William Kerr’s Lab Notebook

Table of contents:

BMS Heater and Cooler

Date: 11/2/2018

Meeting Minutes (11/2/2018):
Original Goals:

- Website (get up and running)
- Process power data on Microprocessor, send it to database
- Get the Power working

What Talmor Did:

- Wrote a working website that retrieves data from the database
- Wrote the database to store different types of data

Right now, an arduino is being used. It’s limited in its capabilities, but it can handle two system sensors at the moment.

We should pick a better one that can handle more memory, and has a better I/O capability (for the program to load onto)

One FONA chip should be able to handle all the traffic from all microcontrollers.
- 2G wireless chip should be more than enough
- We can combine packets

The main idea of the project was/is to get the students using this testing environmenta s far away from the sensitive electronics as possible. Asin, they should be able to configure a port on the website, then plug in the device to record data.

Date: 11/2/2018

Main Goals for Tuesday at 10:30AM:
- Get a good system-level design
- Get a good idea of the constraints

Ideas for improvement:

We don’t need 3 UARTs, we can use 1 UART, and multiplex it.

Obtain CMPE121/L textbook

For the UARTs and the analog sensors, what will the sample rate be?

If we have an 8-bit multiplexer, we can have 256 inputs.
Input 0,
Input 1,
…
Input 255.
Program a timer?

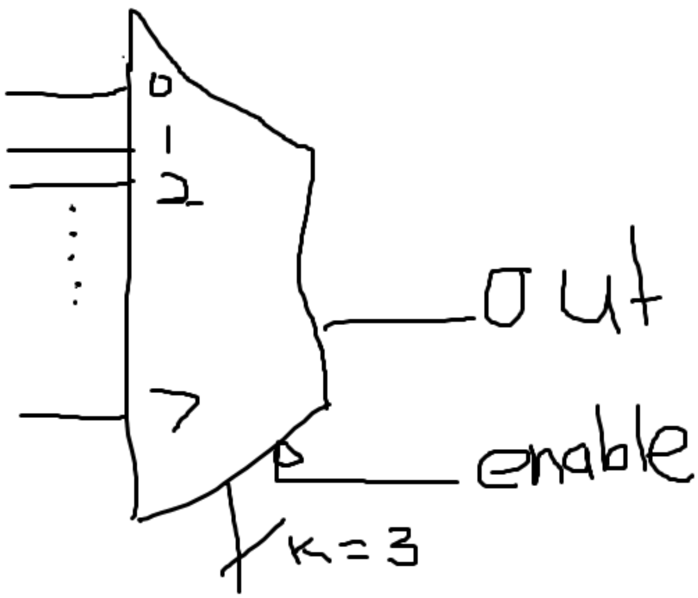
Date: 11/6/2018

Meeting Minutes 11/6/2018

- Review how multiplexers work, both analog and digital
- Review Petersen’s CMPE121/L notebook

Each of the blocks represents something.

Multiplexers and Decoders



Selector s is binary encoded. K = 3 inputs.
It’s like a rotary switch.
Enable is a tristate: active low.

Date: 11/6/2018

1.) Research how A/D pins in microcontrollers work
There’s only 1, though.
2.) Find some representative datasheets.

3.) Engineering Binder:
- App notes
- Datasheets
- Components

4.) Get Documentation in order

5.) Write a system-level diagram such that I can explain how it works to Petersen.

6.) Justify a 64-input multiplexer

Date: 11/6/2018

Find analog multiplexers from the internet, and download them.

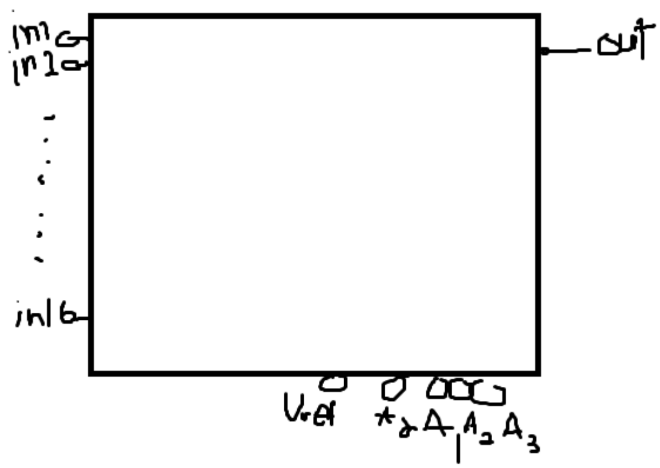
Date: 11/8/18

I found a 16-way analog mux. However, it only comes in SMD form

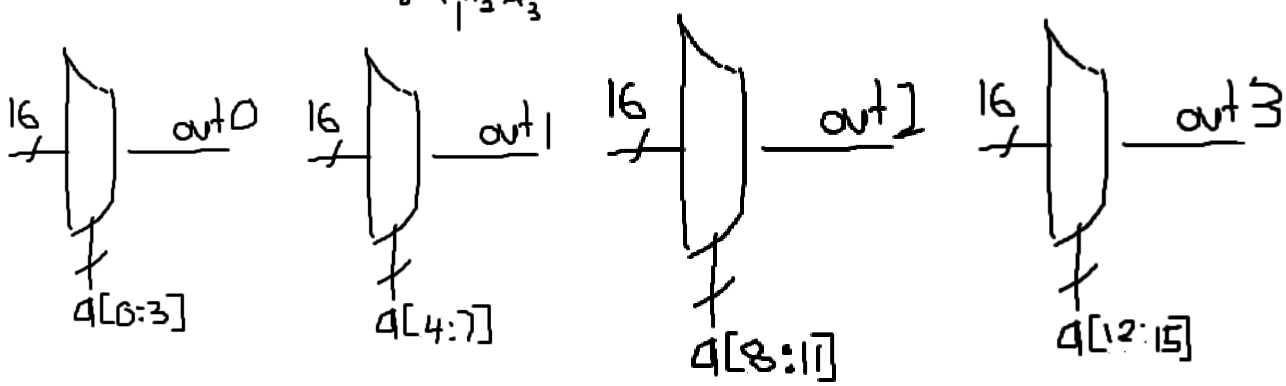
Model number: MPC506A

Expensive!!!
0.3us access time.

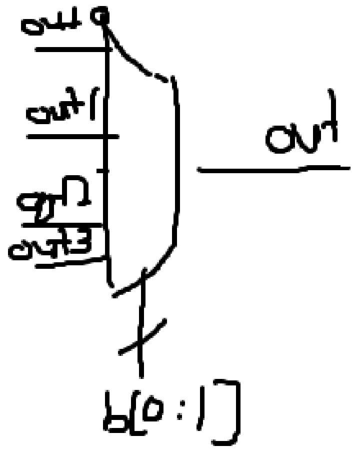
Package:



We want 64 inputs, so let’s model it as this (the mux):



Date: 11/8/2018



From Arduino:
a[0:15]: select each input 0x000: goes to mux 0.
b[0:1]: dependent on a:

```
if a >= 0 and a < 0x000F:
    mux 1, b = 1
if a >= 0x0010 and a < 0x00F0:
    mux 2, b = 2
if a >= 0x0100 and a < 0x0F00:
    mux 3, b = 3
if a >= 0x1000 and a < 0xF000:
    mux 3, b = 3
```

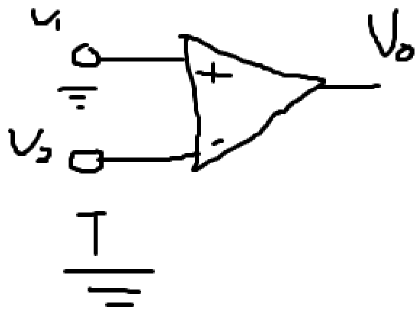
Date: 11/8/2018

8-way muxes are $0.50 each !
16-way muxes are $16.00 each !

Date: 11/13/2018

Differential Multiplexers

Analogy:



Vo = V1 – V2
Each of 16 inputs has 2 wires.

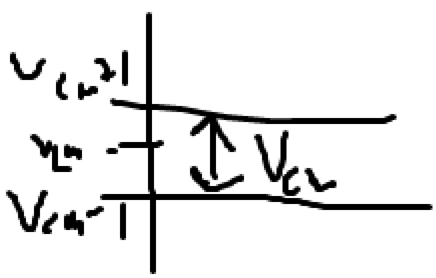
Single ended.
Single.
V1 =

Single Enede

V

V1

Teach yourself DCM vs Common
Common mode voltage
Vcm = (V1 + V2) / 2



Date: 11/13/2018

Adrian Knox?

Continue with familiarization with all relevant hardware ports.
Present the fruit of labor at next meeting
Dive into A/D devices
What I’m doing is research.

Research what the current blocks are.
Create subblocks for blocks if necessary.

Date: 11/13/2018

Have a good system level design written up.
Target for this quarter: get a good top-level system level diagram, with each block under.

Organize all your involvement here.

Make a complete circuit diagram.

For the first quarter, we must produce an interim summary report.

It’s a document you would give your boss.

Start an outline that you can fill in progressively as the quarter goes on.

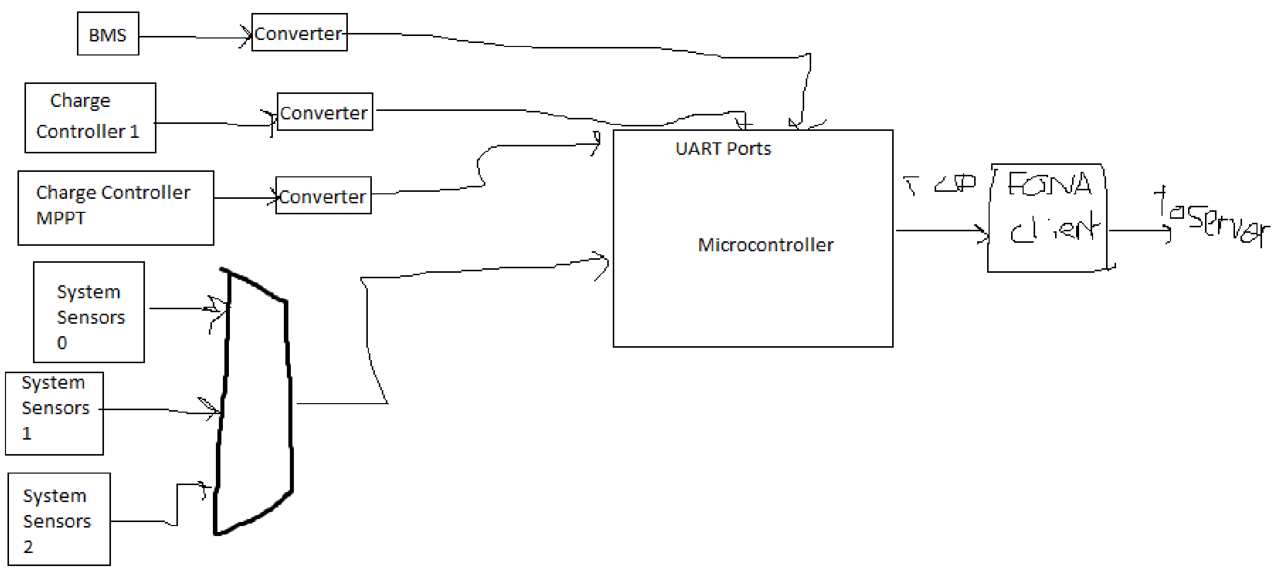
Date: Unknown

Date: 11/15/2018

See:
- J.J. for TCP packet programming advice
- Katiah Obraczka
- Professor Varma: has tips on A/D conversion

Date: 11/16/2018

Hardware High-level System Block Diagram



Date: 11/17/2018

Arduino ATMega2560 Specs:

54 digital I/O pins
(15 can be used as a PWM output)

16 MHz crystal oscillator.
USB connection
Power Jack (12V)
16 analog inputs.

256K flash memory
(8k used for bootloader)
13 built-in LEDs.
Length: 101.52 mm
Width: 53.3 mm

Analog resolution: 10 bits per pin
Measures from GND to 5V (default)
Has a software serial library for allow for serial communication

Date: 11/16/2018

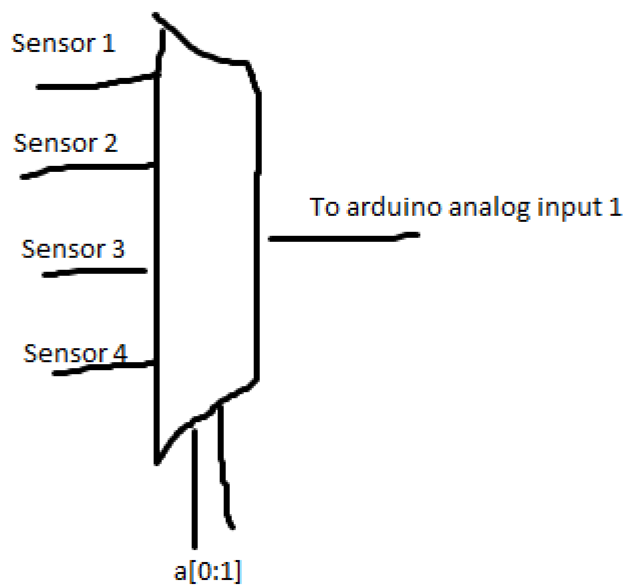
Greenhouse Requirements:

(Right now there are 16 different sockets. We will probably need more).

I think we could separate them into different stations.

We could have a 4-way multiplexer on each receiver.

Example:



Date: 11/16/2018

I found a more powerful microcontroller, also made by Arduino.

“Arduino Due”

Model Name: AT91SAM3X8E.

Input: 12V.
54 digital I/O pins.
12 Analog Input Pins.
512KB flash memory.
84MHz clock Speed.
Length: 101.52 mm.
Width: 53.3 mm.

Date: 11/16/2018

Raspberry Pi 3 Model B+

1.4GHz,
5V input,
1GB SDRAM.
Can load the python language onto its hardware.
No ADC chips on its board naturally, so we must buy one separately, or design.
What precision do we need for our ADC chips?

I found one that is a 10-bit resolution. MPC3002
I found another one that is a 10-bit resolution. MPC3008
with 8 channels.

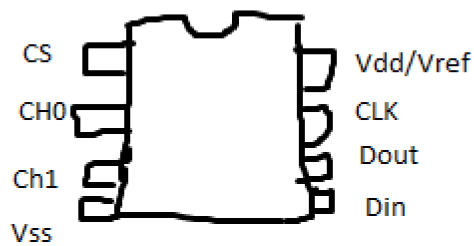
Found a website to teach me to wire an MPC3008.

https://projects.raspberrypi.org/en/projects/physical-computing/15

Date: 11/16/2018

MPC3002 chip:

75,000 samples/second
3.3V supply
10 bit
8 pins

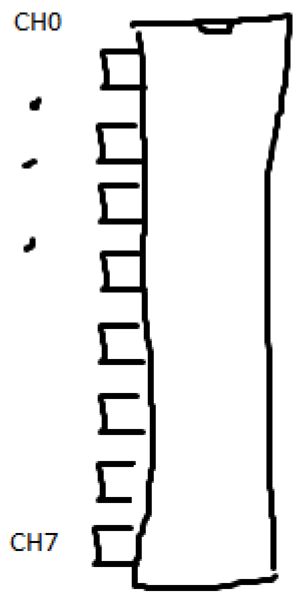


Date: 11/16/2018

MPC3008:

Sample Rate: 1.5 clock cycles
8 input channels
Let's choose the 8-way chip.

