which needs closer to +-0.2°C accuracy. If one accurately measures 110/220VAC voltages they will see voltage drops along the power cable. This indicates several things, provided one knows the accuracy of each measuring system.

Differential accuracy is the accuracy of the difference between two measurements. If there is an offset error (e.g. temperature sensor off by 3°F) yet otherwise accurate and you measure temperature after opening a window or turning on a hot water pipe, then you can do differential measurement to see how heat flows. Differential accuracy might be 10 times better than absolute accuracy and is extremely important at understanding how energy flows in a building.

'recommended_minimum_time_between_measurements' indicates how much time the system waits before doing another measurement. Temperature changes slowly, and therefore might only be measured at a maximum rate of once every 30 seconds. If one tries to measure more quickly, the system returns the last measured value, instead of going through the network, initiating a measurement, and waiting for the response.

'timeToDoMeasurement' is the fastest amount of time a device can perform a measurement.

Product Permanent Status

A product permanent status data model indicates if a product is permanently damaged or has reduced capability. Some products do a self-test when they first turn on; and the result of this test is saved here. A light bulb or a light socket might measure current to see if bulb is operating properly. No current when on while voltage is applied indicates permanent damage. This struct is calculated when the node powers up; and might change if it sees permanent damage while operating.

```
struct DIF_PRODUCT_PERMANENT_STATUS_server
{
    int16   condition_flags;         // bits that describe status,
```

```
                                    // e.g. PermanentlyDamanged, PerformancePermanentlyDegraded
        int16   errorCodeAfterSelfTest;  // error code after running self-test
    }
```

Maintain Last Measured Value

After a measurement of any analog sensor, one saves the last measured value here. We record the time of day, to help the system determine if it needs to go to the network to measure a value, or return previously measured value. When a node transmits a measured value to a client, it might only send a 16bit value to increase network speed. Also, error conditions might be coded in the -32700 to -32765 range so that errors are communicated in the same 2 bytes as data.

```
    struct DIF_LAST_MEASURED_VALUE_1LSB_Units_LowLevel_int16_server
    {
        int16   lastMeasuredValue_Int16_LowLevel; // last measured value, integer, low level units

        int16   condition_flags;       // bits that describe measurement
        int16   integrationflags_COPY; // amount of integration used

        int32   numOfSamples;          // # of samples averaged to produce one value
        int16   errorCode;             // error code
        int16   dayTime;
    }
```

Also we have an int32 version of this data model that contains an int32 lastMeasuredValue field in units of 1/1024 LSB (~0.001 LSB). This is helpful when one measures an A/D multiple times (e.g. 1000) and returns the average value. Averaging is extremely important since it reduces noise, increases resolution and increases differential accuracy.

```
    struct DIF_LAST_MEASURED_VALUE_Point01_LSB_Units_LowLevel_int32_server
    {
        int32   lastMeasuredValue_Point001_LSB_Units; // last measured value in 1/1024 LSB (~0.001) units
        .. same as LAST_MEASURED_VALUE_int16
    }
```

To control integration one uses the following data model:

```
    struct DIF_ANALOG_INPUT_INTEGRATION_server
    {
        int16   integrationflags;   // bits that describe measurement, e.g. Integration_OFF,
                                    // Integration _1PLC, Integration _N_samples,
                                    // Integration_N_uSec, Integration_M_mSec
        int16   Value_N;            // specifies how much integration is performed
    }
```

At some point in the network, DIF glue code converts the above LowLevel values to HighLevel 32bit floating point values in standard SI units (e.g. Volts, Amps, Meters, Kg); and this is what the high level programmer sees.

```
    struct DIF_LAST_MEASURED_VALUE_StdSiUnits_HighLevel_int32_server
    {
        flt32   lastMeasuredValue_HighLevel_StdSiUnits;
    }
```

Light Bulb Control

A light bulb can be turned fully on or fully off; or in some cases, controlled variably.

```
    struct DIF_ON_OFF_CONTROL_server
    {
```

```
        int8    onOrOff;                  // 1 = on, 0 = off
    }

    struct DIF_VARIABLE_CONTROL_server
    {
        int16   variableControlValue;     // set output amplitude, 0 to 32K
    }
```

Product Diagnostics

Product manufacturer optionally provides a URL to provide diagnostics. System can append many parameter values to this URL and then manufacturer website can present a web page with more information. If bulb is only drawing 30% of expected current (Amps), resulting webpage might say in big letters "This bulb has degraded performance. Its energy consumption is 30% less than expected; and is therefore probably emitting 30% light than what one might expect. Replacement is recommended."

```
    Struct DIF_PRODUCT_DIAGNOSTICS_URL_server
    {
        str63   baseProductDiagnosticsUrl1;
    }
```

At this time Comprehensive Data Models do not Exist

Unfortunately no one has built comprehensive data models for typical products and no one has created a website to help manage them. There are several who have dabbled in this area, yet with severe deficiencies. For example, Amazon Alexa and Google Home model a light bulb as a brightness level, yet provide no other information (no power specification, no socket type, no voltage rating). Zigbee refers to device temperature as an integer in Celsius units (+-2°F accuracy), which is not sufficient for many applications (0.01°C units would be better). Also, Zigbee does not provide a way to specify measurement accuracy (i.e. in cluster specification), which makes it difficult to know how to interpret a value. Zigbee has done decent work in areas where their networking test bed identifies issues. This includes security, discovery, provisioning, and managing of clusters. For a summary of several building automation efforts, click here.

Review

This chapter described in detail how one might go about characterizing a simple house hold device like a light bulb. DIF engineers provide comprehensive support to approximately 40 Other Devices in an effort to create a common language by which devices in a building communicate.

Chapter 32) Wall Bus Runs On DIF

Previously we discussed a proposed 4 wire bus standard that connects $1 nodes to a central location. This bus is snail slow at 10Kbits/sec, yet is very reliable, is short circuit protected to 220VAC, and supports a tree topology found in typical building 110/220VAC power wiring. The bus master, referred to as the "Wall Bus Controller" (WBC), manages tiny "node" devices on the network and connects them to a fast local area network via wired Ethernet or Wi-Fi. A typical node microprocessor contains 32KB of flash memory, contains 32KB of dynamic memory, is void of floating point hardware, and cost $0.50. The purpose of the typical node is to perform simple I/O such as measure a temperature, turn on a fan, or close a door. If one wants a more robust processor with more communication bandwidth at a node location they can connect node power to a more powerful processor (at perhaps ten times the cost) with Wi-Fi connectivity and ignore Wall Bus signaling. Or combine the two by using Wall Bus to wake a sleeping Wi-Fi-based processor. The Wall Bus Controller (WBC) is mounted on the Automation Motherboard, and communicates with multiple Wall Bus networks via a custom ASIC, as illustrated here. Each Wall Bus network is typically routed in parallel with 110/220VAC power wires within building walls. The 10Kbit/sec (1mSec per byte) data transfer rate is very slow; however, a typical house might have ~15 circuits (fuses) and we run one wall bus network along each in a 4 + 3 wire cable. Therefore, 15 wall bus networks in parallel yield a total of 150Kbits/sec or 70uSec per byte; which is more reasonable.

Wall Bus Runs on DIF

Wall Bus uses the DIF standard data model system to expose sets of fields (e.g. in C structs) to the network, and to manage a hierarchy of data models (e.g. nested C structs).

DIF Product Hierarchy: Product, SubCircuit, Feature, Interface

Each device exposes to the network a DIF product hierarchy that supports 4 levels:

DifProduct - This is the highest level in the hierarchy. A DifProduct can be thought of as one node (physical wires attach from network), one physical device, and one product that a consumer buys at the store. This supports fields like SKU number, manufacturer name, product model #, etc.

DifSubCircuit - This is one electrical circuit or software module within a product. For example, one product might include a motor controller electrical circuit and a Wall Bus power voltage/current monitoring circuit. Each of these interacts with the network via a standard DIF data model, and each is considered to be one DifSubCircuit.

DifFeature - This is one feature within a SubCircut. For example, a wall bus voltage/current monitoring circuit might have two DifMeasurement features, one for WallBusPwr Voltage and one for WallBusPwr Current. And a motor controller might have a DifMotor feature for each motor.

DifInterface - This is the lowest level and is contained inside of features. For example, the DifMeasurement feature contains one SENSOR_MEASUREMENT_SPECIFICATIONS data model to specify accuracy and one INPUT_PORT model to read an analog value.