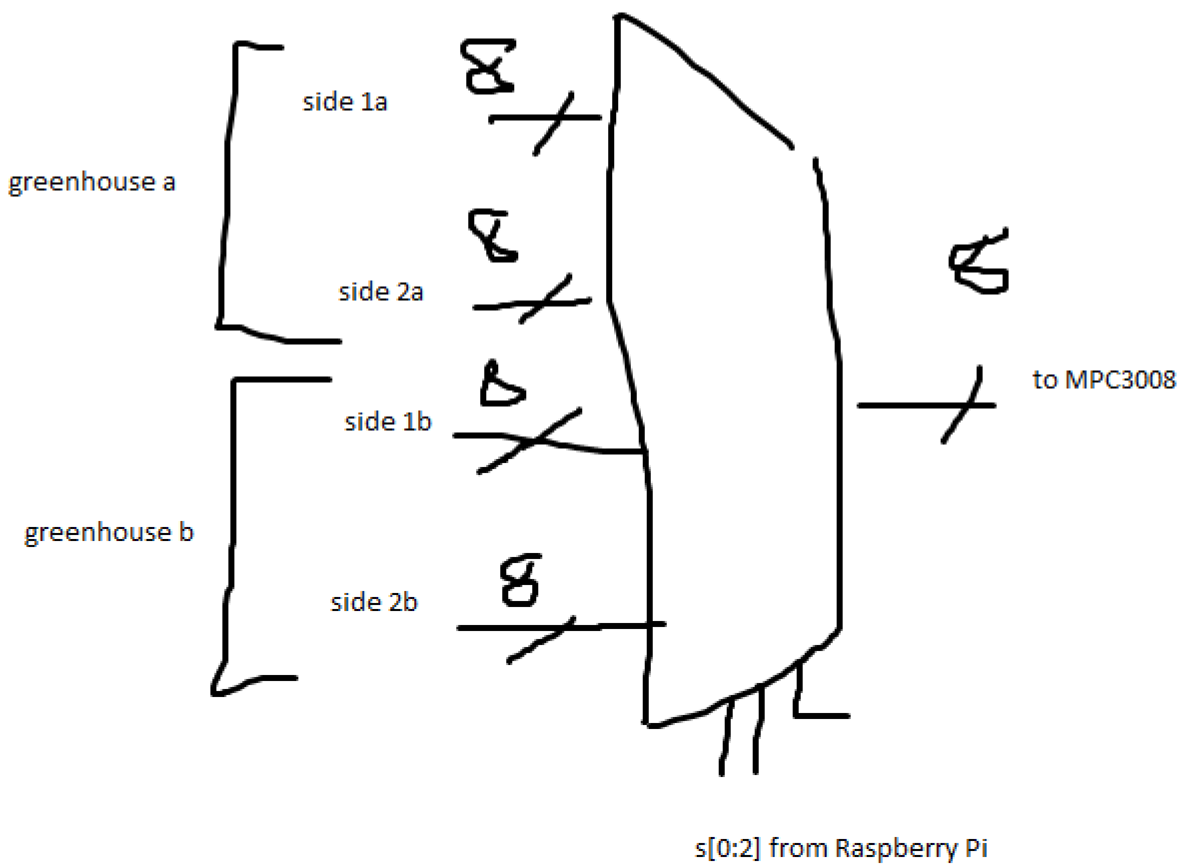
We could choose which side of the room to read from with a mux, and our Raspberry Pi can choose which side.



Date: 11/16/2018

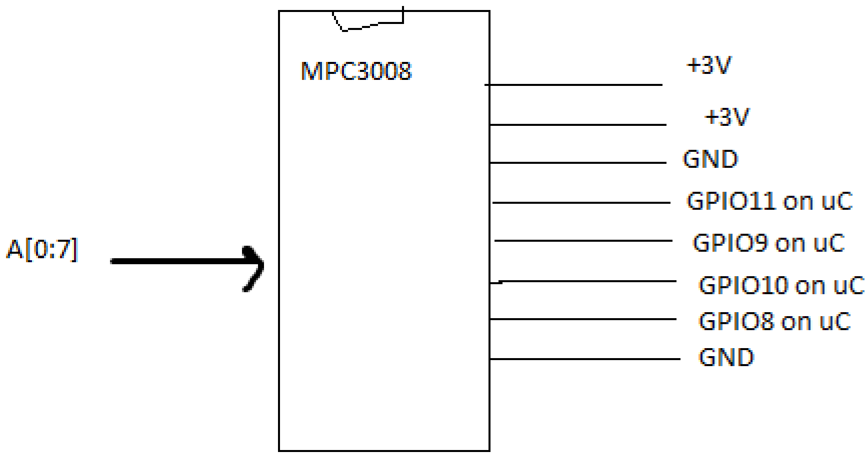
I don't know how to multiplex multiple busses though.

S[0]: GPIO2
S[1]: GPIO3
S[2]: GPIO4 (always high)



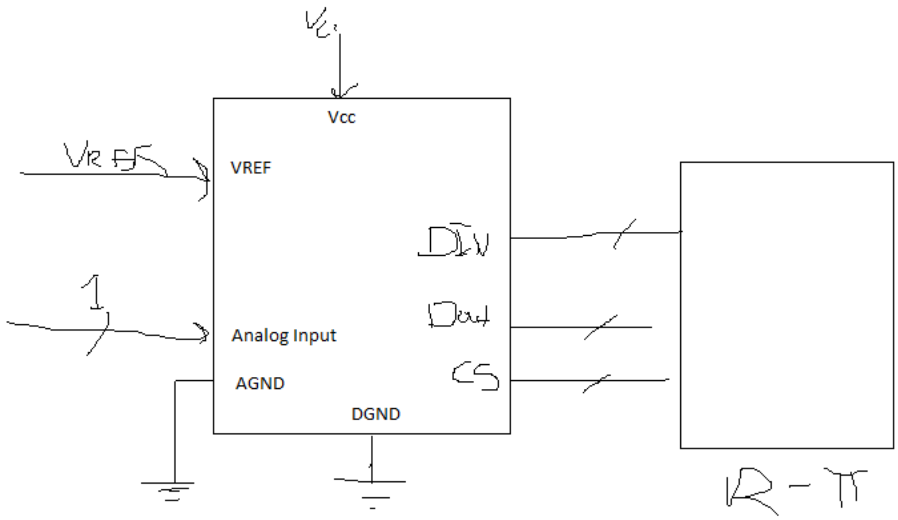
Date: 11/16/2018

MPC3008 setup for Raspberry Pi:



Date: 11/16/2018

MPC3008 setup:



on a PC board, the MPC3008 can have a I2C to USB bridge.

Do we need to…

Tradeoffs:
?

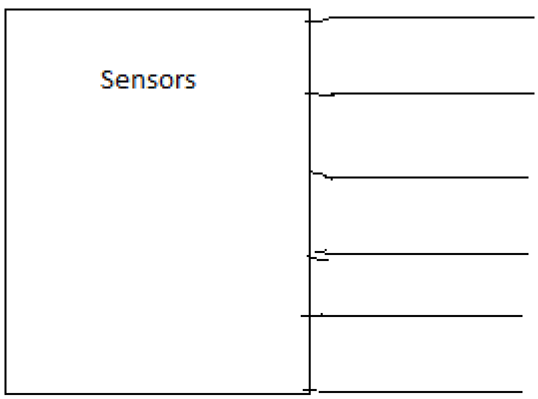
Date: 11/16/2018

How fast do we need the samples?
Quantization range: n = 10 => 2^10 => 1024

20log(1024/1) = 60dB

What resolution is the sensor?

1 scenario?



Date: 11/20/2018

TAKE A LOOK AT THE SENSORS!

Look at the programming side.
We have to program.

We want a full characterization of the system!

As the resolution rate gores up, sampling rate goes down.

Research the serial bus. They’re very popular.

Pin down what sensors there are, what resolution they require, and minimum sample rate!

Email tela. Get the specs from her.

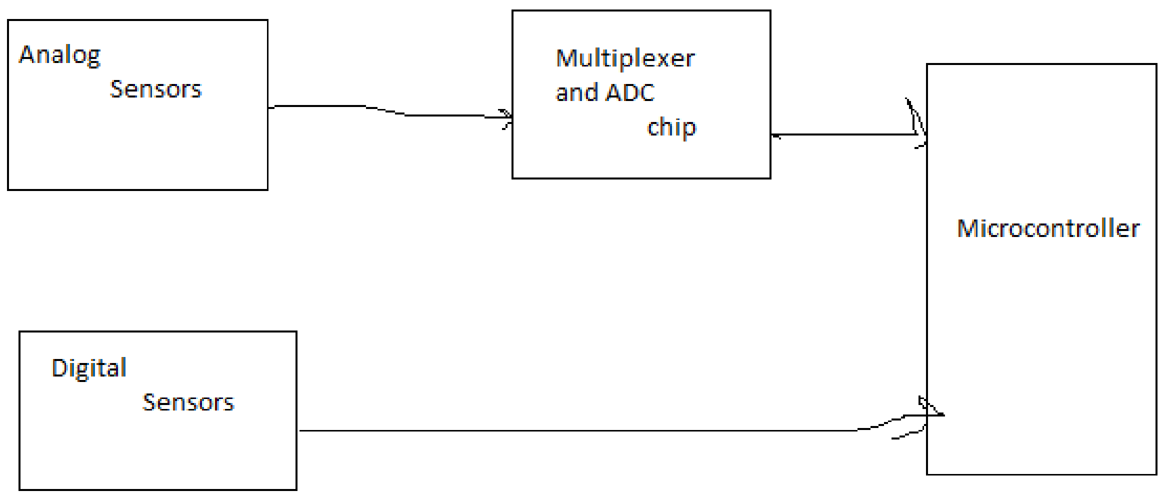
Date: 11/20/2018

Research what the sensor’s characterizations are.
Confirm with Tela.

I haven’t heard back from Tela. Here’s what I know:

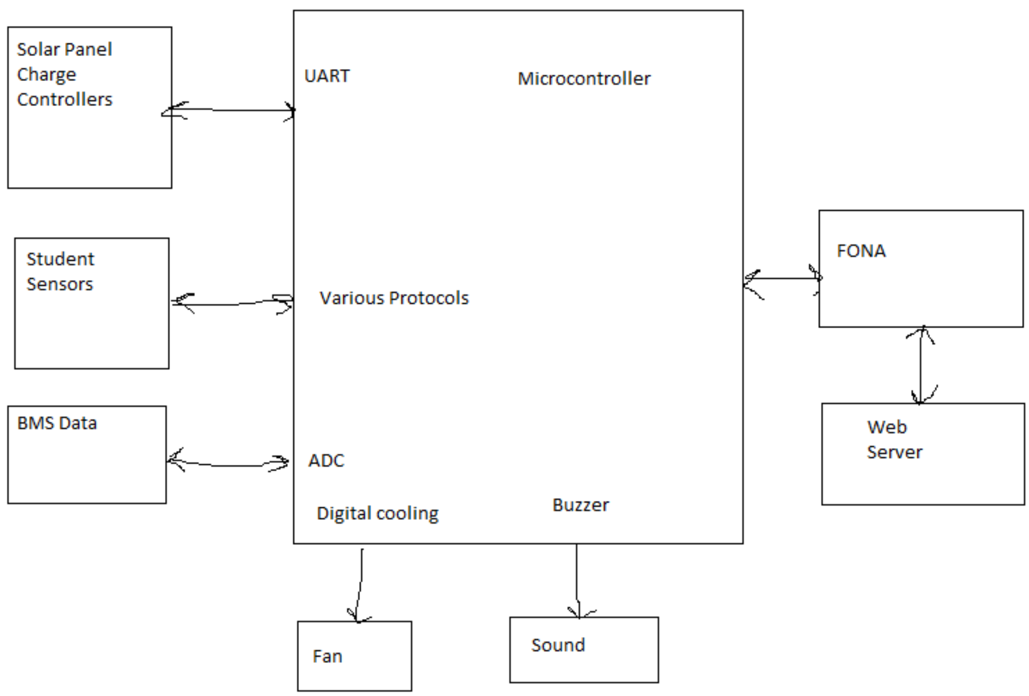
There are two kinds of sensors: those with ADC chips on them, and thosw without.

LUX sensor is one with an ADC chip on it.
Has an I2C protocol.



Date: 11/21/2018

[System] Block Diagram V0.2



Date: 11/22/2018

Rest server API:

Use http:

the ide:
Representational State Transfer

I’m

GET
POST
DELETE

interfaces
API

Use Standard APIs
Use something
SOAP is something we can use.
SOAP web server

Date: 11/22/2018

REST is stateless
SOAP is not. → Script

If I had a web form.

How to get the

REST

There’s an SDK C++ REST client library.
Lots of libraries I can use.
Libsoup
Libcurl.

Date: 11/27/2019

Over the Thanksgiving Break,
I was with my family.

I was able to do research in my down time.

I went to PETCO to survey possible types of sensors I could support.
(for their fish, gecko and turtle accessories)
They include:

Light intenisty
Humidity Sensor
Temperature
Water Level
Air Pressure
Water Pressure
pH Sensor
Hygrometer (measures moisture in the soil)
Voltage Sensor
Torque sensor (for the fan)
Power sensor (for the sensor’s power consumption)
Filter sensor (is a filter installed?)
Sound sensor

I realized that most (but not all) of the sensors you can buy online from our primary parts supplier (ADAFRUIT) are digital.

Also, there’s a cat near the greenhouse named Spencer. An orange tabby cat.

Date: 11/27/2018

All of these sensors, in one way or another, output data. So, on our website, we will have a web form that programs our microcontroller to actuate the appropraite port. We will have these inputs from our user.

- Port Number (we will label these on the greenhouse)
- Units of Measurement

I’m not sure if we will need these inputs, but I’m gonna have them in the advanced settings. If the user messes with these settings, the microcontroller will be programmed for it.

- Bit resolution: 10 bits (default), 12 bits, …?
- Sampling rate 5 samples per second (default)
- Analog or Digital?
- Digital only?
- What protocol used? (I2C, one-wire?)
- Maybe some custom script to potentially be added? (scanned for virues, and in python?) (Admin only)

Date: 11/27/2018

The problem is: will we require a digital sensor to be plugged in? If so, can we throw an error?

If not, how will our microcontroller tell the difference between an analog sensor and a digital sensor? I'm not sure how to approach this problem.

Date: 11/30/2018

Other questions I had (that haven't been answered yet):

- Will I have a lab to work in? (once I actually start designing the pieces) 24/7 access?
- How is and where is the website? (can I use the people.ucsc.edu domain while in development? Afterward, can I get it registered as something more fitting?)
- Will I have to connect the greenhouse to the power grid?
- Will I have to install hardware on the greenhouse myself, or will I have soimehelp.

Also, Talmor emailed me last night. He apparently released a series of tutorial videos for me to watch to get to know the website he built.

MUST HAVE:

- Track down tela: send her an email. Confirm sensors needed:
    - accuracy
    - resolution
    - interface type
- Learn how 1-wire interface works and how the I2C interface works.
- Justify having those particular sensors.
- Start with each temperature.
Current research shows:
- Digital temperature (water) sensor
- Digital temperature and humidity (air) sensor
- Lux.
Compare sensors. Choose 1 from a technically justified standpoint.
Next week:
- I investigated 3 or 4 sensors.
Understand each sensor; Report to Petersen about each sensor.

Date: 11/30/2018

(The light sensors)
If it uses an I2C inteface

Po:

Confirm if each sensor existing is OK.

Familiarize yourself with:
    Digital.

Form Factor:
    The light sensor form is crap.

Tela’s vision: student using these sensors.

We’re going to layout some PCBs when we get there.

Date: 11/30/2018

How do the sensors connect to the greenhouse?

Do we need more sensors than the three?

We’re doing a design review.

Tell her what the quarter goals are.

Right now, it’s hardware verification.

Date: 12/1/2018

Power Data:

We need that as fast as possible.
1 per minute, or faster.

Anything else: (students)
10 minutes.

Date: 12/1/2018

Buck converter (I don’t want to use a linear voltage regulator for the motherboard)

Date: 12/2/2018

Reading Voltage into something meaningful is the tricky part.

Digital

Read the powerful
of the battery being generated from the charge controller.

Give restrictions to the user.

Interface: No Analog (restrict the interface to the system.)

Research what interface(s) are most prevalent.
Think about what happens if there is a pressure sensor. How would we calibrate?

I2C, one-wire,

Date: 12/2/2018

We have the BMS in place. (we have 4)

Most Popular Serial Communications interfaces:

I2C (most of everything is this)
One-wire
UART
RS-232
SPI
SDI-12

Types of Sensors I want to ban:

Analog Sensors
(i.e. those that hjust feed an analog value directly to the microcontroller)

Date: 12/2/2018

I2C:

Allows multiple slave chips to communicate with 1 or more master chips.

Allows up to 1008 slave devices.

Most I2C devices communicate at 100kHz or at 400kHz.

Requires these wires:

- SCL: Clock signal
- SDA: Data signa
- GND: Groun
- Vdd: Voltage Supply
- INT: Interrupt – open drain output (active low)
  Allows for no bus contention

Vdd is usually 3V.

(Requires a 4.7k pull-up resistor on the data lines).

Each slave has an address.

3 extra wires really needed:
SCL
SDA
INT ← not always there.

Date: 12/2/2018

More I2C stuff here.

Advantages: more than 1 I2C slave to 1 master pin.

Thoroughly understand I2C

Date: 12/2/2018

One-Wire:

Same as I2C, but only with 1 data line.

Requires 3 wires:

DQ
Vdd (not really here)
GND
Operates between 2.8V to 5.25V.

Vdd is not necessary; one of the features of one-wire is that it can feed power from the data-line.

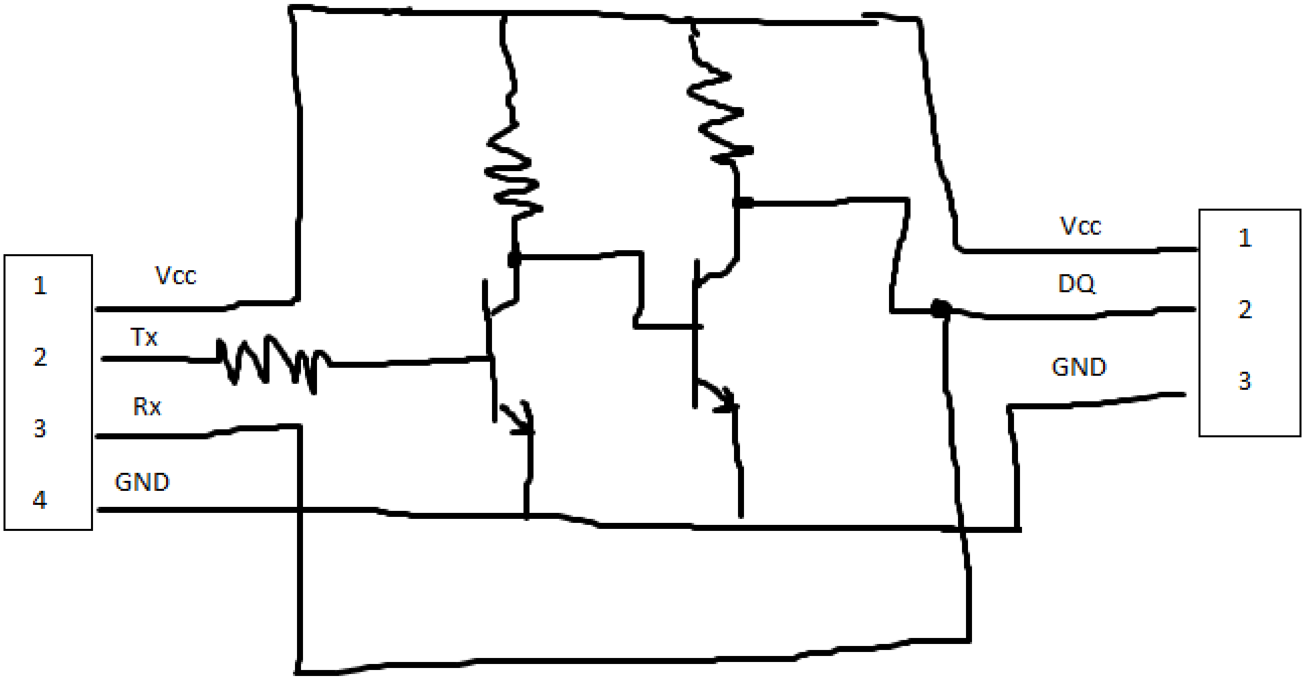
There are only 40 one-wire devices on the market right now. Must we support this?

We have a one-wire device available.

1 wire needed: DQ

Date: 12/2/2018

To use one-wire devices, we could use an adapter to UART.



Then, tell the microcontroller to use the one-wire to UART protocol.
Not sure if it works, but the internet says it works.

Also, I noticed that we only have +12V and GND on these greenhouses. We have no data lines connected. So, we are free to make/buy our own.
Also, (good) water temperature sensors only come with a one-wire interface.

What's the BAUDrate for this adapter? It's 115,200 bits/second(max). Goes to 2-wires.

We can force each one-wire device to use a UART adapter.

Date: 12/2/2018

UART:

Requires 4 wires:

Vcc, Rx, Tx, GND

High bitrate.
Asynchronous. Means that there are start and stop bits.

Must specifiy BAUDRATE. (Frequency of bits incoming)

Start bit, parity bit, stop bit.

Disadvantage: One Master, one slave.

Advantages: widely documented.

Bit intervals are the same, but derived from 2 different clocks.

Date: 12/2/2018

To implement all of them we will need a 6-wire cable:

2 wires for Vdd/Vcc and ground, and
4 wires for data. (if possible: have a mechanism to detect if certain wires are plugged in.)

We could have something like this:

	UART	I2C	SPI
red	Vcc	Vdd	Vdd
black	GND	GND	GND
blue	Rx	SDA	MOSI
green	Tx	SCL	SCLK
yellow			MISO
white			SS/CS

Then we could program the microcontroller to read the input as a selected input. e.g. if we had an SPI sensor, it would use all 6 wires, whereas if we use a UART sensor, it would only use 4.

Date: 12/2/2018

SPI:

Any # of bits can be sent.

4 wires:

MOSI: master out slave in
MISO: master in slave out
SCLK: Line for the clock signal
SS/CS: slave select/chip selected
10Mbps

synchronous

4 wires.

Date: 12/2/2018

Why are the sensor ports actually 12V power supplies? Documentation is lacking.

Ask petersen in next meeting, plus these:

- How to prioritize power data-line
- Since I2C can support 1000 slaves, can we make the power sensors I2C?

Date: 12/7/2018

Meeting Minutes with Petersen

Each interface has its own unique protocol.

Work on a block diagram for the webform.

What variables are necessary for each protocol?

Webform Do-able. Where is that webform?
Ease of use?
Have a candidate webform written up by next week.

Date: 12/7/2018

Correction constant for the polynomial
Requires special attention!
Thermistors require special attention
Providing an analog to digital conversion method.
We ought to limit the types.

Do research on it.

Look at candidate sensors.

Come up with a design.

Same.

RS-232 to UART:

Look into that as well.

Action Items:

- Come up with several webform candidates.

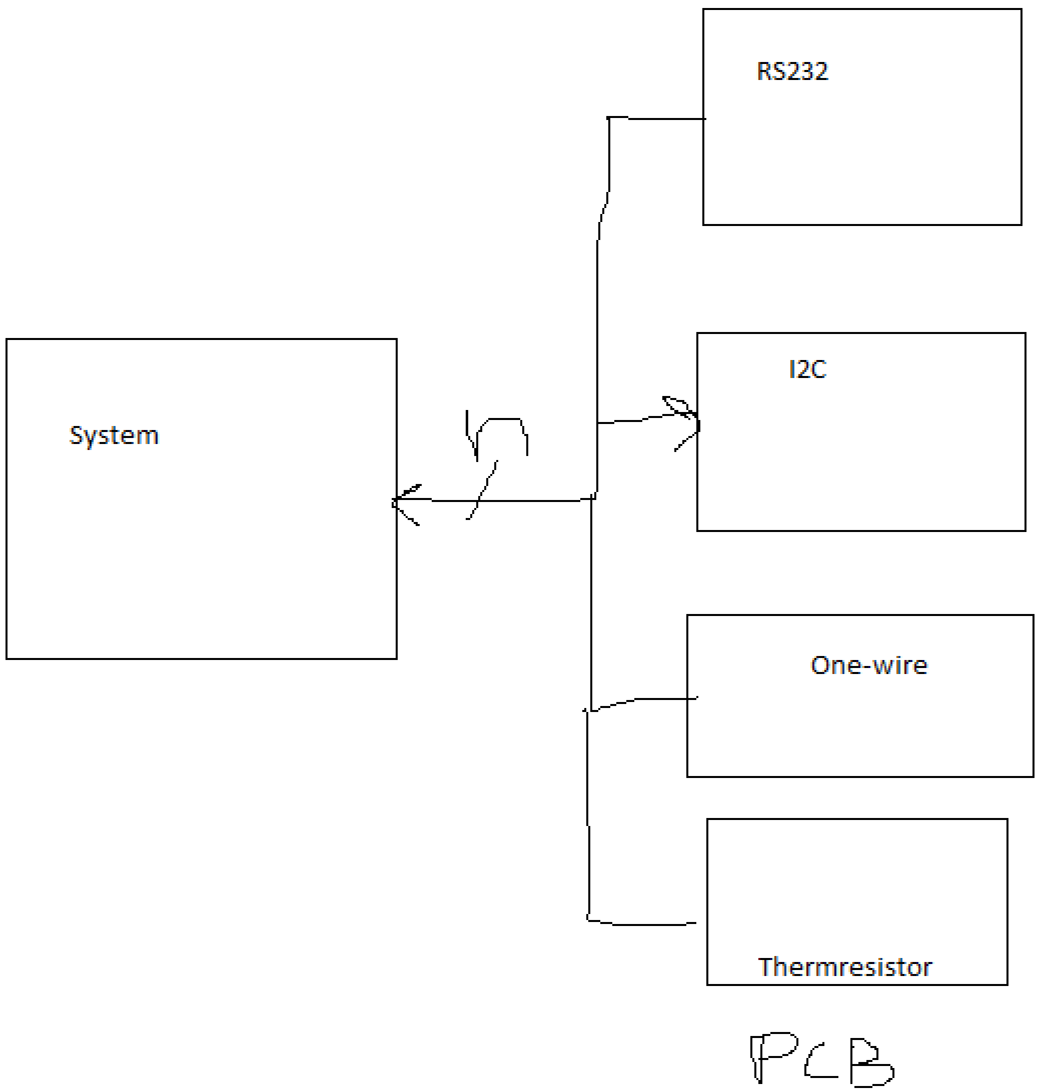
- Start thinking about what’s going to be required for sensors.

- Look for a multiconductor wire.

- Ask Tela how many sensors per given experiment.

Date: 12/7/2018

Significant part:



Make LEDs as status indicators.

User/students.
Make them feel good.

Date: 12/7/2018

Separate problem: how to transform the sensor data into meaningful data

Suggestions:

One scenario: work with Raspberry Pi. Your system provides input to their own raspberry pi.

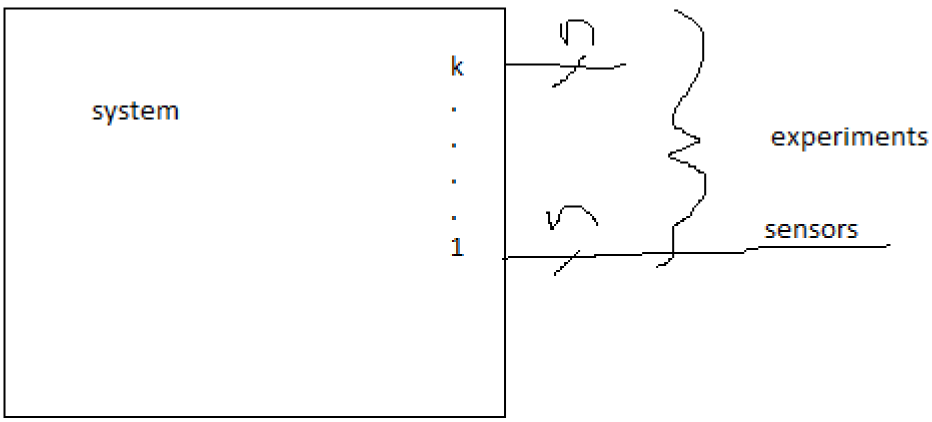
Look into sending code through the web!
You don’t have to process the data, we just have to present it.

If no python coding wanted, do we just spit the valve to …?

Date: 12/7/2018

Questions for Tela:

1.) How many student experiments can be run at the same time?
2.) How many sensors per experiments?



K = # of experiments
n = # of wires per experiment

System health:

Fan status
Start thinking about microcontroller choice

Date: 12/7/2018

We want:
- a candidate hardware design
- written in both schematic form and as a written summary.

Get a complete block diagram by next week.

Date: 12/10/2018

Battery type:

Each one is a lithium-ion battery cell.

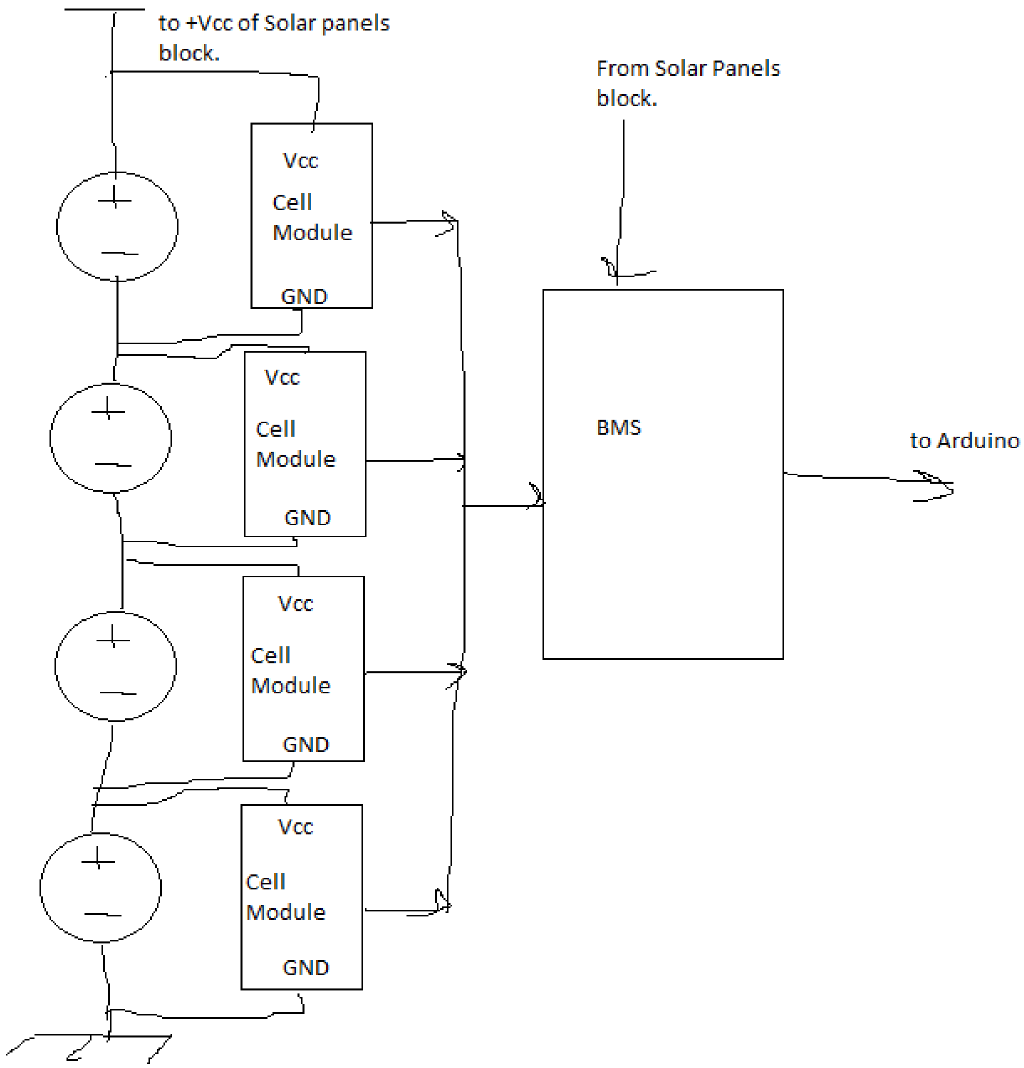
Model Number: IEP71/180/278-CA180FI

Nominal Voltage: 3.2V
Rated Capacity: 180Ah
Rated Energy: 576Wh.