Curtain Motor Control Example

A physical module at the top of a house window that contains a motor and moves a cloth curtain might look like this:

```
DifProduct     DifHouseWindowCurtainMotorControlModule

DifInterface     DIF_PRODUCT_PERMANENT_STATUS (e.g. errorCodeAfterSelfTest)
DifInterface     DIF_PRODUCT_DIAGNOSTICS_URL
DifInterface     DIF_BASE_PRODUCT_DESCRIPTION
DifInterface     DIF_BASE_UNIT_DESCRIPTION
DifInterface     DIF_PRODUCT_PHYSICAL_LOCATION

DifInterface     DifHouseWindowCurtainMotorControlModule__PRODUCT_SPECIFICATIONS

DifSubCircuit    Dif_WALLBUS_4WIRE_POWERSUPPLY_MONITOR  DifWallBusPowerSupplyMonitor

DifFeature          Dif_WALLBUS_ANALOG_MEASUREMENT  DifWallBusPowerSupplyMonitor_Voltage
DifInterface          DIF_ANALOG_INPUT_INTEGRATION
DifInterface          DIF_SENSOR_MEASUREMENT_SPECIFICATIONS
DifInterface          DIF_LAST_MEASURED_VALUE_Point01_LSB_Units_LowLevel_int32 -- GetValue() returns voltage
DifInterface          DIF_INPUT_PORT  inputPort_measureVoltage -- GetValue() returns voltage

DifFeature          Dif_WALLBUS_ANALOG_MEASUREMENT  DifWallBusPowerSupplyMonitor_Current
DifInterface          DIF_ANALOG_INPUT_INTEGRATION
DifInterface          DIF_SENSOR_MEASUREMENT_SPECIFICATIONS
DifInterface          DIF_LAST_MEASURED_VALUE_Point01_LSB_Units_LowLevel_int32 -- GetValue() returns current
DifInterface          DIF_INPUT_PORT  inputPort_measureCurrent -- GetValue() returns voltage

DifSubCircuit    Dif_MOTOR_CONTROLLER curtain_MotorController

DifFeature          Dif_ONE_MOTOR  curtain_motor
DifInterface          DIF_MOTOR_SPECIFATIONS
DifInterface          DIF_MOTOR_CURRENT_POSITION_AND_STATUS
DifInterface          DIF_MOTOR_CONTROL -- SetValue() sets position of motor
```

The above might seem a bit bulky; however, if fields for each data model consume 25 bytes each and one has 20 data models, the total is only 500 bytes.

Below is an example line of client code to read wall bus power voltage:

```
FrontHouseWindow_DifDevice.DifProduct.DifWallBusPowerSupplyMonitor.
  DifWallBusPowerSupplyMonitor_Voltage.inputPort_measureVoltage.GetValue();
```

To improve readability, the DIF code generator creates wrappers and places them into client classes. For example, the following method is placed into the Dif_WALLBUS_4WIRE_POWERSUPPLY_MONITOR client class.

```
public void Measure_WallBusPowerSupply_Voltage()
{
    DifWallBusPowerSupplyMonitor_Voltage.inputPort_measureVoltage.GetValue();
}
```

DIF R/W Field Request

All exposed data models are stored in a device master DIF data block and the positions of all fields are known by all clients on the network (e.g. smartphone, building computer, building brain).

A client can read or write any exposed field via an array of DIF r/w field requests. These come in two flavors, int16 and int32. The DIF int16 R/W request uses one bit to indicate direction (r or w), 3 bits to indicate data size (1, 2, 4, 8, 16, 32, 64, and 128 bytes), 2 bits to specify indirection (value, pointer, handle, pointer to handle), and 10bits to indicate offset into the master DIF data block.

The DIF int32 r/w request is similar yet offset is 18bits (up to 0.25MBytes) instead of 10bits and the exact number of bytes transferred is specified with 11bits (0...511).

Notice we support fields that are pointers, handles and pointers to handles The DIF glue does what it takes to access each value (e.g. node processor adds offset and does 1 to 4 indirections to calculate address of data). This DIF glue code enables a client programmer to easily read or write any exposed field, which allows the client and device programmers to focus on other issues.

Wall Bus Handles R/W Field Requests In Real-time

In the typical network a client sends a read request to a server and the server responds with a separate data packet. Wall bus is so slow the r/w requests can be handled by the node processor as they are being received. This means the node response payload data buffer can be packed while the request frame is being received (1msec per byte), and can be tacked onto the end of the request frame (bus switches from being driven by WBC to being driven by node). In other words, instead of responding with an ACK byte (acknowledge), the node processor returns the requested data followed by a response CRC. This allows read field requests to execute faster and allows the node to go back to sleep instead of later arbitrating for the bus and sending a new data packet.

Wall Bus Requirements

- The Wall Bus Controller processor is much more powerful than node processor (e.g. 100 times); and therefore handles floating point operations like converting LowLevel int16 data to HighLevel SI unit flt32 data.
- Wall Bus performs building automation and control functions that occur at a maximum rate of once every 0.1 to 1 second.
- Network must durable since it is embedded into wall of building that might last 100 years.
- Hardware must be reliable and low cost to encourage massive adoption.
- Network bandwidth (i.e. 1K bytes/sec on each circuit, ~15 circuits/house) is low; therefore data packets are small.
- Automation Box contains Wall Bus Controller (WBC) which is the glue between higher level software and Wall Bus devices.
- WBC and high software interacts with nodes via arrays of R/W Field Request. These read or write device memory locations with one or more levels of indirection and support variable data lengths.
- WBC and high level software interact with nodes via subroutine calls where input parameters are passed inside of a struct and output parameters are returned within an output struct.

Wall Bus Physical Address

The wall bus controller assigns a 16 bit physical address (WbcDevicePhysicalAddress) to a node when it is first installed (e.g. first time powered on after being installed by electrician).

8bits	_4WireBusInAutomationBox_PhysicalIndex	Up to 256 independent four-wire buses emanating from automation box (e.g. one of these busses for each fuse)
8bits	NodeOn4WireBus_PhysicalIndex	Supports 1 to 127 nodes that are connected and an additional 1 to 127 that are not connected

_4WireBusInAutomationBox_PhysicalIndex is determined by the physical location of the 4wire bus on the automation box PCB; and it never changes. One typically has one 4wire bus for each 110/220VAC fuse, since we route 4wire bus with 3 power wires (e.g. fuse box with 20 fuses has adjacent automation box with 20 qty 4wire busses).

NodeOn4WireBus_PhysicalIndex refers to a node (e.g. $1 processor) on a 4 wire bus. This value is provisioned by WBC to each node when node is first physically installed (e.g. module inserted by electrician). The pairing between node unique 64bit serial number and NodePhysicalAddress is copied into Flash (non-volatile) memory on Automation Box PCB, and into Node flash memory. System tries to maintain this pairing such that system components (e.g. WBC, node, etc.) can power cycle or reset, while this address is maintained.

If physical module on network with Plug and Walk capability is removed and replaced (i.e. we see insertion into same physical socket), then new module is given same NodeOn4WireBus_PhysicalIndex. In other words, one can remove a physical module and replace with a different module of same type, and system continues to function (e.g. home owner replaces window opener module in window frame and walks without further interaction).

The WBC maintains jagged arrays that convert from the 16bit physical address to a class that manages each device; subsequently the WBC can receive a packet with a WbcDevicePhysicalAddress and quickly understand how to manage it.

Programmable Timer

Most wall bus node processors can be set up by a client to wake from sleep at a specific time or at specific intervals and implement at task. An example would be to for an A/D converter to measure a temperature every 3minutes and place the result into internal field memory. The A/D might do this by reading a 12 bit A/D many times over 16mSec and calculating the average value to obtain more accuracy and resolution. At any time a client reads the temperature (e.g. +-200°C in 0.01°C units via int16) quickly via a DIF R/W field request.

Alarm inside Node

A client can set up a timer inside the node processor to wake periodically, look for something, and then send a specific data packet depending on the response. For example, a wall switch processor wakes every 100mSec and looks for a change. If it sees a change it sends a data packet to a lamp and tells that lamp to turn on or off. Processors can check doors, windows, occupancy, temperature, wind pressure; and respond as programmed by a client.

Summary

Wall Bus devices work closely with the Wall Bus Controller to provide reliable and low cost access to building windows, building doors, ceiling occupancy sensors, fans, and dampers.

Chapter 33) Innovate With Contests

Manhattan 2 uses carefully designed contests to encourage innovation and to encourage the world to focus on creating low cost technology that reduces energy consumption or reduces dependence on fossil fuels.

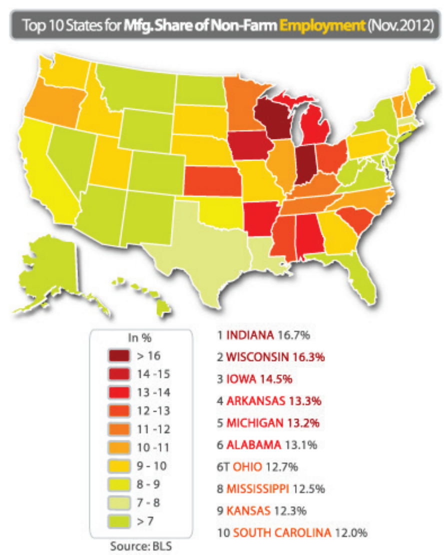
All entrants are required to upload to a server all source code, notes, design files, simulations, internal reports, models, calculations and spreadsheets. These are made available to the public on a free gnu public license basis to help the world manage long term energy challenges. This is not for everyone since many people are interested in controlling IP.

Entrants typically are teams of engineers that consist of 1 to 30 people. Everyone in the world is invited to participate. Materials are submitted in English language. Judges are not told nationality or origin of each entrant; they are to be judged on merit. Judges are selected from the top engineering universities in the USA.

Each entrant has a team leader that determines how prize money is distributed among team members with the larger of 10% and 1/N going to team leader where N is number of people in team.

Contests Managed by Large School Close To USA Manufacturing

The ultimate goal of Manhattan 2 is to develop techniques to replace existing fossil fuel based energy infrastructure; and manufacturing companies are responsible for making tools and supplies to support those technologies.