Will be modeled as a battery cell.

Date: 12/10/2018

BMS System Model



Date: 12/10/2018

Cell Module

Model: CM070A-A
(listed on board itself)

BMS Module:

Remus Beattery Management System

Part Number:
CU010D
34 pin header on it.

How does it deliver power on a cloudy day?

Need cable management!

We have a switch on the bottom of the box that does nothing!

Idea: we could have a robotic arm that opens and closes the shutters.

Why is there a random beam hanging there?
Beginning to think I need a hard hat!

Date: 12/14/2018

Throroughly Understand:
- I2C protocol
- UART

Addressing scheme part of the protocol.

Slave 1 & Slave 2
All slaves.

Wakes up.

Something in the serial data that tells me what data is being read.

Date: 12/14/2018

UART:

[Start Bit][Bits 0-7][Stop Bits]

Have a block diagram.

Date: 1/5/2018

Morningstar Sure-sine inverter
300W rated.

Sure-sine-300

12 Vcc battery.

AC: 300W, 115 Vac, 2.5A, 50Hz.

What are the 12V outputs used for?

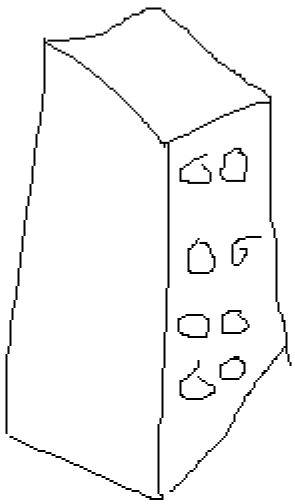
Date: 1/5/2019

First plan to run sensors:

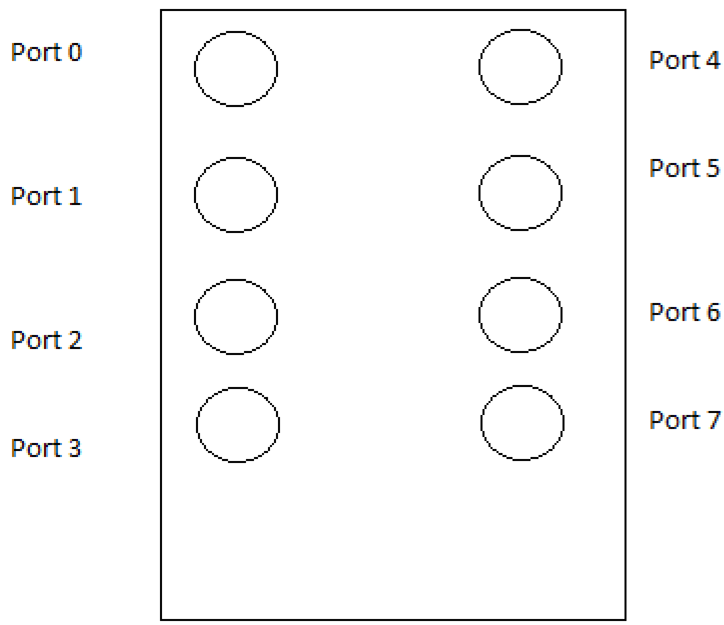
Throw every single input to an I2C bus.

Block Diagram: I’m gonna go with 8 sensors per experiment.

Top Level:



Face:



Each port face has 6 pins:

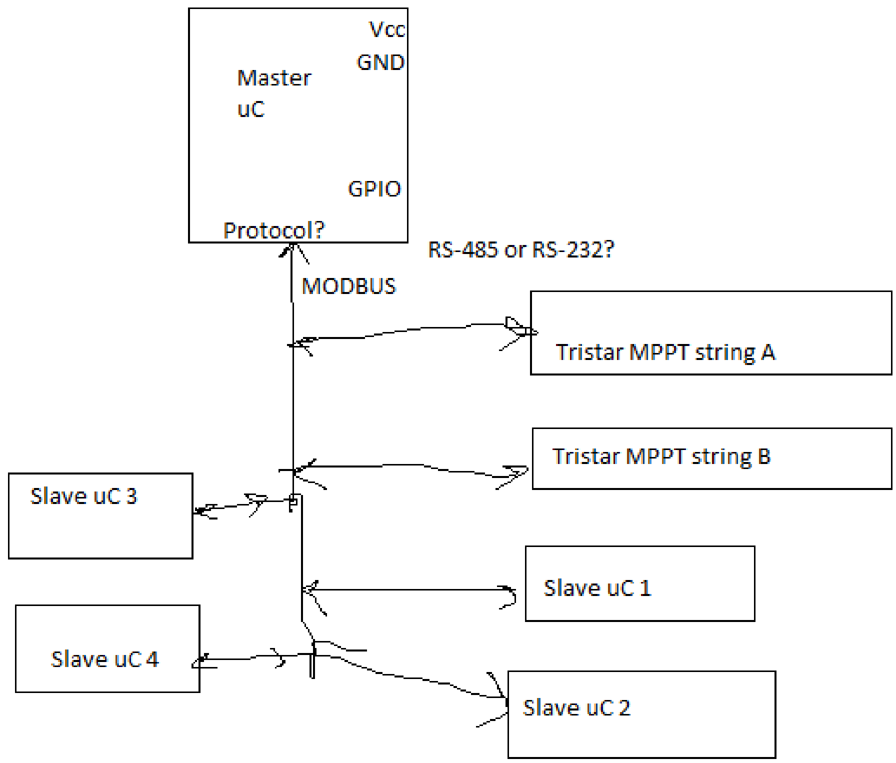
A	B
C	D
E	F

Ports 0 & 4 will be locked into 1-wire so we can have a bridge.

Date: 1/6/2019

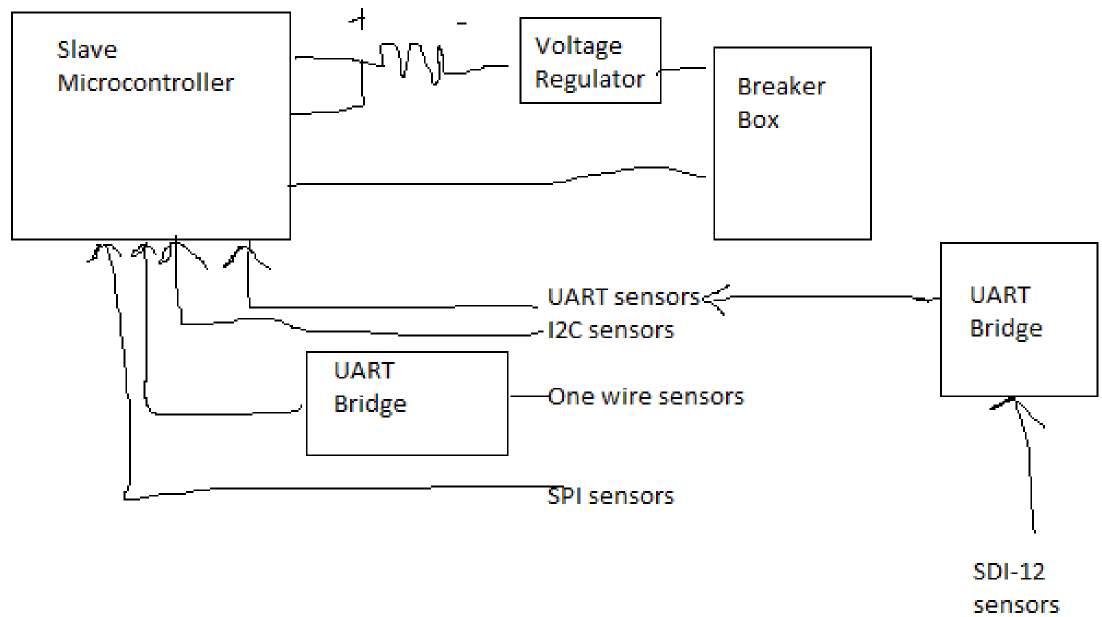
All solar panels are linked in series.

Microcontroller setup:



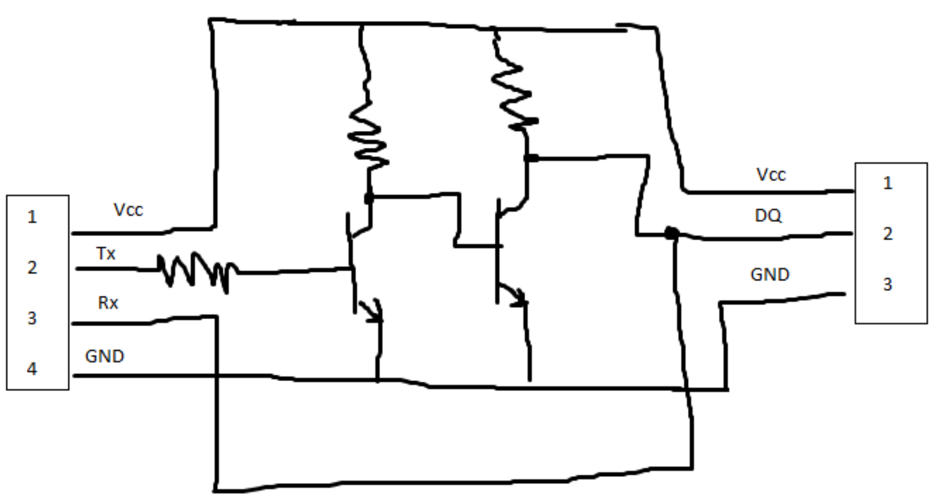
Date: 1/6/2019

Each slave microcontroller can gather data on its own.
They will be mounted next to the 12V breaker boxes already mounted. They could use a voltage regulator (unless a buck converter is necessary), and could be programmed by the master.
Each slave can collect its own power data.



The slaves emulate a MODBUS slave.

1-wire to UART bridge:



Date: 1/6/2019

To multiplex UART, use a shift register.

8-way UART multiplexer requires 3 bits.
16-way UART multiplexer requires 4 bits.

We could have UART, I2C, SPI and 1-wire to USB briges, and then plug it in to the USB hub.

Designing our own temperature sensor?

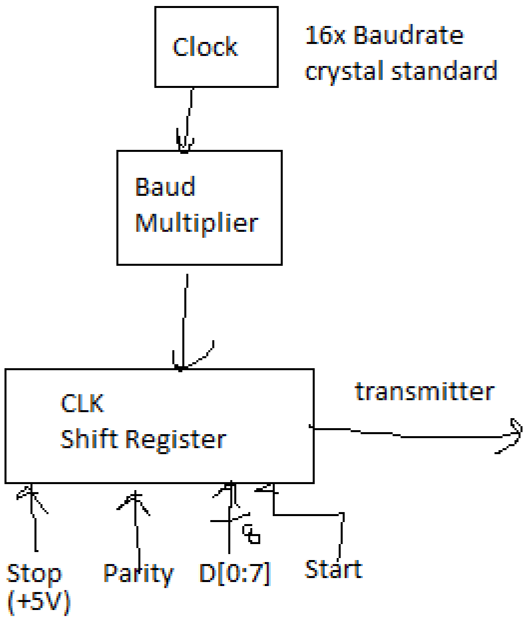
Date: 1/8/2019

The MODBUS protocol should use an RS-485 electrical interface for long transmission lines. I think we already have the code for MODBUS interpretation.

Date: 1/8/2019

UART:

Generation:

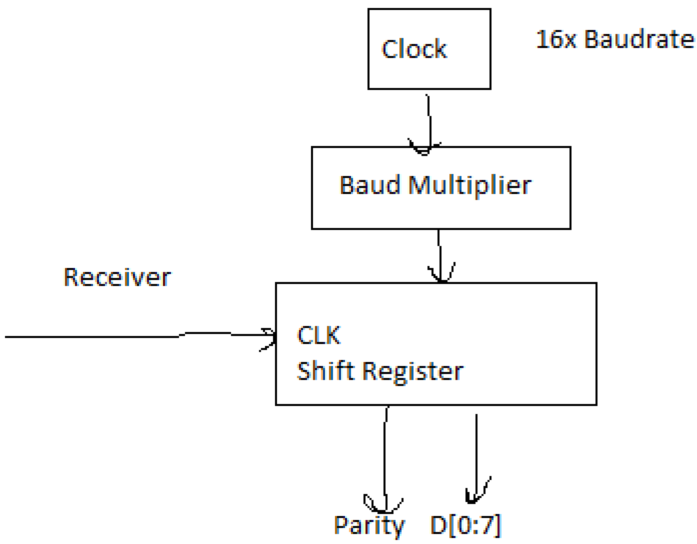


See app notes for UART

Date: 1/8/2019

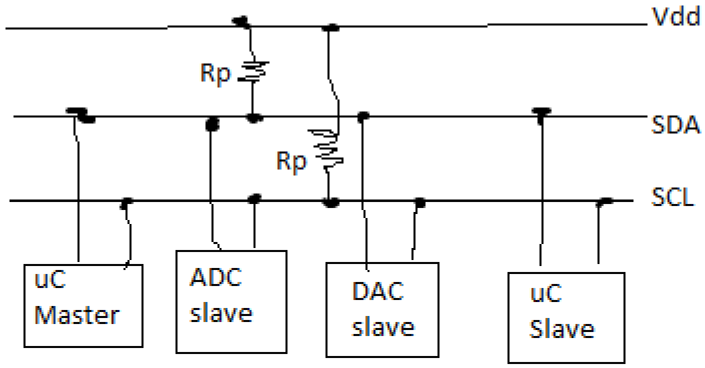
~~I2C~~ UART

Write down my understanding of UART.



See APP NOTES for I2C

I2C:



Each I2C slave devices has a 7-bit address. Needs to be unique.

Master generates the clock.

Communication is initiated by the master.

Generates a start condition followed by address of slave device.

START condition:

SCL must be high all the time.
SDA goes down.

STOP condition:

SCL must be high all the time.
SDA goes up.

Bus capacitance must not exceed 400 pF.
So, short cables are necessary.

Each byte must be acknowledged.

SDA = LOW
1 clock pulse.

Date: 1/10/2019

Meeting

Fully understand RS-232 and RS-485.

Capacitance per foot, etc.

use engineering.

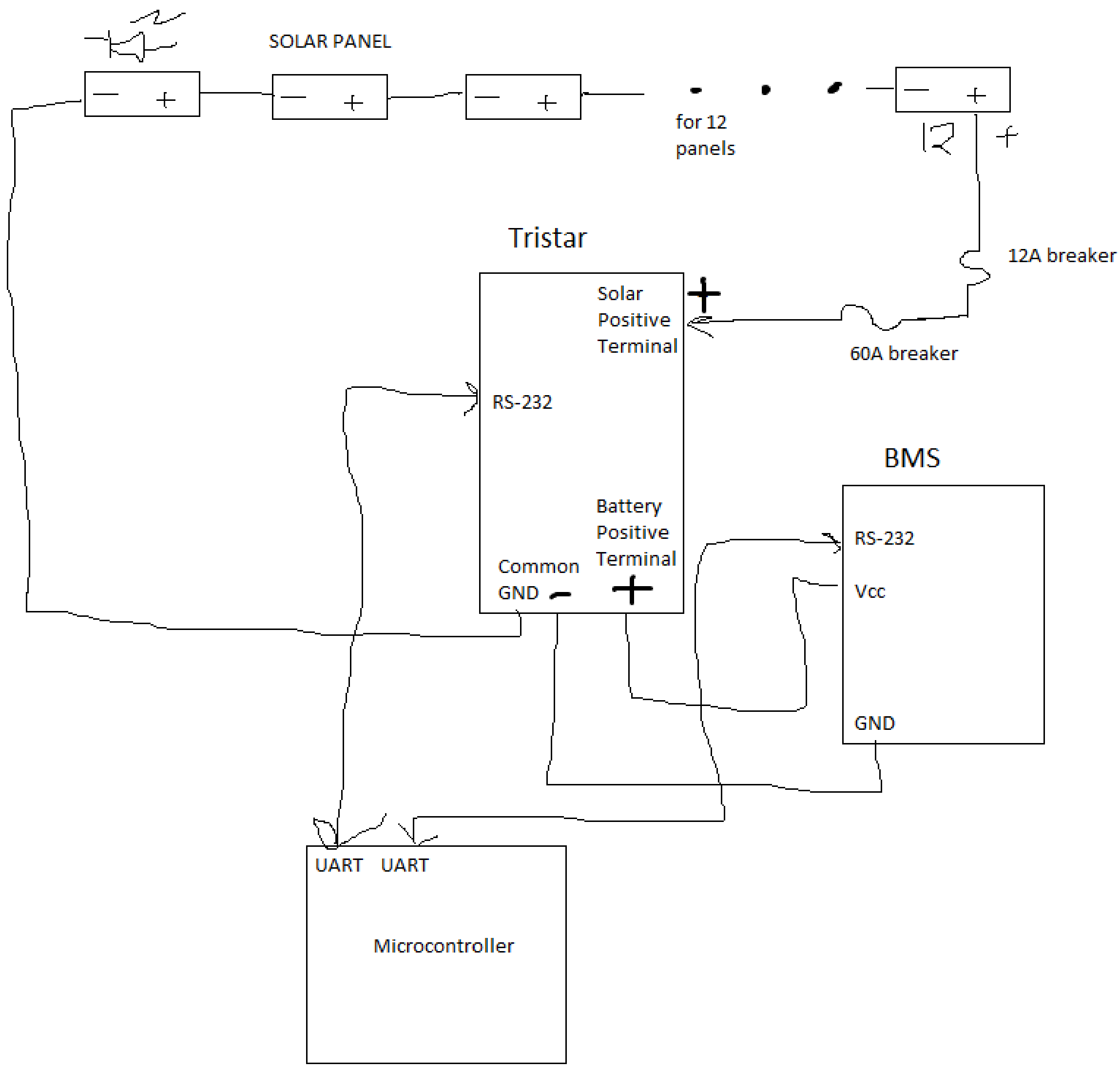
Let’s have a typewritten document.
It’s easier to do.

Get engineering notes into professional order
- Name, date.

Meet at 5:30pm on Friday Nights.

Date: 1/15/2019

Tristar MPPT Block Diagram:



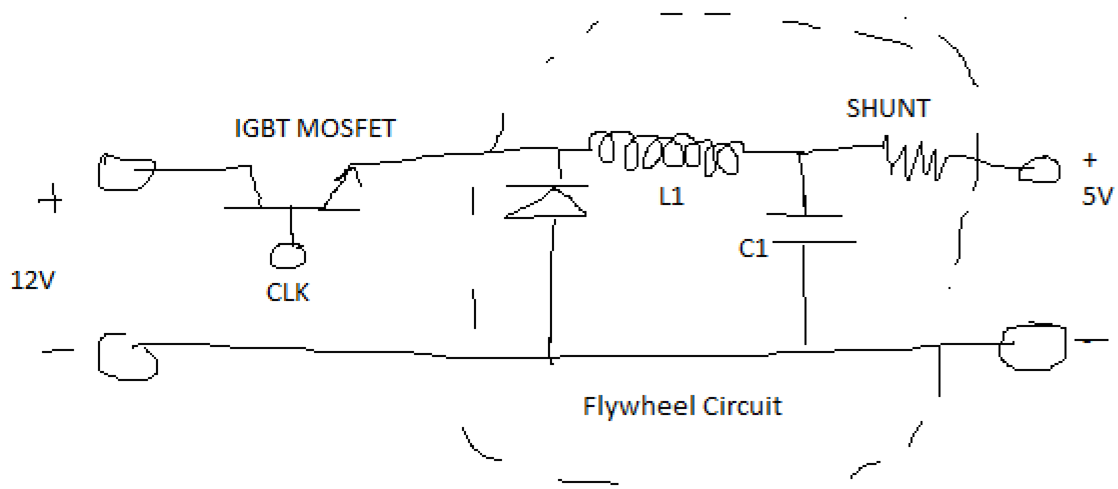
(Arduino Existing)

Date: 1/17/2019

Power Transformation

I don’t want to use a 5V linear regulator for this.

I want a full-fledged buck converter. Not just any converter, either. I want one that logs power data, too. Here’s my schematic for the transformation:



Vout = Vin*ton/T

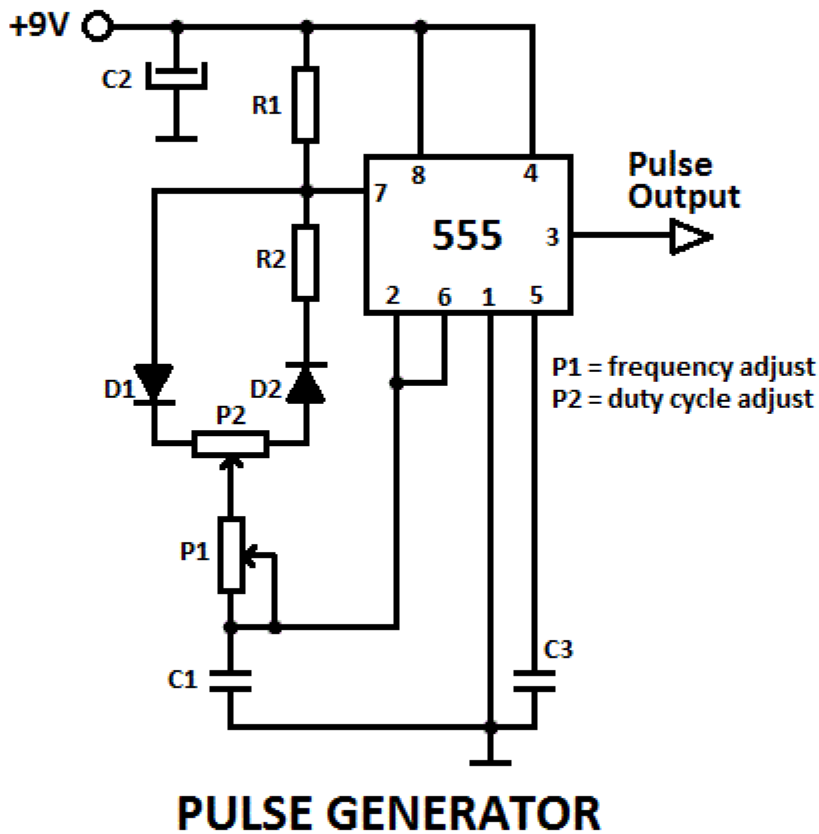
Duty cycle = 0.141666666 = 14.167%

To control the clock, we would use a 555 timer. Let’s use a pulse generator circuit (a PWM circuit)

We need to have one because a voltage needs to be changed.

Date: 1/17/2019

PWM schematics from electroschematics.com:



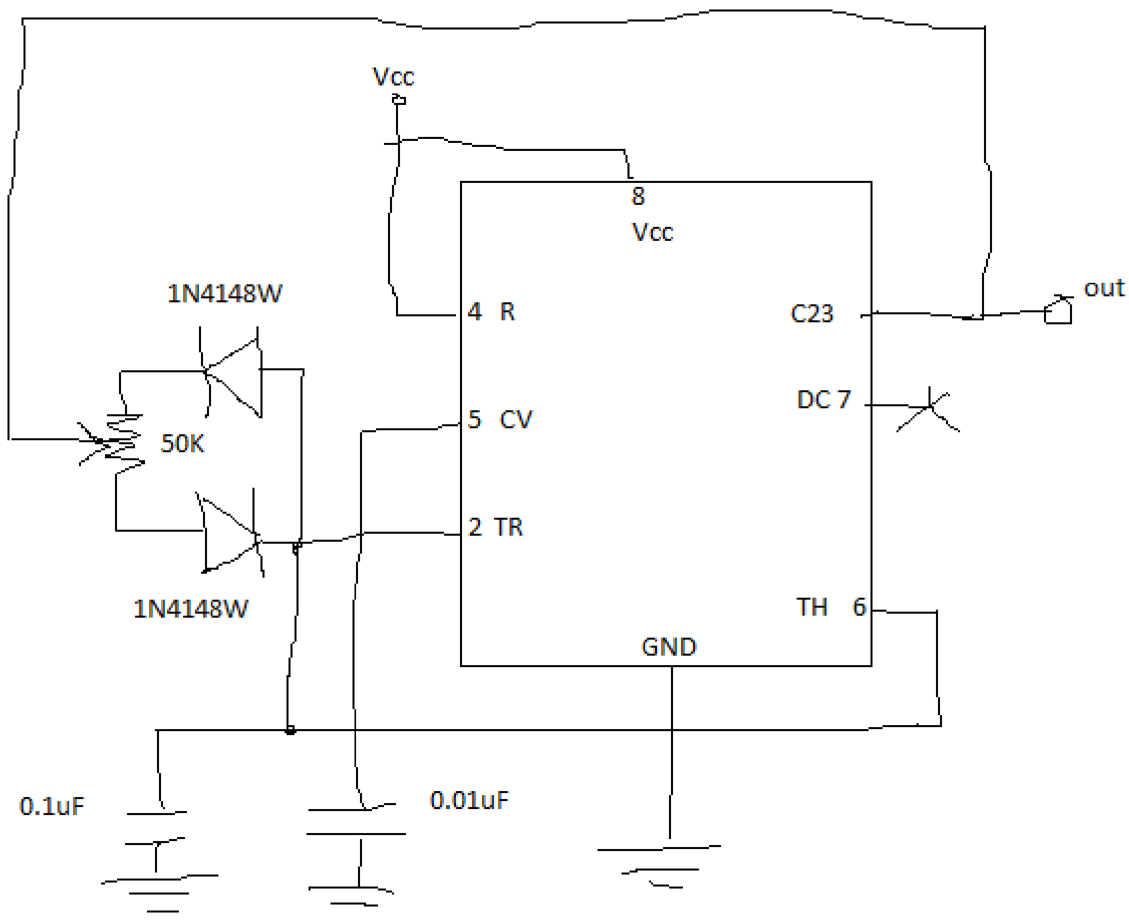
frequency,

f = (0.69) / ((2*R1 + R2 + 47k)*C1)

duty cycle, D = (1+R2/ R1)

Source: https://www.electroschematics.com/5834/pulse-generator-with-555/

PWM output from allaboutcircuits.com:



Source: ???

Date: 1/17/2019

Flywheel Circuit: let’s talk about it.

Date: 1/17/2019

Shunt Resistors:

0.01 ohms usually
Need to withstand about 0.5W of power
(since 2A usually flows).
Use a chip.

Is Power a Big question/factor in microcontroller choice?

Justify the choice.
The only choice
would tela like this to be efficient?

Powering the microcontrollers and sensors

Come up with a justification with numbers.

Create a power budget for this.

A spreadsheet

Raspberry Pi

Use power budget to justify buck regulator.

Date: 1/17/2019

uC Choice:

I choose the Raspberry Pi, since I’m getting more and more used to it. We can program it using Wi-Ffi, and I can send data through it.

The slaves will be Raspberry Pi Zeroes ($5/each)

Just as programmable as the mother ship.

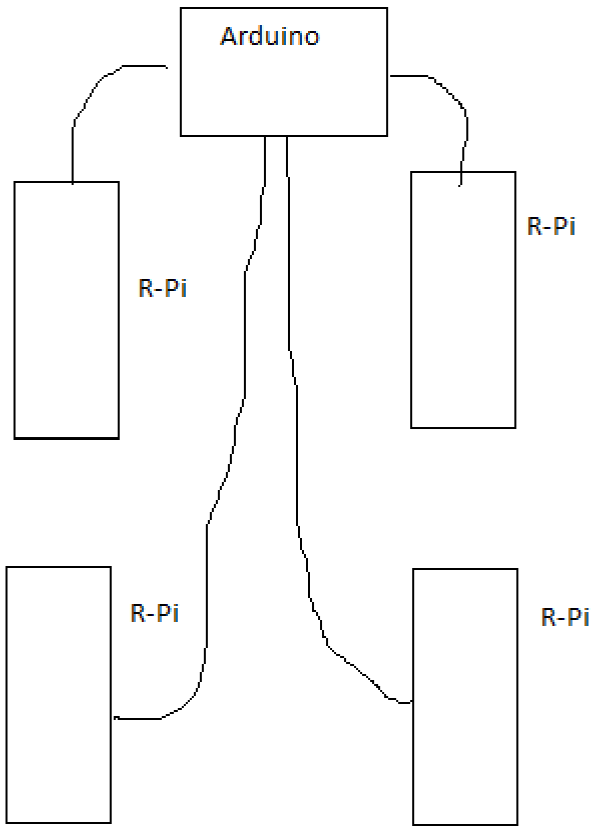
Action items:

- Create a power budget
- Justify a Buck Regulator using a power budget.
- Get concept of a power regulator down in a block diagram.
- Finish binder organization
- Proceed with a more detailed engineering schematic
- Show sensors in the schematic

Date: 1/17/2019

Have 4 slave uC's.
How am I gonna design the slave uC?

Full hardware designing



Look at the power of the arduino

Commit this to paper: what protocol and why? (for intra-uC communication)

Date: 1/24/2019

Ask Tela:

Maximum number of experiments that can be run at the same time is.

How many sensors per experiment?

Preface: this was one of the action items from the meeting with Petersen.

Justify using a raspberry pi.

Have a list of things the arduino needs to do.

Create a block diagram of the Arduino’s necessary capabilities.

Decide if the Arduino is suitable.

Date: 1/24/2019

Meeting Minutes:

Canvas:

Design Notebook from Tela:

Use the criteria matrix

Have a tradeoff study that shows my thinking

Power budget is separate from Design Notebook

Have Tela review it.

Memory budget: compare processors:

I/O pins

Wi-Fi

Bluetooth

Bluetooth:

Put those in my tradeoffs.

Power Budget: Needs to be presentable. Need additional columns.

Once I have selected a microprocessor, we can make a better block diagram.

Justify the use of the power regulator.

Pull the design notebook together.

Have a specification document.

Trade study.

Clean up block diagram.

Have specs. Include sensors she outlined.

Network.

Each one needs to be able to communicate with one another.

Date: 1/24/2019

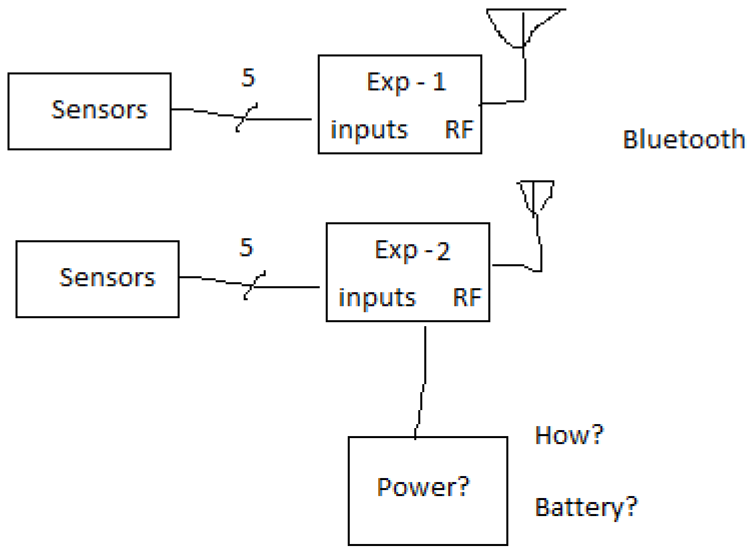
Ask Tela:

If a pressure sensor is needed:

What type of pressure is needed?
Atmospheric? Water?

Date: 1/24/2019

Current:



Date: 2/1/2018

Meeting Minutes:

Action Items:

- Look into buck regulators that come with a PWM signal generator
- Read Petersen’s app notes
- Pull together a specification document
- Put it in a design notebook.