Subsequently, Manhattan 2 selects a large university in the center of American manufacturing to manage Manhattan 2 contests. If we look at states involved in manufacturing, shown to the right, we see Illinois, Indiana, Ohio, and Michigan in the center. Large manufacturing schools in this region are U of MI @ Ann Arbor, U of IL @ Champaign/Urbana, and Perdue.

Contestants Collaborate with Industry

Contestants collaborate with industry in several ways: industry engineer helps contestant, company joins university-based team, company hires team member, company invites team members to visit, and team uses company facilities to help build prototypes.

Also, in some cases, the contestant *is* industry. Anyone can enter a contest, which means a company can do so as well. Or a wealthy individual who puts together a team of engineers. Or a retired engineer who has some money and wants to get out of the house and do something.

Annual Manhattan 2 Event

Manhattan 2 hosts an annual event near Chicago O'Hare airport during a month of nice weather. "May 2" might be a fitting since is easy to remember "May 2 is MA2 day".

Contestants and Manhattan 2 engineers present their work, contestants are judged, prizes are awarded, and some people go back to O'Hare with one additional bag of luggage.

We go to O'Hare because convenience encourages participation. From O'Hare, eight airlines fly direct to London in 8 hours, four fly direct to Tokyo in 13 hours, and many fly direct to American cities in 1 to 3 hours.

Contest: $1M for Best Smart Solar Material Design and Manufacturing Process

$1M is awarded to the team that develops over a 24 month period the best design and manufacturing process for smart solar fabric. This includes:

- Material mechanical design, electrical design, power design, signaling standard to solar controller.
- Submit a 1x2m sample that produces AC and DC waveforms of programmable voltage and frequency. Model properties of final product over 100 year lifespan (adhesion force, response to UV/wind/water, etc.).
- Test samples in an environmental chamber (subject sample to heat, cold, salt water, UV).
- Design in software a production line to mass product material in in a factory; inducing factory process design, simulations, and cost models.
- Design of solar power switching, filtering and routing to typical consumers (AC motor, DC motor, heating element, 110/220VAC outlet).
- Design, simulate and prototype Power PCB (PPCB) that interfaces to solar cells of different voltages (e.g. 1V, 2V, 4V, etc.). PPCB performs switching via mosfets/triacs, applies analog filters, and routes final output to consumers. Simulation involves characterization of each component.
- PPCB contains A/D that monitors current and voltage of each individual cell under differing loads.
- Microprocessor and/or FPGA controls power switching; and interfaces to Windows computer via wired Ethernet.
- Designer models individual electrical components under stress and calculates mean-time-between-failure (MTF). This includes filtering components that might get hot and blow.
- Study and prototype filtering requirements and techniques.
- Characterize, simulate, and prototype a variety of loads (e.g. inductive motor, switching power supply).

Contest: Design $3, 100mA/12V, 0.01 ATM pump with Pressure Sensor and Valve

Vacuum insulation in windows, walls and appliances can possibly be maintained with a small pump. Teams design $3 (1k qty pricing), 100mA/12V, and 0.01 ATM (1kpa) pumps to be used for such applications. Pump includes motorized valve(s) which connect pump to vacuum, and includes sensor that measures vacuum pressure. One might have two valves with sensor in-between to reduce leak through valve. Pump is tiny since it fits into small spaces.

There are 3 contests, each is 12 months long and each includes a $.33M prize. Contest #1 involves computer design, analysis, MTF analysis, written reports, cost models and simulation (no prototyping). Contest #2 is to create physical prototypes using all information exposed from contest #1. Contest #3 is to design a manufacturing assembly line that manufactures the Contest #2 winning design. Teams in contest #3 create a package for a factory to manufacture. This includes working prototype in final form, mold designs for metal and plastic parts, PCB artwork, test jig design and software. These three contests combined yield a final product ready for manufacture (hopefully). See also: AWG and VIG.

Contest: $1M for Best Automated Window and Door Design

Team designs automated window cavities and modules. There are two contests, each with a $0.5M prize and each is over 24mths. Contest #1 is computer design only and includes 3D drawings, simulations, MTF analysis, cost models, and written reports. Contest #2 is to build working prototypes based on material exposed in Contest #1. Both contests include new construction internal cavities, *and* chassis embedded in wall above existing construction (teams prototype both).

Contest: $1M for Best Vacuum in Glass Design & Manufacturing Process

Same as the Automated Window design contest, yet applies to vacuum in glass window products. This includes simulation of the vacuum, monitoring of the vacuum and control over pump. Teams also design manufacturing process. This includes factory assembly line drawings, simulations, written reports, product cost models and assembly line capital equipment cost models (cost of machines in factory).

Contest: $1M for Best Hot/Cold Water Storage Tank & Manufacturing Process

Similar to above yet create 200gallon water tank that stores heat as well as cold with vacuum insulation. System creates ice that does not damage unit. Design builds on existing technology yet lasts much longer than typical hot water tank, according to MTF simulation.

Contest: Geothermal Trenching

MA2 supports geothermal trenching contests, as described here.

Contest: The $1Million Dollar Chicken

This contest requires team to cook delicious 2LBs of chicken, 1LBs of vegetables in broth, bread, and rice in a vacuum insulated chamber where the end user is not involved after cooking begins.

The cooking chamber is a 6-sided box with vacuum insulation. It contains three internal chambers that are stacked on top of each other that include the top 13"x13"x13" primary chamber, the middle 2" high drawer, and the lower 2" high drawer. Optional refrigeration keeps ingredients cold before cook; subsequently, operator is not required to be present when cooking begins.

Vacuum insulation between chambers enables computer to maintain different temperatures in each. A computer controlled hatch between chambers allow heat to flow between them. The bottom chamber is either for cooking (e.g. bread, corn bread, rice, vegetables) or for an optional plug-in refrigeration unit that chills food before cooking. For example, chef prepares food at night, sleeps, goes to work in morning, and returns to beautiful chicken dinner at 6pm. The middle chamber is used for cooking. Food cooks in covered Pyrex-like casserole dishes inside each chamber. System must kill bacteria (achieve at least 190°F inside chicken).

The prototype is not expected to look nice. It is acceptable for wires, PCB's, and laptops to be scattered around the chamber. The prototype vacuum pump and monitoring is done with external off the shelf products and is not considered to be part of the contest. Design-for-manufacturing and ascetic design is handled later (after the contest).

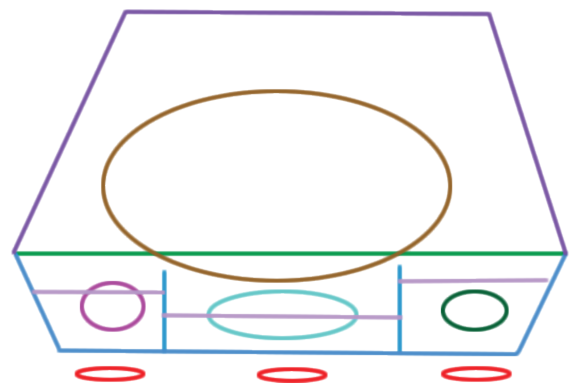
Team monitors 110VAC power entering unit with a laptop (current, voltage, watts, kWh). Also, team measures power that enters other oven products, and compares.

Prototype chamber attaches to laptop via USB or Ethernet, and laptop controls the chamber. Laptop reports in real-time the expected status of each food in chamber. Teams open chamber and cut into food to check accuracy of their food modeling.

Chamber mixes flavors from different foods and distributes liquid over chicken by some mechanism that is easily cleaned. One option is to periodically flip the entire main chamber Pyrex assembly (i.e. turned upside down) to mix flavors and facilitate a juicy end user experience. A dinner plate under the assembly can collect a small amount of debris that might leak during flip. A tube that moves liquid is difficult to clean. System must not produce rotting food smells after cooking.

Juice has more flavor if it is smaller in quantity, and less flavor if diluted in water where it eventually becomes a soup. End user requests a position on this continuum, and machine then specifies how much liquid to add.

System must detect contents (e.g. camera, measure weight), and use that information to determine temperature profile (temperature vs time) for each item. All food does not cook at the same rate (e.g. carrots vs chicken); therefore, system must pump in varying amounts of heat to insure that all foods finish at the same time. The end user is *not* manually pre-cooking in a different chamber to pre-soften items with longer cooking times. When testing, assume you have one main course and three sides (e.g. chicken, carrots, potatoes, and broccoli) in primary chamber. For ideas, type into google: glass container with partition. The hint to the right may or may not be helpful.



Containers (e.g. Pyrex top and bottom) must be easily washed in dishwasher.

System communicates to App on smartphone and App contains user interface.

There are two contests. Each is 24 months and each involves a $.5M price.

Contest #1) Design system on computer with analysis, yet no prototypes. This involves mechanical design and extensive simulation, thermal analysis, manufacturing cost models, energy consumption models, electrical schematic and PCB design & simulation, and written reports. Applicants submit reports, mechanical drawings, simulations, and analysis for review. Also, applicant writes Android or iPhone App to demonstrate User Interface. All application material is made available to the public under a free gnu public license.

Contest #2) Teams create and tests a physical prototype based on any of the material submitted in Contest #1. Contest #1 participants are partially different from Contest #2 participants. For example, Sally wins Contest #1 and Debra wins Contest 2 with a design that is based partially on Sally's work. All materials (drawings, software, analysis, reports) are made available to the public under a free gnu license. Teams are judged partially on flavor. If flavors mix while cooking, end user experienced is often enhanced which causes judges to ruminate, "Is *this* chicken the $1M chicken?"

Summary

Manhattan 2 sponsors contests to encourage innovation and to encourage both universities and industry to become more involved in developing energy saving new technologies.

Chapter 34) Manhattan 2 Plan & Strategy

Manhattan 2 Plan

Our plan is to implement the following in stated order:

1.	*Blueprint 1.0*	Publish list of projects in book ("Blueprint 1.0"), website & PowerPoint presentation. ✓
2.	*Headquarters*	Establish Manhattan 2 Headquarters (HQ) at a private facility or university.
3.	*Blueprint 2.0*	Key personnel rework and reissue Blueprint 2.0.
4.	*Recruit*	Request CV's and 1 to 3 page proposals from professors, PhD students and postdoctoral fellows willing to deliver on Blueprint 2.0.
5.	*Directors*	Select Board of Directors.
6.	*Funding*	Raise initial $25M seed capital and solicit commitments for full funding.
7.	*CEO*	Conclude search and hire CEO.
8.	*Staffing*	CEO prepares staffing plan and hires key personnel.
9.	*Commitments*	Trigger letters of commitment to fund 5 year program.
10.	*Innovation*	Design, simulate, create cost models, and study multiple approaches to each problem. Structure the method of sharing intellectual property to the public on free gnu license basis.
11.	*Prototypes*	Build Feasible prototypes as necessary.
12.	*Systems*	Build systems that involve multiple components.
13.	*Field Test*	Deploy systems into field for testing.
14.	*Verification*	Insure systems are working properly with comprehensive testing.
15.	*Standards*	Propose standards to standards body for components and their interconnections.
16.	*Spawn Industry*	Companies manufacture system components.
17.	*Support*	Support industry engineers who build on top of MA2 reference designs.

Manhattan 2 Global Strategy

Manhattan 2's strategy for solving the global warming/limited fossil fuel problems is as follows:

Run Out	The world will run out of coal, oil, and gas over the next 100 years; *and* burning of fossil fuels contribute to global warming.
Replace	Replacing today's energy infrastructure will take decades.
Plan	The world does not have a plan for how to create massive amounts of non-carbon energy at reasonable cost; and how to reduce energy consumption.
Race	We are in a race to *Replace* before *Run Out*; yet we need a *Plan* before we begin.
World's Plan	Manhattan 2's goal is to create the *World's Plan* for how to build a low carbon society.
Accurate	Plan must be accurate; therefore, all gadgetry in plan needs to be prototyped, extensively tested, extensively studied for longevity via stress tests, and accurately cost modeled.
Low Cost	Gadgetry in plan must be *low cost*; otherwise it is ignored and plan is ignored.
Low Risk	Plan utilizes existing technology and is therefore low risk. Technology that a physicist might develop at a date unknown is considered high risk and is avoided.

Chapter 35) Manhattan 2 Organizational Structure

How to Innovate Quickly and at Low Cost

MBA schools have studied productivity levels associated with different organizational structures. They have found that a few engineers do most of the work. Some MBA professors refer to the "three percenters" as the 3% of engineers who do 97% of the work.

Who are the three percenters and how do we recruit them? They typically have the following attributes:

- Dislike bureaucracy which often wastes their time.
- Annoyed with managers who control their lives yet do not understand their work.
- Enjoy working with other bright engineers.
- Desire a path by which they can be highly productive and useful each day.

Small Group of PhD's, No Management

A small group of highly skilled engineers and *no* non-technical managers above them tend to be very productive. An example is the university professor with several experienced industry engineers on sabbatical, postdoctoral fellows and PhD students. This structure is as the core of Manhattan 2.

Tuxedo Park

In 1939 Alfred Loomis set up a laboratory at a beautiful mansion in Tuxedo Park, NY; and invited scientist to visit and do research at this location. He attracted great talent, including Albert Einstein, and contributed to the development of nuclear science and radar.

MA2 is Tuxedo Park

Manhattan 2 offers MA2 engineers one room in a large beautiful home near campus to help attract outstanding talent.

MA2 Engineers can work out of a mansion, or on campus, or both. MA2 rents multiple nice homes, close to campus, an example of which is to the right.

Each house is encouraged to sponsor a weekly evening activity. For example, Monday is poker night at Robotics House, Tuesday is BBQ chicken night at DIF House, and Wednesday is Burger and Scientific lecture night at GCF House. Heavy drinking is discouraged since engineers need to focus on their work.

An MA2 engineer might prefer to live in their own home, which is acceptable as well. They can still attend events and work at a mansion during the day.

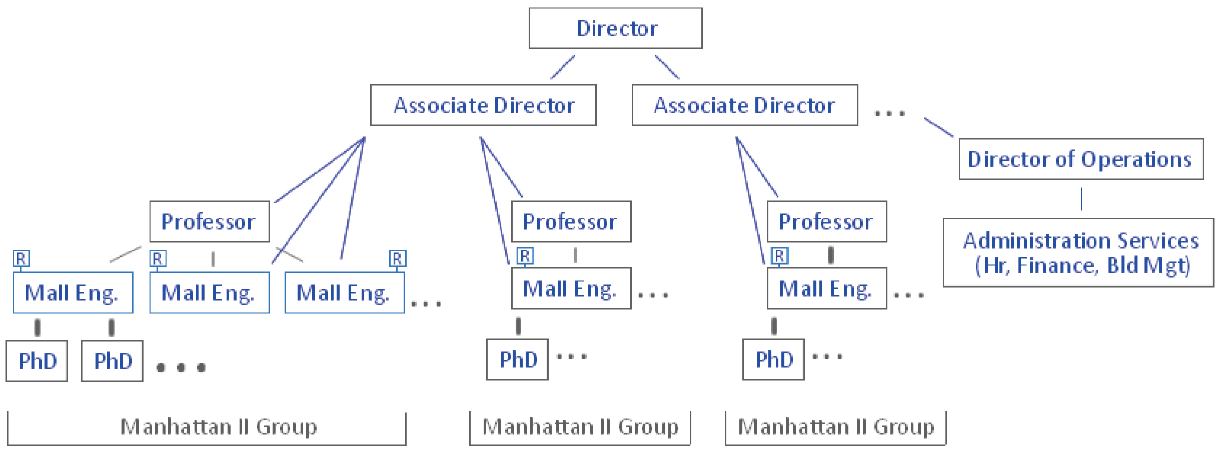
Example Homes near Campus

Below are examples of nice homes that help to attract outstanding talent.

- 1332 Westhaven is $20K per year person (10*12/6) is close to Caltech in Pasadena, California. If one likes to work outside under a tree, sipping ice-tea with fresh lemon, in short sleeves, they might find this fetching.
- 3800 Barton is close to UT Austin and cost $14K/person/year (6*12/5).
- 1724 Pelham is close to Georgia Tech and cost $14K/person/year (6*12/5).
- 6 Kirkland is close to Harvard and MIT and cost $31K/person/year (13*12/5).

Manhattan 2 Organization Chart

The Manhattan 2 (MA2) organization chart is as follows:



The Director and Associate Directors are engineers who have a fantastic amount of experience developing products and systems. They are not just managers -- they have done actual engineering themselves.

PhD students take classes in their first year and are often not productive until their 3rd or 4th years; subsequently, MA2 favors PhD's after their 2nd year. Postdoctoral fellows have their PhD's and are known (one can observe previous work); therefore they are less risky and can handle more responsibility.

To insure professional quality, at least 2/3 of each team consists of experienced engineers and postdoctoral researchers, and ≤ 1/3 are PhD students.

Administration (HR, Finance, and Building Management) reports to the Director of Operations. Administration is a service to AD's and the Director. If an AD or Director is not happy w/ an individual in Administration, they are to report their concern to the Director of Operations. The Director of Operations has the authority to hire and let go people underneath him or her. MA2 does not hire or fund marketing, sales, public relations, and product management people or services. The Director of Operations and his/her people are not responsible for funding.

Manhattan 2 Group

MA2 Group	- University research laboratory overseen by Professor
MA2 Group Leader	- Leader of Group, which is typically a Professor.
MA2 Engineer	- PhD student, Postdoctoral fellow, or industry engineer on sabbatical

Associate Directors

Associate directors (AD) create MA2 Groups, hire MA2 Engineers (post docs), and manage Groups. AD's have the authority to stop the funding to a group or stop the funding to an MA2 Engineer at any time.

MA2 Sabbatical Program

The MA2 sabbatical program allows engineers from selected companies & institutions to take a 1 to 36 month sabbatical and work for MA2. Also, they are offered one room in an MA2 mansion. MA2 groups select engineers associated with their work. For example, an MA2 heating/ac group invites a talented Honeywell programmer to participate. While on leave, MA2 reimburses parent company for salary and benefits. Participants can withdraw at any time, and be let go at any time. Also, they can work one week at a time with no commitment, and adjust as needed.

MA2 Mansion Donation Program

If a mansion owner has an underutilized facility and they want to help the MA2 program, they can make it available to MA2 engineers on their own terms. At any time, they can have any or all engineers vacate.

MA2 Reviewer

Each MA2 Engineer has an outside reviewer who is not associated with their university. This is denoted with an "R" in the above illustration. If the Reviewer concludes an MA2 Engineer is unproductive, it is their responsibility to report their observation to the AD and Group Leader. Reviewers are (supposedly) talented individuals who also provide guidance.