Aim for a specification document:
- She told me: 5 sensors/experiment
    with 2 experiments
- Add microcontroller to:
    Add to trade-off study
    Pull things together
    Then my block diagrams make sense
    Certify that based on the number of sensors
    Beaglebone
    Have a specification document that builds progressively.
    Have some specs as a readable block diagram.

Date: 2/1/2019

PWM chips that are appropriate for this application:

Lecture notes on power supply stuff.

Example app notes:

Do we have:

Fill out design notebook. Have a spreadsheet with pros and cons. Have Tela look at it.

Use power budget for new things (like fans and heaters).

Engine
Then

It’s OK to buy one (a buck regulator), just have a justification for it.

Date: 2/1/2019

We’re allowed to use RS-232 to USB converters!!!

Date: 2/1/2019

Greenhouse.

Students can plug stuff into the greenhouse. It’s a power distribution scheme.

RS-232

Date: 2/8/2019

Meeting Minutes

Access to 310 lab?
Come to BELS on monday at 11:00 am or so.

Carry over action items:

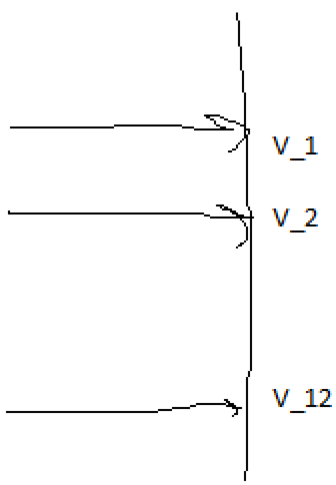
Look into Buck Converters and Petersen's notes.
Get the specifications into a Design notebook.
Design notebook → Get her Buy in! Her say is final!

Put block diagrams!
Learn how to characterize a solar panel over a day and over a year! A day: sunrise → sunset. Year:
elevation of the sun in the sky.

Daily average: depends on day of the year.

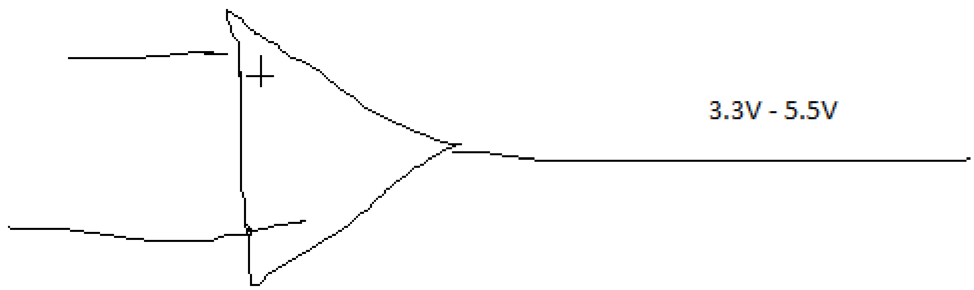
Put Page Numbers!!!

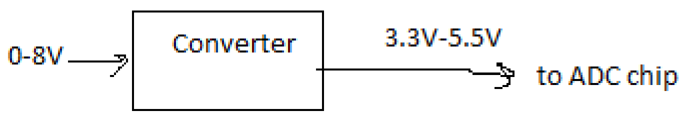
High level block diagram!



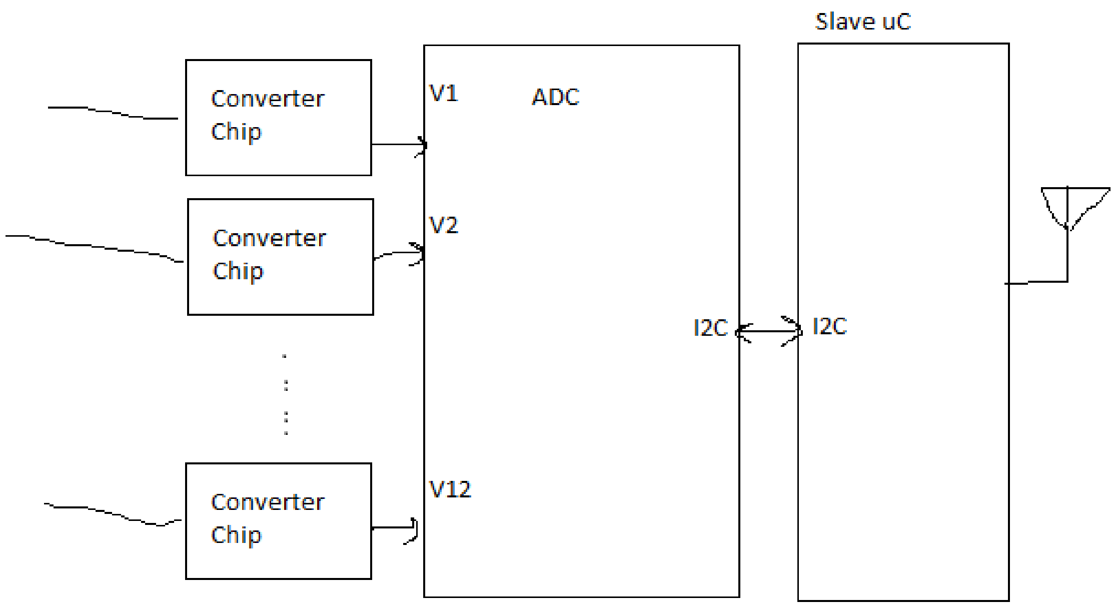
Use an op-amp to scale it!

Date: 2/8/2019





Draw on these:



Power: comes from the 5V buck converter.

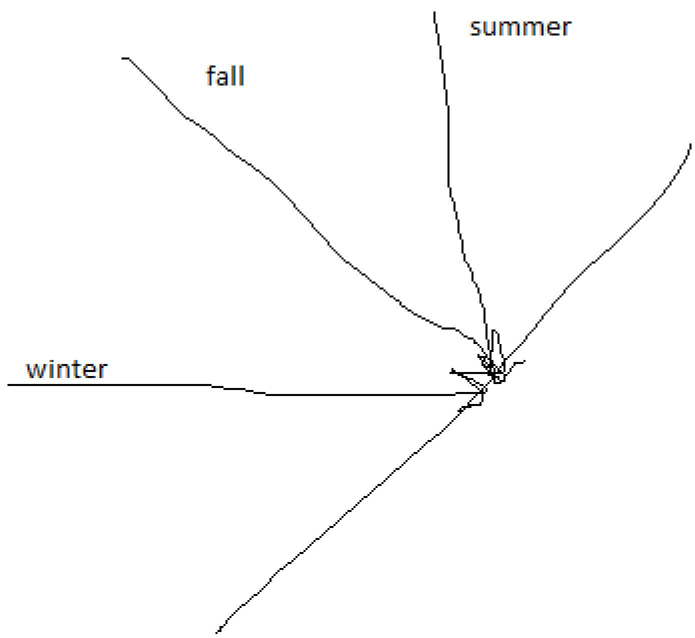
Have block diagram.

Date: 2/8/2019

Max
if

talk to tela about Power Generation.
If they move, what is the spacial orientation of the solar panels.

Example:



Estimate solar [output in writing]

Date: 2/8/2019

Angle ot the sun,
Power per square meter.
At the equator, it’s 1kW/m^2. Here, it’s 750W/m^2.

Solar exposure from sunrise to sunset.

Some sort of sine wave. We need to ….

Note: what kind of light is absorbed?

Hardware read.

System specification!

Another group: needs to characterize a solar panel using a data logger.

Date: 2/15/2019

Meeting Minutes

Next time we go down to the greenhouse:

Confirm dimensions of greenhouses
- Solar Panel Dimensions
- Greenhouse Dimensions
- Thing
- What type of Fan
- Check Filtration for Fans
- Do we have connected wires?
- Do we have a heater/cooler?
- Mention a leak
- Plug in lights?
    Put in 15W fluorescent lights.
- Take these, have an index in our binder.

Date: 2/15/2019

Thematic notes:
- How to measure sine inverter voltage and power drawn.
- Somehow have a mechanism to open the shutters if it’s too hot.
- Specifications down before moving forward!

Date: 2/15/2019

What else do we need:

- Solar panels (size)
- Confirm Battery Capacity with a datasheet.
- Block Diagram
- Engineering schematic of the system
- uC is squared away
- Specify:
    - Performance of the system. What’s that?
    - Specification: 2 experiments, 5 sensors each.
    - Battery charging and delivery.
        - How much power are we generating and consuming.
    - Battery:
        - Maximum capacity
        - Current capacity
        - How many batteries
        - Are we losing power from batteries or charging them?
    - Characterize:
        - MPPT: 100W

Date: 2/15/2019

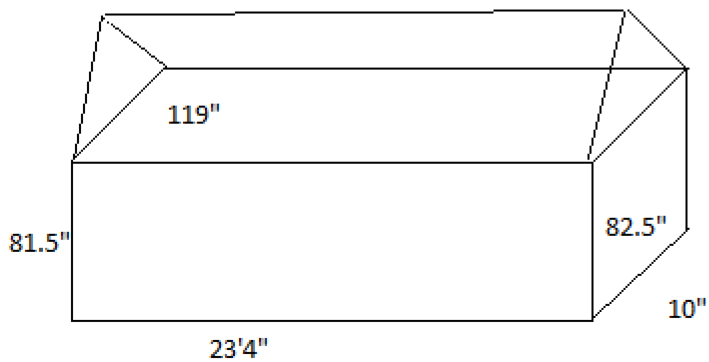
Action Items:

- Confirm wiring of solar panels
- Take dimensions of solar panels
- Identify Mystery Box
- Figure out how to measure temperature (ask Tela first)
- Confirm ventilation with Tela
- Identify specifications for performance.
- Determine Batery capacity for enclosure.
- How is the power distributed from the batteries? I still don’t know.
- Solar panels: wiring.
  Then: determine from a single string. Wire 2 strings: 88V.
- Output from solar panels, verify that it’s appropriate for the Tristar MPPT.

Specifications must come together!!

Date: 2/20/2019

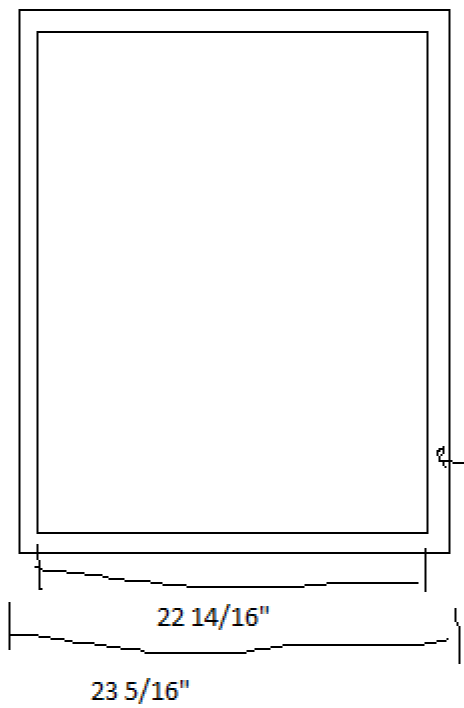
Greenhouse measurements:

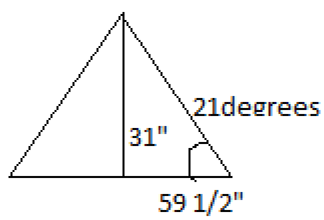


measure: height, length, width, diagonals
Basically all the lines.

Solar Panel Measurements:

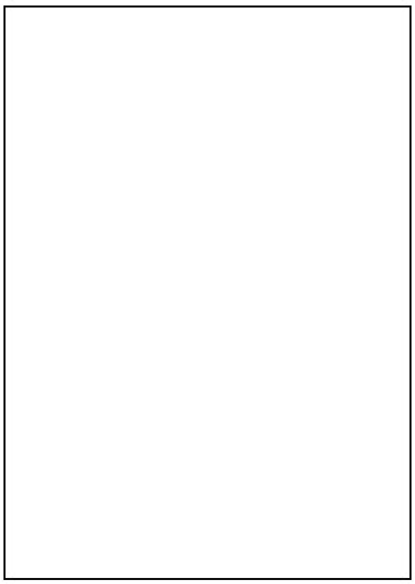
Solar Panel Model: LUMO0/4





Date: 2/20/2019

Door Frame:



Date: 2/22/2019

Reading for specs.

I read the previous team’s documentation.

The fan is of the brand “SNAP-FAN”.

Apparently, they don’t work. It seizes at times.

They operate at 20W.

Stall torque operation when it seizes.

They were supposed to order a new one, of the same brand, power cost of 34W.

We need one that operates at 12V. Tela asked them if she should order the fans.

Apparently there were supposed to be 60-75W water pumps attached.

And 35W air pumps.

Apparently the user’s power budget was 1.80 kWhr/day.

Date: 2/22/2019

Plan:

1.) Get specs approved.
2.) Figure out how to send data thru FONA
3.) Figure out how to read RS-232 data from USB port
4.) Figure out how to communicate between 2 Bluetooth uC's and send data.
5.) Figure out where to put the uC slaves and their power sources. (It's gonna be a wall bug.)

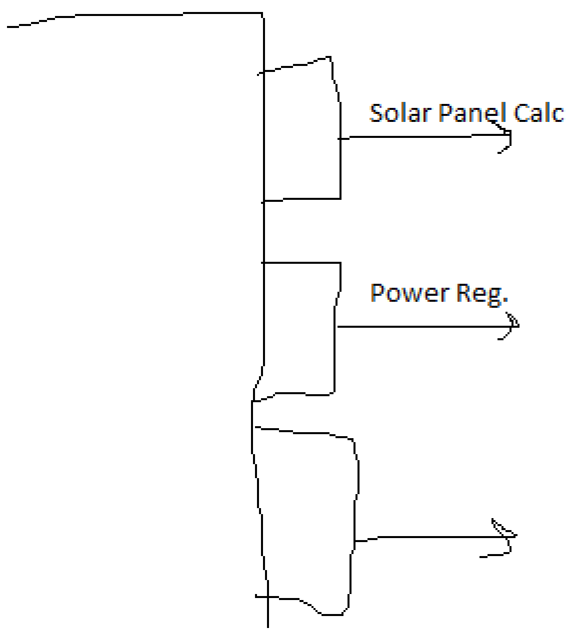
Make a section in binder parallel to engineering notebook.

All earlier versions.

Solar panel calculations

scheme.

Solar panel calculations



Date: 2/22/2019

Summarize these things in a report, hand that to Petersen.

To make revisions, specify which page you’re revising, and revise only that page, and print only that page.

Each battery: 576Whr.

Action items:

Measure the physical volume of the batteries.

Strage circuit board: figure out what it does.

Organize the tuff in the binder.
→ THEN START SPECS!

Draw an engineering schematic of the BMS system (that makes sens)

How does the BMS route power?

Categorize datasheets. Itemize them

Digest the BMS manual.

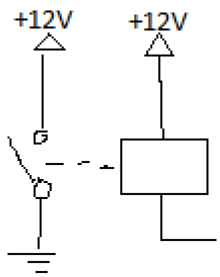
Characterize BMS system!

Date: 3/1/2019

Since the BMS monitors temperature, and warns you if the batteries are too hot, it comes with a heating and cooling fan control. We will need to install a heater/cooler inside the battery box to protect the batteries!

BMS system has an event system. It can store the last 32 events in its registers.

How to connect BMS fan:

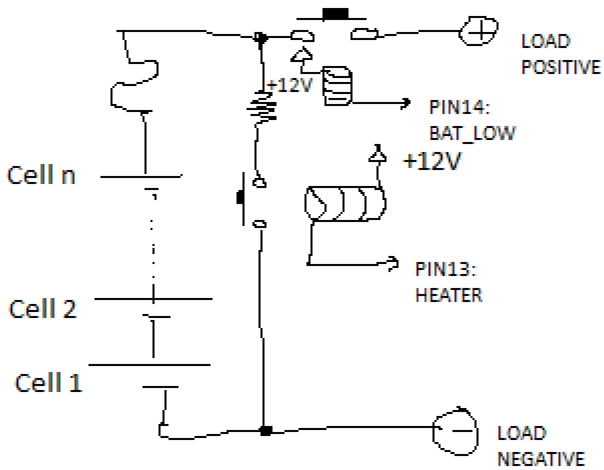


PF13 Battery Connects to pin 16.