Transparency leads to Trust, and Trust leads to Adoption

MA2 engineers designate one folder on their computer as a repository for all MA2 work. This includes source code, analysis, reports, test results, designs, etc. On a 24 hour cycle, all files in this folder are synchronized to a read-only server and made available to the public on a free gnu license basis. We do this because we want industry worldwide to make use of materials developed by MA2 and we want them to accept standards developed by MA2.

We want countries throughout the world to reduce carbon emissions. If they are suspicious of a USA led effort, they might ignore their work. Transparency and free gnu helps to alleviate their fears.

Some researchers and universities might be uncomfortable with the MA2 level of transparency, yet this is the MA2 program. MA2 is not for those who want to own IP, file patents, or start a company with proprietary technology.

Webpage for Each MA2 Engineer

A webpage is provided to each MA2 Engineer. This enables them to publish their work, host a blog, and provide a link to their files.

MA2 Begs MIT BOD for W41

If MIT decides to be heavily involved with Manhattan 2, then the MA2 engineers might beg the MIT Board of Directors to flip them W41 for 5 years. This is an underutilized warehouse on the MIT campus that could quickly be set up for MA2 projects, including a prototype solar material production line.

Budget

MA2 spends a total of $1B over 5 years. MA2 does not spend money unless it is beneficial, and therefore spends less than $200M/yr while ramping up.

Typically, a sponsor pays a university ~$100K/yr for a post-doctoral fellow ($65K/yr salary to post doc and $35K/yr overhead to university). MA2 might spend another $100K/yr for all other MA2 related expenses such as MA2 overhead (Director, AD, Operations), prototyping, software, travel and rent. Therefore, the total fully loaded cost for each post doc is ~$200k/yr. PhD students cost less, and engineers on sabbatical cost more. On average MA2 might spend $200k/yr salary + overhead for each MA2 engineer; therefore $200M/yr average corresponds to 1000 MA2 Engineers.

Funding

MA2 is funded by a government, a foundation, or an Individual.

Conclusion

This document is a path to a path. It describes specific projects for engineers to pursue over 5 years. And hopefully the results of these efforts tell us how to responsibly replace our existing energy infrastructure with sustainable low-carbon technology.

Chapter 36) Appendix I - Further Considerations: Low Cost Geothermal Drilling

Below are more comments on how to install underground pipes that connect chilly water to a heat pump. This topic was discussed previously, and is continued here.

Radar, Rocks, Inspectors and Website

A rock can easily damage a trencher. This is costly in addition to irritating. To reduce this risk one would first need to drive along the proposed path with a Ground-penetrating radar that can see rocks and tree roots. If one sees an obstacle, they might be able to adjust the path. In some cases, trenching is not be possible due to too many obstacles.

The trench designer would want to budget much space and money for radar due to the cost of the rock. A 1x1x0.3 meter (3x3x1ft) box with $10K of electronics might be reasonable. The radar designer might protest and say they doesn't need this level of support; yet due to rock cost, they might. For details, click here.

The city/county building inspector might want a radar report before providing a permit to dig; therefore one might need a jeep-like vehicle w/ a radar to visit, scan the site, upload data to website, and drive away. To reduce the cost of this scanning, one would need an automated system to manage this process. This would be a website where one specifies their order, submits a credit card, gets physically scanned by a funny looking jeep within 14 days, and views data online. This preliminary feasibility project might cost $250 and would be done before deciding on the main geothermal project. To keep this cost low, the guy w/ the jeep would need an App on their smartphone that quickly routes them from one job to the next. Radar is obscured by water; therefore a feasibility scan would be done when the ground was dry. The App on the phone would instruct the jeep operator when to chill, no pun intended, based on weather reports.

The preliminary report and google earth image/map data would help determine a cutting route that is approved by the city/count.

Radar Scan Website Automation

The only way to make these cost numbers work would be a website that automates the feasibility radar scan, permitting by city/country, scheduling, implementation, and payment. The website could be like Uber where you have people driving around your community looking for work, you see them on a map, you click on one that is 28min away, and then you bite your nails hoping they don't make a big mess in your beautiful yard, which they will.

Next Steps

Many private companies will not develop an automated trenching machine due to too many risks. If a company sees 10 risks between today and revenue, they tend to pass. Example risks are: "Can radar see rocks 4m down in many soil types?", "Can machine cut two trenches a day and not break often?", "Can machine move around small yard and not accidently bump into things?", "Is website bug free and managed by talented people?".

To move industry forward, MA2 research first needs to clear these risks.

Step 1: World Innovates With Contests

During the first 12 months the world is encouraged to design machines in software (e.g. Solidworks, Inventor), run simulations, develop accurate cost models and write reports. To move this forward, Manhattan 2 sponsors a contest with a $1M prize. All materials submitted are made available to the world under a free gnu license.

$1M is considered a lot of money for a contest, yet this phase is important and not a time to be cheap. Tens of thousands of these machines might be in service ten years from now, and we want to head in a decent direction early. Top mechanical, civil, and electrical engineering professors select a winner.

Step 2: Build Prototypes

The next step is to build prototypes of the top three designs. One would start with a list of things that need to be done and the number of researchers needed for each item. The winner of the first innovation contest may or may not be involved in prototyping.

Manhattan 2 prototypes three different machines at $25M each; subsequently the total cost to demonstrate three different approaches is $75M.

Also, there is a contest where each entrant is a team that produces a working prototype. The winning team receives $1M which is distributed among team members in amounts determined by their team leader. The three Manhattan 2 teams enter the contest, yet also anyone in the world is invited to participate.

Excellent vehicle and robotics engineering universities are encouraged to participate. These include U. of Illinois at Urbana–Champaign, U. of Michigan at Ann Arbor, Georgia Tech, Carnegie Mellon University (CMU), MIT, U. of California at Berkeley, and Stanford.

CMU is good with robotics. Perdue and Michigan schools are good with vehicle design. Colorado School of Mines and Texas A&M know soil. Subsequently, Manhattan 2 encourages these schools to participate.

Top engineers from farm and earth moving equipment manufacturing companies are invited to participate in the MA2 Sabbatical program. This includes companies like John Deere, Caterpillar, Steyr, Case IH, Komatsu, Hitachi, Sany, Zoomlion, Terex, and Doosan.

John Deere is based in Moline, IL and Caterpillar R&D is based in Peoria, IL. Both are close to the University of Illinois at Urbana–Champaign. At least one of the design teams would probably be located in this area.

Conclusion

The world can reduce energy consumption dramatically if it can develop a low cost way to attach a heat pump to shallow geothermal temperatures.

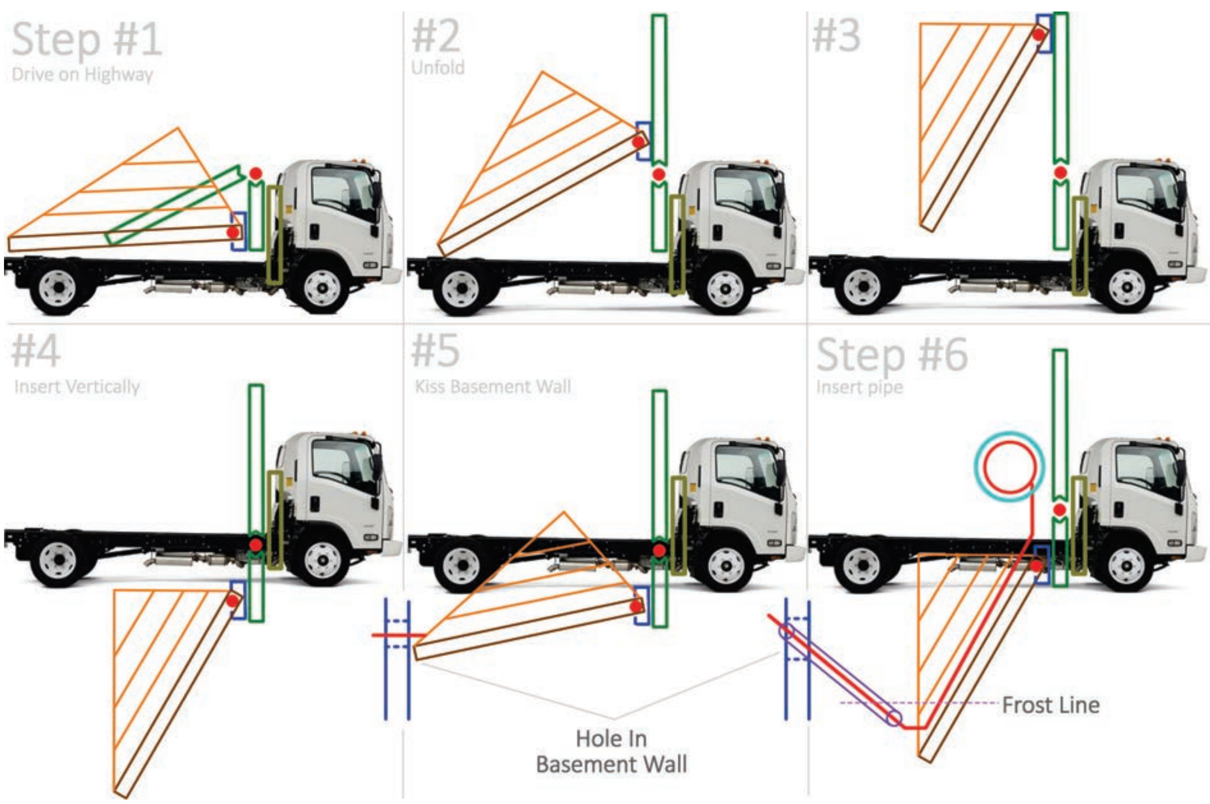
Chapter 37) Appendix II - Further Considerations: Advanced Chain Trencher

Below are more comments on the advance chain trencher vehicle, which was discussed [previously](#).

Need Low Center of Gravity While Driving

Load is associated with weight since one needs heavy metal to bear loads. And weight effects center of gravity, which must be low in order to keep the truck from tipping when rounding a curve. Our longest beam is 5m (15ft). Collapsing or folding is a bit complex; therefore one might consider rail and pivot, as illustrated below. This shows a General Motors ~$50k 6m (20ft) long [Model 4500](#) truck attached to a trenching machine.

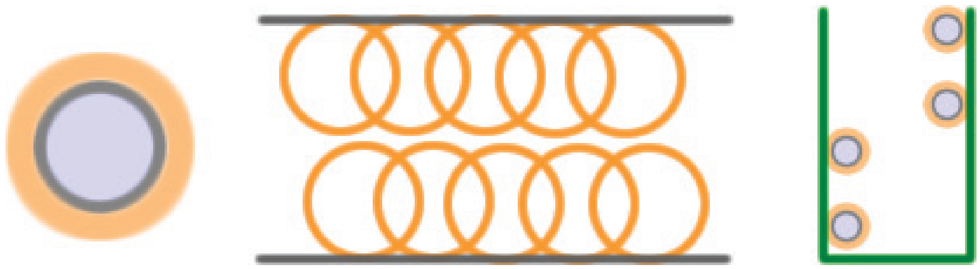
The chain mechanism (brown) mounts on a pivot (red), which attaches to a slide (blue), which moves along a rail (green). A secondary rail (gold) moves assembly closer to earth level. When one first begins, the cutting systems drives vertically into the ground. When moving on the highway, the 5m heavy beam (brown) needs to be low to the ground. Pipe is routed to hole in basement wall before machine begins digging trench, as shown in step 5. Water in the ground stops freezing at a depth ranging from 0.3 to 2m (1 to 6ft) depending on one's location (frost line). Larger pipe (purple) applies warm air from basement around pipe between hole in basement wall and 4m (13ft) depth to lessen risk of freezing.



Improving Pipe-to-Soil Thermal Conduction

Since pipe installation is done by a machine, we have more control over thermal conductivity, and we need to take advantage of it.

There are often tiny pockets of air between soil particles, and between soil and pipe. To help control this, we can surround the pipe with a layer of "interface" material by pressing damp material against each loop with a hydraulic press and mold before placing loop into the ground ("grout"). When soil is ultimately pressed against loop, this interface material deforms and provides a thermal connection between pipe and soil. In the below illustration, interface material is shown in orange and pipe in gray.



An example interface material might be sand mixed with a finer grain material. Alternatively, one might prepare interface material on site by grinding site soil slightly and then pre-pressing it against pipe, to reduce air gaps. Or blend site soil with a finer grain substance that is supplied by trenching machine.

The two right-most above illustrations show how one might place two rows of pipe, to reduce trench length.

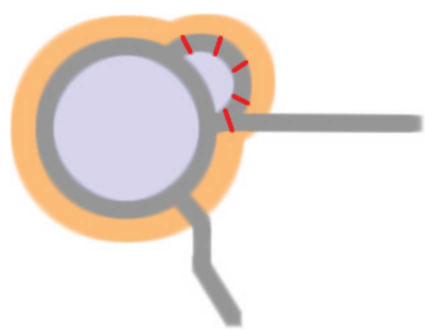
If the trench is 35meters (110ft) long and one surrounds 200 slinky loops of 5cm (2in) diameter plastic pipe with 0.5cm (0.25in) interface material, for example, the total interface material volume would be 0.4cubic meters (13 cubic ft.). If this material was sand it would weigh 700Kg (1500LBs) and be of manageable weight.

If machine completes two projects per day, it might load one cubic meter of interface material and one spool of geothermal pipe in the morning, and then visit two sites.

Insuring a good thermal connection between pipe and earth is this machine's responsibility.

Soil Moisture Control System

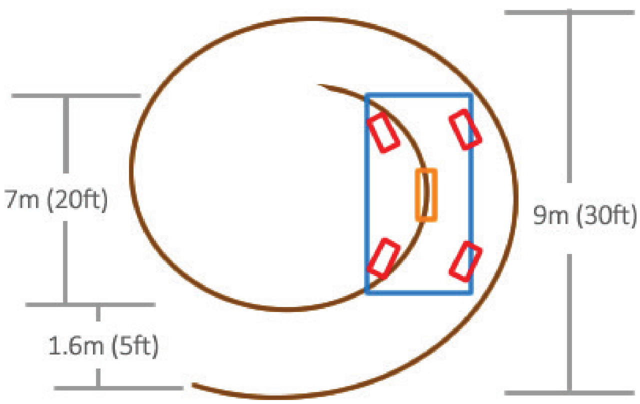
If one studies soil thermal conductivity data they will notice a big difference between dry and wet soil. Wet conducts heat 15 times better than dry. Subsequently one might consider adding a moisture maintenance feature, as shown in the below illustration. This involves a secondary pipe that is smaller than the primary, and contains small holes (red in below illustration). One could periodically place the secondary pipe under pressure to move water into the region around the pipe (e.g. 10minutes a day). In the below illustration, orange color represents coarse sand. This is important, because we want to control water flow and not have it escape via a random, least resistance path.



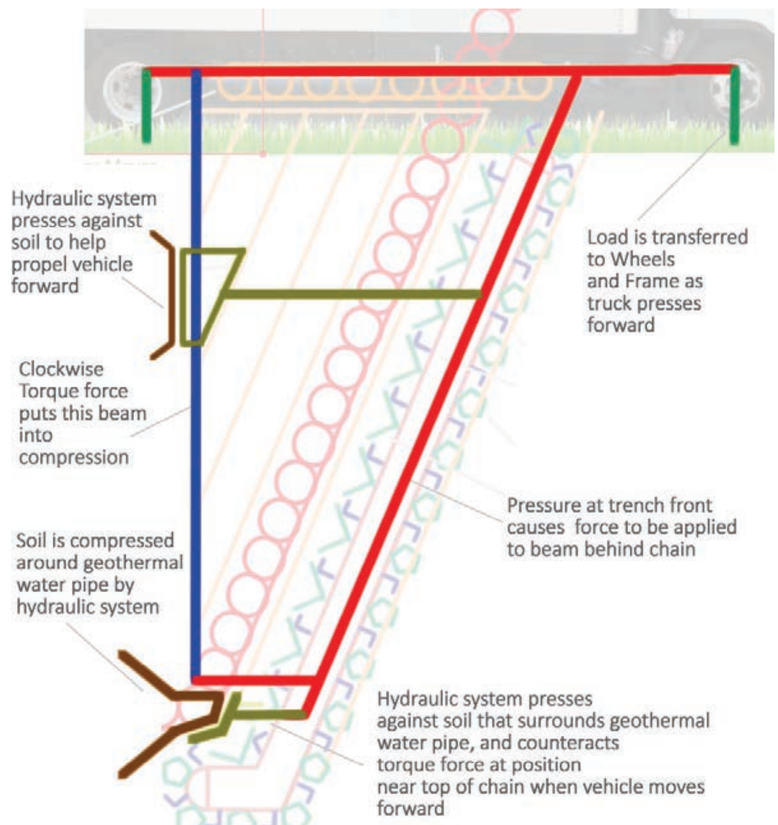
Manhattan 2 engineers need to create many different pipe concepts in software, simulate, create prototypes, and test.

Supporting the Standard Yard

A vehicle with 4 wheel steering (not just 4 wheel drive) provides a minimum turning radius that is ~20% more than the distance between the two axels (wheelbase). In our concept drawing, our wheelbase is 3m (10ft), therefore we could expect a 4m (13ft) minimum turning radius if all 4 wheels turn. We have approximately 0.5m from outside radius to center of truck, therefore we can expect 3.5m from truck center to center of smallest circle (7m minimum diameter). If we place 1.6m (5ft) between spiral arms, we can install a 35m (110ft) long trench within a 9m (30ft) diameter circle. The trench machine structure protrudes outward, therefore a total of 11m (36ft) diameter is required. If placed into a rectangle, we consume 121 square meters (1300 sq. ft.) which is 0.03 acres. The average yard in the USA is 0.2 acres, for reference. It turns out our 11x11m requirement barely fits into many USA suburban yards. In summary, this will fit most homes in rural areas, fit perhaps 50% of homes in suburban areas, and fit very few in urban areas. This machine could easily support commercial applications where a field of grass is available for trenching.



There are many variables that determine trenching requirements including air temperature, ground temperature, thermal conductivity of soil, and required heat capacity. For example, larger homes in very hot or very cold climates often need more than an 11x11m surface to spread out heat.



Managing Load Forces

The illustration to the right shows primary load forces, which is where one can expect mechanical stress. The structure needs enough metal at these positions to bear these loads, yet not too much to make the system excessively heavy. The system is designed to cut soil, not rock, and support a 0.3m (1ft) wide trench.

Typical Cost per Chain Trench

To keep cost low, one needs to keep these machines busy. If one can service two homes per day, for a total of 400 per year (100% utilization), then costs per home might be:

$200K capital equipment amortized over 5 years ($40K/yr, $100/trench)
$50K/yr maintenance ($125/trench)

$120K/yr operator salary + benefits ($300/trench)
$120K/yr company overhead salaries + benefits ($300/trench)
$120K/yr other company costs ($300/trench)
$750/trench for plastic pipe material

$1.9k/trench total (100+125+300+300+300+750)

This assumes machine is busy 100% of the time. If it was busy 75% of the time, then this would be $2.5K. If an individual owns the machine and operates out of their home with no overhead, then costs would be lower.

Grind While Driving

The trenching machine utilizes ground penetrating radar and GPS to aid in navigation.

If chain trencher hits a rock, sensors try to detect this (e.g. accelerometer detects vibration), yet damage might occur before detection (due to not enough signal in noise). If teeth along chain are damaged, they can be replaced, and therefore one has a remedy in the event of damage.

While driving on the highway, a camera and grinding wheel inspects and sharpens cutting teeth. They need to be controlled since their state effects trench cutting speed, which ultimately effects cost. Also, one needs to cut tree roots, to some extent, and dealing with this requires controlling teeth. Also, the camera can inspect each tooth while trenching, with a blast of compressed air before image capture. If camera sees damage, then it means it hit something and needs to back up.

One could also place camera and light on chain, and use that to inspect trench front surface.

Chain Trencher Prototype Design Team

An example way of dividing up the chain trenching prototype construct project is as follows. Parenthesis () indicated number of researchers. This is not a final list, it is just one example. The best way to manage a project is to identify the brightest people in each area, get them money, and then get out of the way.

Truck Central Computer (13): computer & interface hardware (1), cameras/GPS hardware (1), software interface to radar (1), user interface for operator (3), software interface to website (1), interface hardware to internet via G4 (1), software to manage teeth inspection and cutting (2), software to manage cutting (1), sensors hardware (1), interface software to sensors (1)

Mechanism (5): hydraulics (1), chain + teeth (1), conveyor system (1), channel structure (1), vertical beam & slider in truck (1)

Thermal Conductivity (6): design pipe, interface material (e.g. sand), hydraulics for packing, build & test prototypes (3), research moisture control system (inject water around pipe), build & test prototypes (3)

Radar (8): analog electronics (1), digital electronics (1), internal software (1), digital signal processing software (2), user interface software (1), website user interface for viewing (1), database backend to manages images (1)

Truck (4): truck frame (1), road vehicle standards (1), 4 wheel steering hardware (1), 4 wheel steering software (1)

Website (3): design (1), user interface (1), database backend (1)

Chapter 38) Appendix III - Further Considerations: HDPE Pipe Insertion

Below are more comments on how to insert HDPE pipe, which is done by the Underground Robotic Gimbal Chamber and the Vertical Bore Hole machine, which was discussed previously.

Complexity is Free

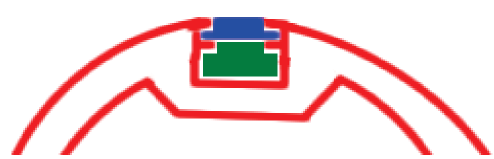
There are several techniques for drilling and we will explore one of them. To the right is an example HDPE flexible plastic water pipe that is 8cm (3") in diameter. With HDPE extrusions, one can produce a complex shape for the cost of the plastic material. Complex is free. This contains an internal pipe (brown) for outgoing water, and a secondary outer sleeve for return.

A low cost permanent PVC head (blue) is glued to pipe front. The head manages drilling water out (orange) and dirty drilling water return (light green).

Four internal tiny control pipes (purple) help steer head by applying pressure as needed via holes that are close to head (yellow). When finished drilling, internal head termination pipe (gold) applies pressure to invoke spring loaded gadget that seals drilling head and allows cooling fluid to flow normally. Also, after drilling, grouting fluid (e.g. cement) is pumped into system via four internal control pipes (purple, yellow).

There are limitations on how far one can push HDPE plastic pipe forward and this determines how many pipes one needs. If one pre-drills with a steel auger and then pushes HDPE pipe, they can apply less mechanical stress on pipe. This increases maximum length and also decreases aging process. Another approach is to place a low cost metal auger bit at head and rotate this with circulating water.

Cement-based grout under pressure is messy. It is difficult to clean, can damage equipment (after spraying and hardening), and is often rejected by regulators. Subsequently, engineers explore alternatives. In the below concept illustration, the HDPE pipe wall includes a conduit by which one can pump water mixed with a sand-like material (green in illustration) to thermally connect pipe to soil. A metal strip (blue) is withdrawn slowly such that the material (i.e. grout) exits the conduit first at the pipe end, and then gradually works its way toward the origin as the metal strip is slid back by a robot.



Three of these features spaced every 120° allow control over direction while drilling forward, and then they are used to install earth friendly grout at low pressure upon completion. Will this work? Friction would limit pipe length, obviously. MA2 study this technique and variations of this technique. What can we do w/ robotics and complex HDPE pipe shapes?

A variety of devices affix to steel gimbal in chamber including ground penetrating radar, HDPE pipe pushers, inspection robots, and interconnection pipe robots.

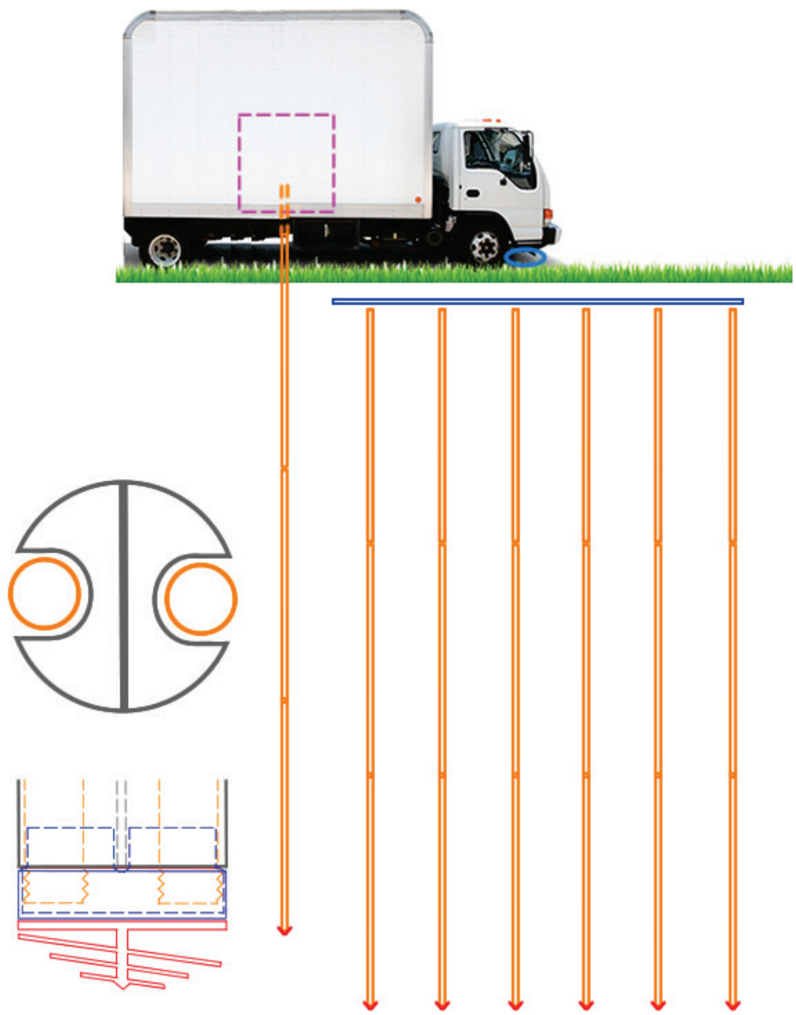
If all materials and tools reside on flatbed truck and are accessed by truck-mounted-mechanical-arm, then operator's role is minimal (i.e. there is nothing for operator to touch). Subsequently system requires only one operator, which dramatically reduces cost.

If the permitting system with the city is automated, costs are reduced.

Also costs are reduced if standards govern the mechanical arm, access-hole diameter, robots that insert into this hole, chamber gimbal structure, and underground piping. Given standards, industry can make robots that fit into all access holes and all chambers; which encourages participation.

Several advantages of robotic chamber approach are:

- If one pipe fails, robot can identify with testing and bypass.
- Almost no tree roots at lower 6m depths.
- Clear cutting from above, as done in concept #1, consumes much energy. With robotic chamber, one consumes energy only once digging to a lower depth.
- Supports lower depths, such as 6m to 8m.



Is this approach difficult and expensive, or can engineers figure out how to make it work and at low cost? As a next step, Manhattan 2 engineers describe this system in software (e.g. SolidWorks, Inventor), improve it, simulate, improve further, create cost models, write reports, give talks, and travel to different sites interested in this topic to discuss.

Same Concept yet Apply from Above?

If we devise a method for pushing 3" diameter HDPE coaxial pipe (e.g. 10 pipes each 60ft) directly into soil from a gimbal 20ft below the ground, can we do the same yet from above, as illustrated to the right? MA2 engineers study this "variation" of the gimbal approach.