Fan output.
Responds to Battery Temperature being too high.

Date: 3/1/2019

BMS heater:



How to connect multiple of these cells:

Use CAN group modules. We can have >1 battery box this way.

1 battery box can be 1 group module.
Battery.
Every group must have its own charger.

This is a group of batteries. It's a series-wired group of batteries.

Usually, a CAN-based charger is used. I don't know which one we have. I think we have a CAN charger, considering that we've got a bunch of CAN-themed code lying around.

Date: 3/1/2019

Meeting Minutes

Action Item:

Figure out how RS-232 protocol talks to CAN bus.

How are the bytes loaded in the BMS.

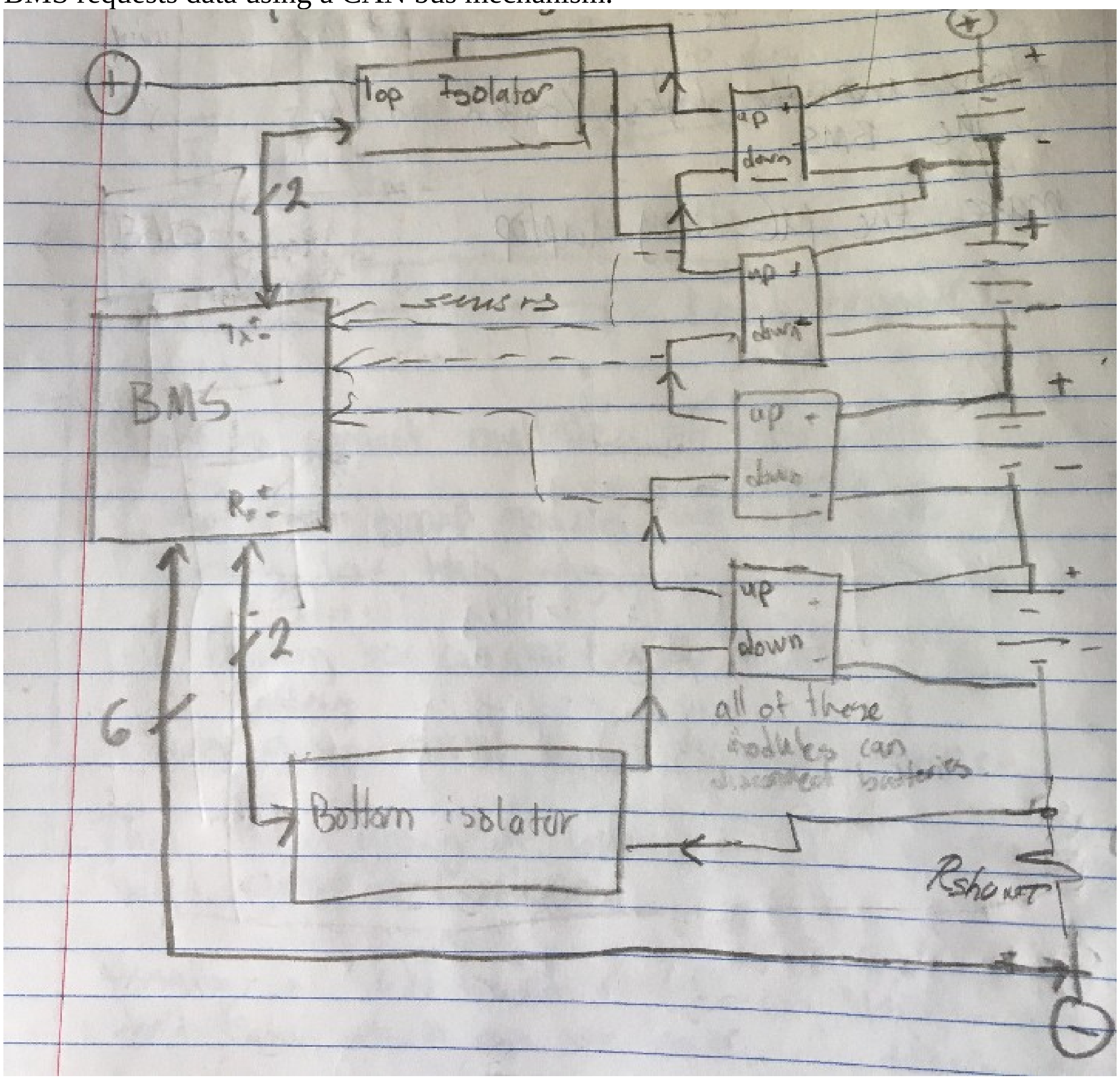
Make the uC my laptop!.

Date: 3/8/2019

My understanding:

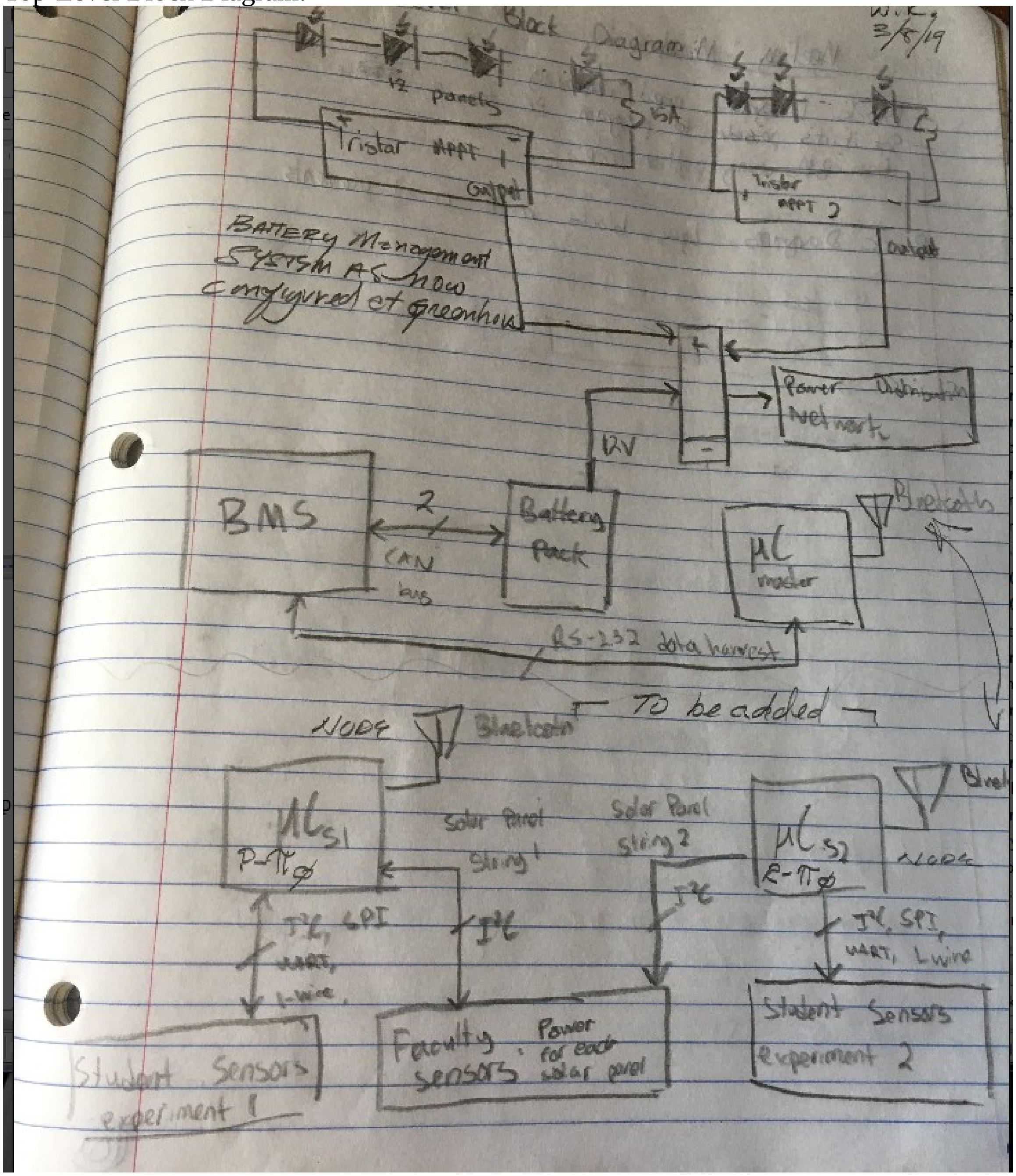
BMS can send data using sentences. Request Data using sentences using the RS-232 electrical protocol.

BMS requests data using a CAN bus mechanism.



Date: 3/8/2019

Top-Level Block Diagram:



Date: 3/8/2019

Meeting Minutes:

Block diagrams need improving
See if it’s possible to program PC to read from BMS using python code.

Block diagrams: Less blocks, more circuit elements.

Date: 4/19/2019

Baud rate calculation:

Waveform looks like this.

High point:

	800ns	800ns
Low point:		
	7.2us	7200ns
	9us	9000ns
Rx	period: 8200ns – 8.2us	

frequency: 1/8.2 MHz = 0.12195 MHz
= 121kHz
= 121950 Hz
= 121950 Baud

Date: 4/19/2019

Logical low

Tx:

12us high
30.4 us low

Tx on high logical:

- 45.826 ms between data points.
- 45.744 ms

0.

64.69ms to 64.78ms
0.09ms

90us = 11111 baud

Jack shit is coming from this.

It seems as if the baudrate just changes randomly.

This note has experimental data I worked with measuring/baudrate etc.

Date: 5/9/2019

BMS Pinout

22-pin connector
Dual Header

Top View
Make a header for it!
Don’t rely on EMUS!

1 GND	2 +12V
3 Cell Rx-	4 Cell Rx+
5 Cell Tx-	6 Cell Tx+
7 GND	8 USB PWR
9 USB D-	10 USB D+
11 DISP Rx	12 DISP Tx
13 BAT_LOW	14 HEATER
15 CHG_IND	16 BUZZER
17 FAST_CHG	18 CHARGER
19 AC.SENSE	20 IGN. IN
21 CAN-	22 CAN+
23 SOC OUT	24 SPEED IN
25 GND	26 +5V out
27 INPUT3	28 INPUT2
29 INPUT1	30 INPUT2

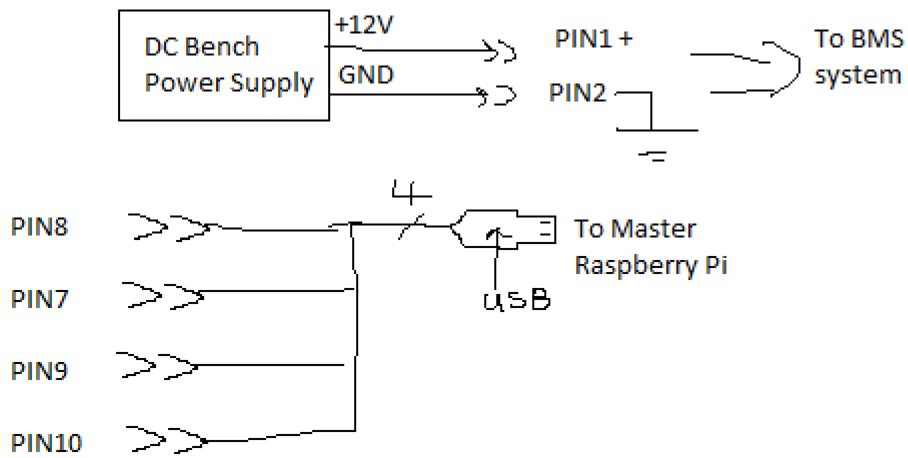
Date: 5/9/2019

Commit it to a cadence schematic!

Date: 5/17/2019

BMS Diagram:

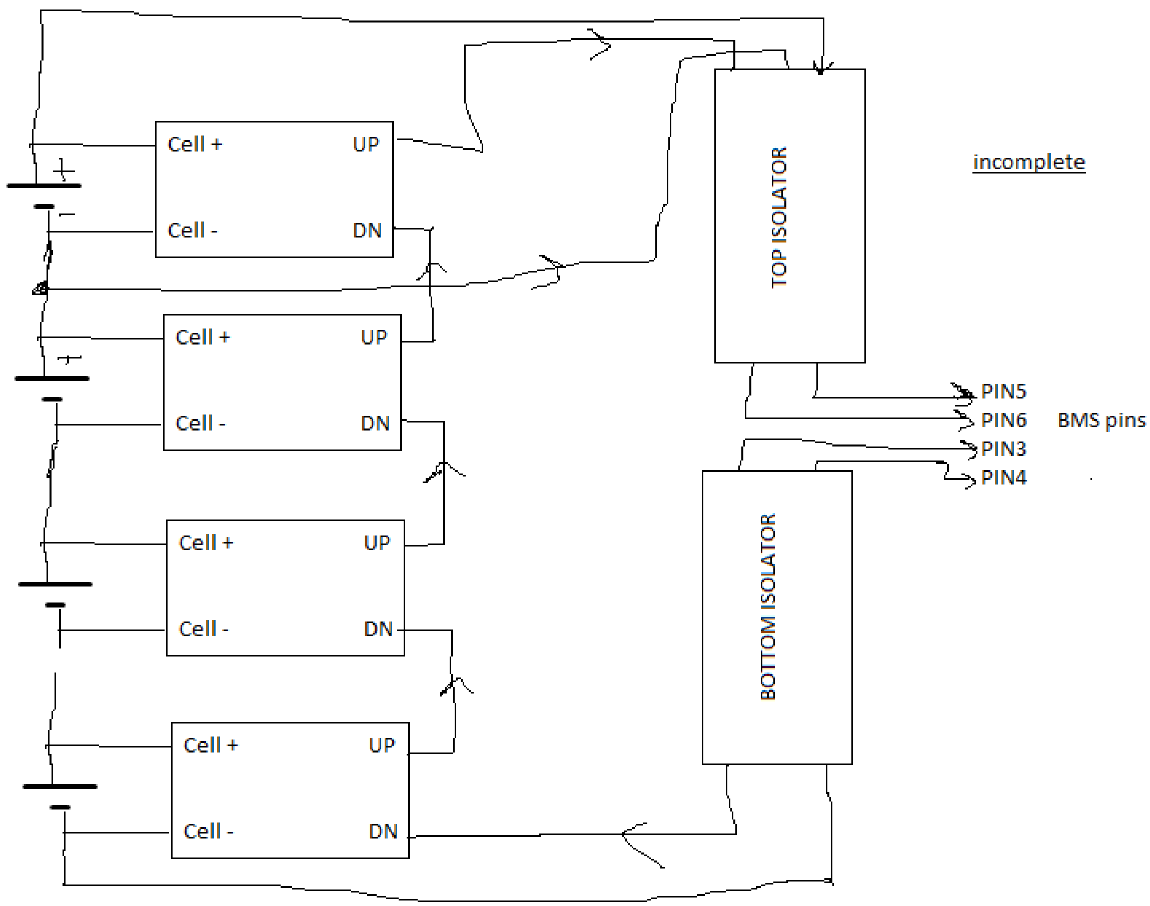
Wiring diagram For Testing at BE310:



Date: 5/17/2019

Wiring Diagram for Greenhouse:

Isolator setup:

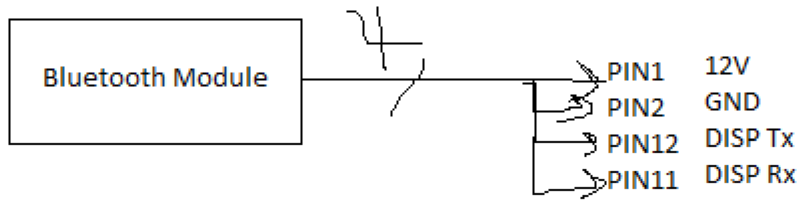


This is how you setup the isolator boards to work with the BMS.