Instrumented HDPE Pipe

One could have a channel within the HDPE pipe for sensors, shown to the right in purple. These could be used with prototype and/or production systems. Monitoring options include:

- Study thermal conductivity between pipe water and soil given different grouting techniques (i.e. measure temperatures).
- Monitor temperatures along water path to determine how heat is moving throughout the system.
- Measure acceleration, stress, and/or load during drilling (e.g. from vibration).
- Look for water leaks within system.
- Look for ground water outside pipe.

One strategy would be to place an instrumentation PCB in the drilling head and have it measure acceleration and load during drilling; and measure pipe water temperature during normal operation. If one wants to measure soil temperature along pipe, for example, they could stop pumping, let sit for several hours (for water to acclimate to soil temperature), and then move water slowly past temperature sensor (e.g. 1meter every 20seconds) to determine approximate soil temperature along length of pipe.

One could also place an instrumented PVC device at the top of the pipe, perhaps 1ft below ground level, to monitor temperature and connect insertion pipe to daisy-chain pipe.

Conclusion - HDPE Pipe Insertion

After refining various concepts, several are selected for prototype. All materials produced by Manhattan 2 engineers are made available to the public on a free gnu license basis to encourage industry participation.

To learn more about large commercial pipe insertion projects, click here.

Chapter 39) Appendix IV - Further Considerations: Wall Bus Modules

Typical Sensors

With the help of low cost additional components this $0.58 processor is capable of: temperature measurement, wind pressure measurement, smoke/fire detection, door/window state detection (e.g. burglar alarm), vacuum pressure measurement and voltage/current measurement.

Typical Control

With the help of an additional electronics, this processor is capable of controlling: 110/220VAC devices, motors (e.g. to open or close a window or door), lights, fans, or duct damper.

Wall Bus Involves Small Data on 100mSec Timescales

The network reads and writes to sensors and controls. An example sensor might be a temperature sensor, a wind pressure sensor, or a vacuum in insulation sensor. An example control might be to turn on a pump to further increase vacuum in insulation. In order to do measurement and control with a $1 processor and 10Kbps network, data packets are small.

Sleeping Nodes

One might have 30 nodes in a home, and the home owner probably does not want all processing continuously and drawing power. Subsequently, nodes power down to sleep when not in use. One of the requirements of a new communications protocol would be to wake a node when needed (e.g. to read a temperature). While sleeping, a node might draw 2 to 5mA of current. 5mA at 12Volts is 60mW, which is $0.10 of electricity per year. To facilitate waking the processor, the communications IC needs to wake up when the data+- wires move, read the first 8 bytes of each data frame, and ignore it if frame is not intended for self (otherwise, wake microprocessor).

Real-time Clock

Each node has an internal real-time clock which periodically wakes the node to complete a task (e.g. broadcast temperature once every 10 minutes).

Security

Physical switches on the Automation Motherboard PCB provide end user with physical control over security. For example, one can have a physical switch to Block or Allow modification of the Automation Box. If set to Block, then an attempt to modify is met with a message "you need to physically flip … switch in Automation Box to proceed…". If a rogue element obtains physical access to the Automation Box (e.g. you invite Tom Cruise to your Christmas party and he slips into basement), they win; therefore, a cover with a lock helps to control this box.

Audio Bugs

Wall Bus supports 10K bit per second data transfer rates between rooms and this central location. This is fast enough for rough speech yet not real-time video. Subsequently, a bad actor with physical access to Automation

Box and a room, can bug that room. For this reason, secure locations such as military, government and financial might avoid this system.

Old Construction vs. New Construction

New construction has an advantage over old construction since it can be wired for Automation (e.g. 4 wires between central location and each electrical box); therefore some of the technology being discussed here only applies to new construction.

Screw Force Is Great

Our Four Wire (two 18c and two 22c) Wall Bus connects 1 to 64 modules in parallel and carries up to 2Amps, with 99.999% reliability. Crimp contact connectors, pictured at the left, do not meet our reliability requirement and access to these inside a wall is limited (repair is difficult). Therefore, screw force is required. Screw force is an amazing thing. When one clamps down on a screw, thousands of PSI (LBs per square inch) is exerted on the wire, causing the metal to flatten slightly. The typical connector pictured on the left applies significantly less force at the interface. The cost of a 4 pin screw terminal, pictured on the right, is $0.16 (1k qty, China Terminal Block 15Amp #CY381v-381-6P-1-4. This is of the high quality machined type and not the less costly metal tab type.


Hot Swap & Walk Away Replacement

It is important to provide a positive end user experience, and this includes dealing with problems. One type of problem is a hard failure inside the Wall Bus controller (i.e. it breaks). To help remedy this problem, the end user should be able to buy a new controller from Amazon (e.g. $50), unplug the old, plug in the new, and walk away. Walking away without further end user interaction would require state be stored, and then restored when new module is inserted.

New Term: Plug & Walk

In the electronics industry the term "hot swap" refers to no permanent damaged when end user plugs or unplugs a connection while power is applied. USB is an example of this.

We need a new term called "Plug & Walk" which is similar, except one can plug in a new module with power applied, and walks away without further interaction. This is easy to do, provided the computer programmers are told they must meet this requirement, programmers have a place to store state locally (e.g. flash ram next to module connector), and they have someone to talk to when the module is first inserted to discuss where state is stored (e.g. small $1 processor next to controller connector). State includes things like a list of devices on the network and list of tasks that are implemented by the controller.

Plug & Walk Physical Label

All physical modules (e.g. power supply, motor in window module, android like device in central control box) are Plug & Walk compatible. To help with Plug &Walk, a one sentence message is printed on all Plug & Walk products. For example, one might see the following on a power supply: "To Plug & Walk: Replace with STWA-4-32-*Anything*".

99.999% Reliability

The wired telephone specification mandates that the system be 99.999% reliable (referred to as "five nines"). Subsequently an end user must be able to make a phone call 99.999% of the time, which works out to being out of service for 5 minutes a year. Let's assume that we are to meet that same standard. What effect does that have on our system? It turns out this helps us make decisions since many options involve worse reliability.

Radio is less reliable, due to a variety of issues beyond the scope of this document. Communication over power line is also unreliable as well. So we can rule both of those out.

Test labs for HomePlug power line communication observed 99.0% reliability in many metrics, missing our goal by a factor of 1000. Power wires also include noise from switching power supplies (e.g. 20 KHz square wave), noise from light dimmers (switching Triac), and noise from microwave ovens (> 1GHz). A building could easily have over 100 of these sources, all added together. Also, one needs to deal with multiple circuits connected together at a central fuse box (i.e. signal travels to fuse box and then out again). Imagine you have 5 dimmers and 20 switching power supplies on 10 different circuits that route to a fuse box. Your signal routes from one circuit to the fuse box, and then out another circuit. The receiver does not see a beautiful 60Hz sine wave; therefore, power line communication is not 99.999% reliable.

Batteries are not 99.999% reliable since they fail more often, so we rules these out.

This leaves one choice, which is wire to device.

$50 Tablet Controls Network via USB

The Wall Bus Controller attaches to the Automation Motherboard via its USB connector. There is a flavor of USB called OTG/DR, which means the automation motherboard can initiate a transfer (slave and master roles can reverse). An ASIC sits between the tablet USB connector and the Wall
Bus data+- wires to facilitate communication. Here's an example: the tablet sends a message to the ASIC that says "measure living room temperature" (100uSec), the ASIC in turn sends a request to the $1 PCB in the living room (25mSec) which responds with a temperature packet back to the ASIC (25mSec), the sleeping tablet is interrupted with a data packet from the ASIC via USB peripheral bus mastering.

Power Supply

Power Supplies are cheap. A 12V 30Amp power supply cost $40. So we don't need to be cheap with power.

A 12VDC power supply resides in the Automation Box (next to central fuse box). Also, it has a standard connector system and physical dimensions allowing end user to Plug and Walk. The standard units might be something like 20W (6" long x 5" wide x 2" tall), 33W (9" long x 5" wide x 2" tall), 50W (12" long x 8" wide x 3" tall), etc.

Power Management

If the maximum power to each circuit is 2Amps, and a motor that lifts a window draws 0.5Amps, then one cannot lifts too many windows at one time. Subsequently, the network needs to sequence large consumers of network power (e.g. lift one large window at a time).

Can we transmit data on 12V power wires?

We cannot transmit data over the same wires that we transmit power (i.e. cannot reduce above cable from 4 to 2 wires) because the 12V Pwr- wire might be connected to the 110/220VAC low side or ground, and this might contain significant noise which is coupled to signal (causing one to get power line communication like signal vs noise). The thermostat in your house has several dedicated wires, and is very reliable. We need that kind of reliability, which means avoiding uncontrolled environments.

Why not Wireless *or* Power Line Communication?

- Wireless and powerline communications is 99.0% reliable as seen in testing labs; whereas dedicated wire is 99.999%. Examples of existing systems with dedicated wire are HVAC thermostat and wired telephone -- both are ~99.999%.
- Devices need power, which entails wire, and adding another two wires for data involves minimal additional cost.
- Radio and powerline communications require millions of multiplication operations per second while continuously listening; whereas changing voltage on dedicated wire can wake sleeping $.58 processor at minimal cost.