Date: 5/17/2019

Wiring Diagram for Greenhouse:

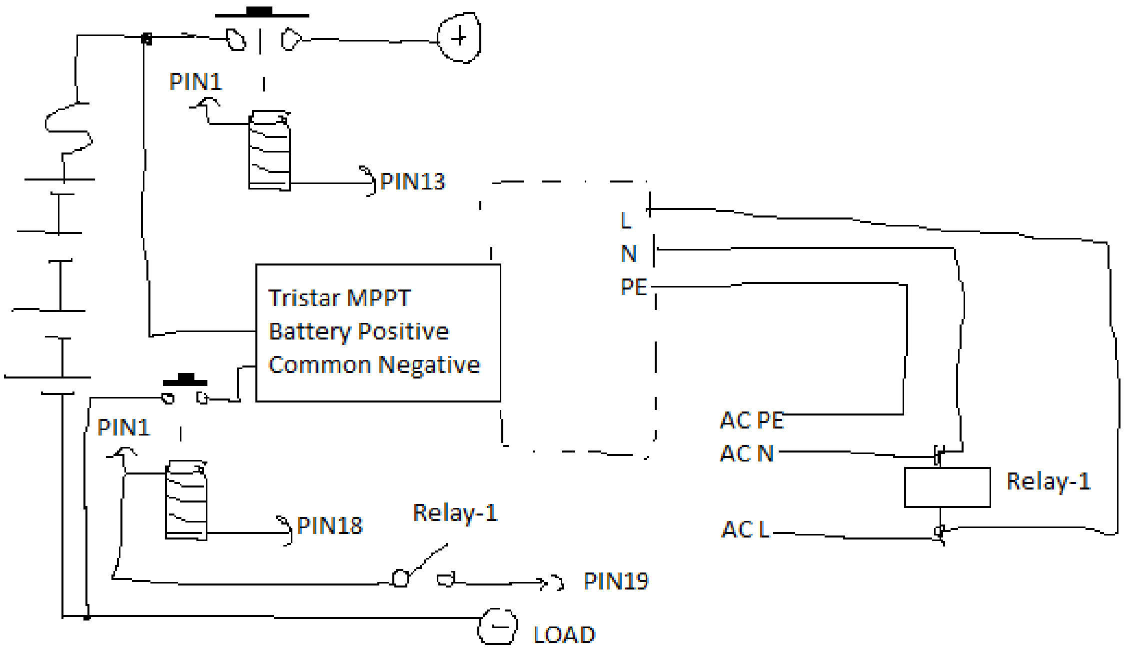
Bluetooth Setup for BMS:



Date: 5/17/2019

Wiring Diagram for Greenhouse:

To hookup Tristar MPPT to BMS system:



Complete this!

Date: 5/24/2019

Agenda:

- Discuss the scope and deliverables of the project
- Do we need a website? Review website.
- Do we need a sensor uC? Sensors? Review sensor lists. Tela email and uC. 1/25/2019
  → Heaters, coolers, lights, AC outlet, water pump, air pump.
- Review Battery monitoring scheme.
- Review Tristar monitoring scheme.
- Does the block diagram have everything?
  Review completeness of Block Diagram Rev 1.2 dated 5/24/2019

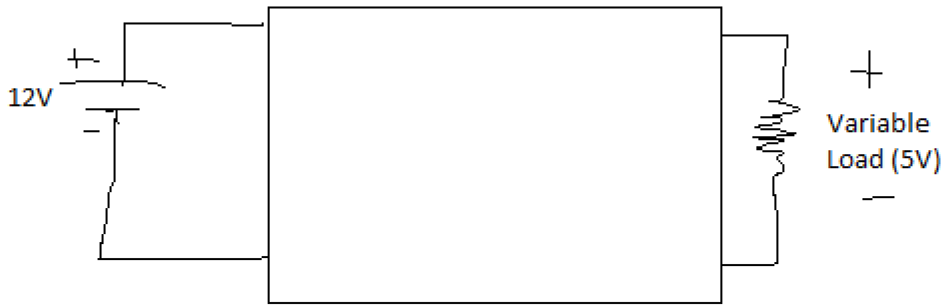
Review:

Top-level Block Diagram Rev 1.2
BMS Wiring Diagram (BE310) Rev 1.1
BMS Wiring Diagram (Greenhouse) Rev 1.01

I would like to meet, and here is the proposed agenda.

Date: 6/7/2019

5V power supply



Date: 6/7/2019

Meeting Minutes

Add another temperature sensor.

Independent temperature sensors independent of the student sensors.

- Light
- Temperature
- Humidity

(the state of the greenhouse)

Any protocol. Mounted on the electrical specifications on

humidity – indicator
light – accuracy, angle, exposure to the sun.
temperature: out of the mppt and slightly outside the temperature box.

2 light sensors.

Determine appropriate range and temps.

Date: 6/7/2019

What are the specs?

Get the greenhouse wired up.

August 31st, 2019: the greenhouse moves!

Must be done before it moves!

Get BMS modeled (electrical)

2 documents for each thing:
the battery and the electrical powerful

Think about what temperature the greenhouse will get to.

Date: 6/19/2019

STEPS:

Step 1: Finish Wiring Diagram

Wiring Diagram – something you can use to rebuild the greenhouse from scratch.
→ the circuitry inside, not the greenhouse itself.
Documents must draw on sources you need.

The BMS control unit, the Tristar control unit, etc.

Step 2: Work on new design to monitor the state of the greenhouse.

Add on individual Raspberry Pi that has external sensors.

External sensors must have:
- acceptable range
- acceptable sensitivity

How will I proceed with my new design?

Design a system to monitor the state of the greenhouse. Review the design with Tela. Propose a system to Tela. After you have a complete system.

Date: 6/19/2019

Retails are circuits and component selection
Keep:

Thesis will be written to DC standards.

Ask Tela what she wants in the thesis?

What should be in the thesis?

I have a question: can we use car batteries in parallel? If not, why not?

They can store (at least my car can) 12V at 900A cold-cranking for 30 seconds.

That’s 324,000 Ws of power

= 5400 Whr
= 5.4 kWhr

B/C it’s a lead-acid battery (not good for the environment.)

Date: unknown

Date: 6/28/2019

Let’s choose the components:

Solar panels:

The solar panesl are from Soliculture. They let all light except for red light (bounces off)

100W LUMO 20M100GH

10.6V, 9.50A.

Voc = 13.3
Isc = 9.96A.
Dimensions: 64.9” x 38.0” x 0.165”
17kg

Temperature coefficient:

Pmax: -0.39% / C
Voc: -0.31% / C
Isc: -0.045% / C

Max system voltage: **1000V**
Max series fuse rating: 15A
We have 12 solar panels in series.
This gives us 120V of power at 14A (maximum) or 15A (max fuse rating)

That’s the one she wants, so we’re going with it.

Date: 6/28/2019

Charge controller: It must be a good one.
Capable of working with solar panels.

We could stay with the Tristar. Let’s do that.

Tirstar Morningstar MPPT 150V.

TS-MPPT-60

Up to 60A of continuous battery current.

Date: 7/5/2019

Engineering Schematic:

All of the interconnections between the ports.

Wiring diagram:

How you connect them.

First, start with the block idagram.

Create schematics in KiCad.
Multipage scheamtics.

All pages will fully express anything you’ve designed.

Use Capture to design a set of engineering schematics.

Create multiple sheets that show interconnections.

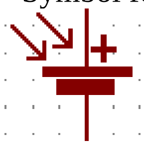
Do this in Capture. It becomes a working diagram.

Date: 7/5/2019

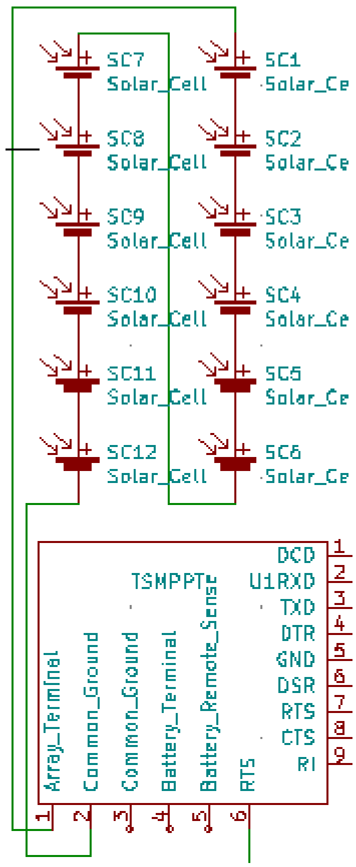
Use engineering schematics to create a wiring diagram.
Wiring diagrams are for non-engineers to connect things together.

Date: 7/8/2019

Symbol for Solar Cell / Photovoltaic cell

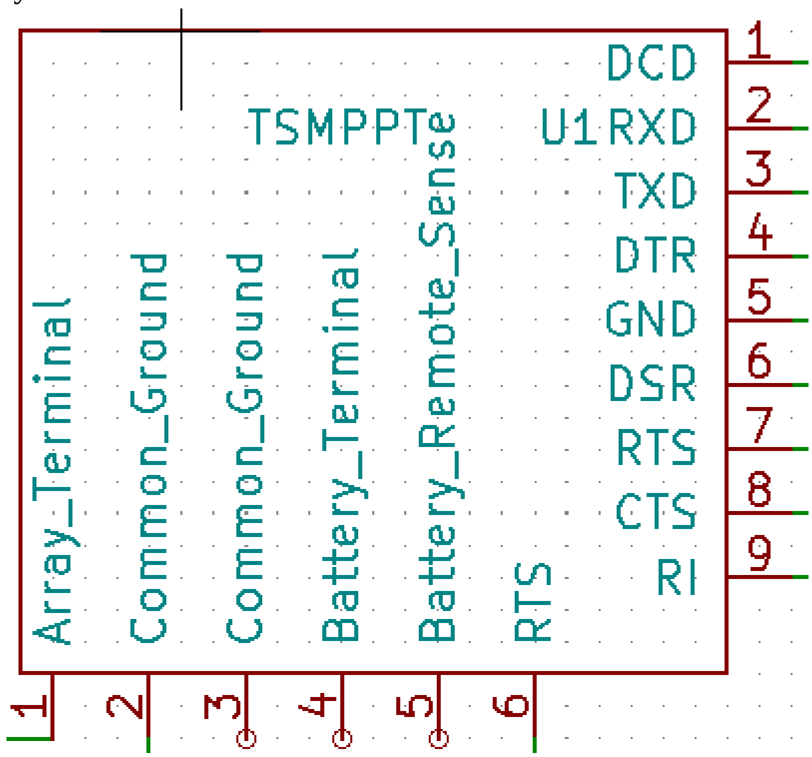


Engineering Schematic Draft 2



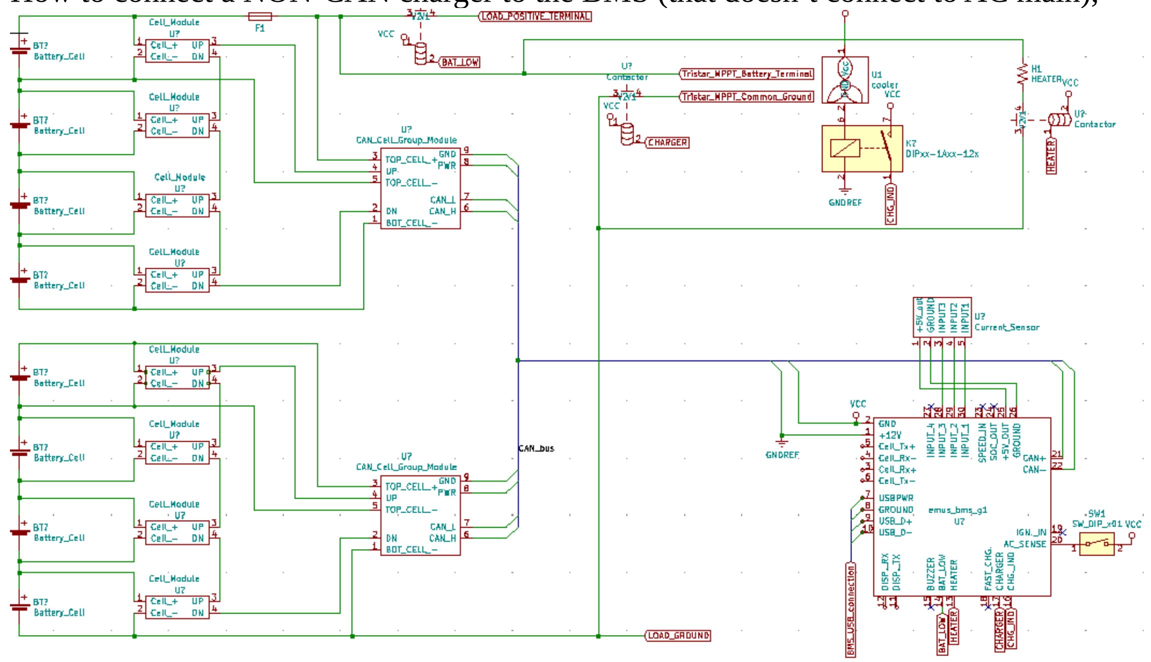
Date: 7/8/2019

Symbol for TSMPPT:



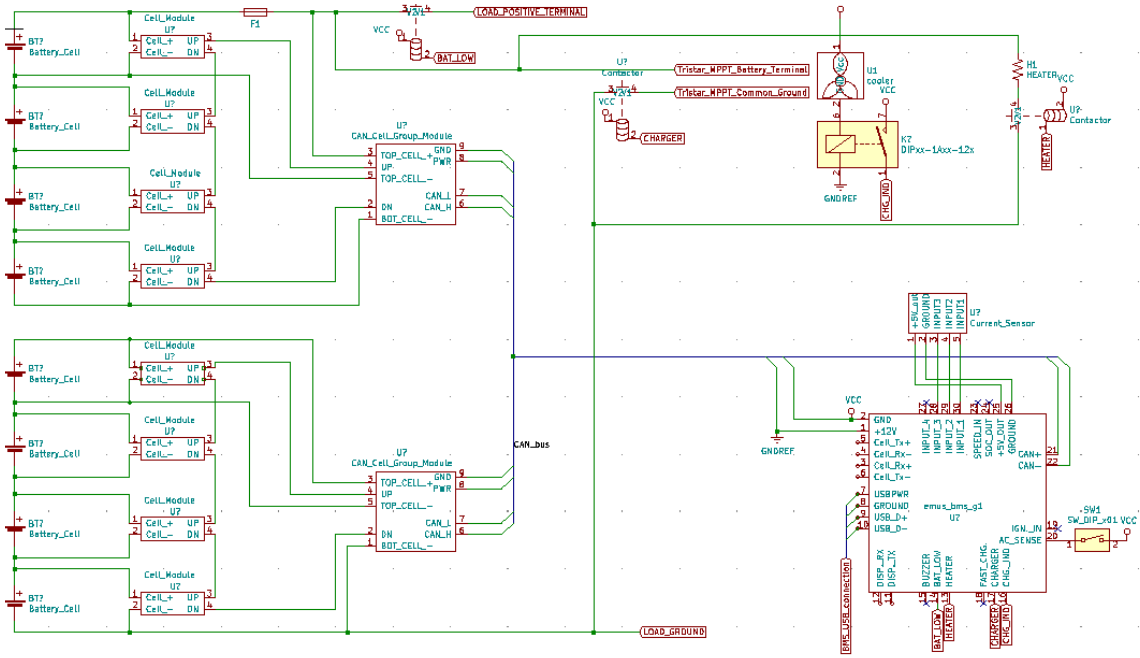
Date: 7/14/2019

How to connect a NON-CAN charger to the BMS (that doesn't connect to AC main),



Date: 7/15/2019

How to connect a EMUS BMS current sensor the BMS:



Date: 7/15/2019

How to connect a cooling fan and a heater: