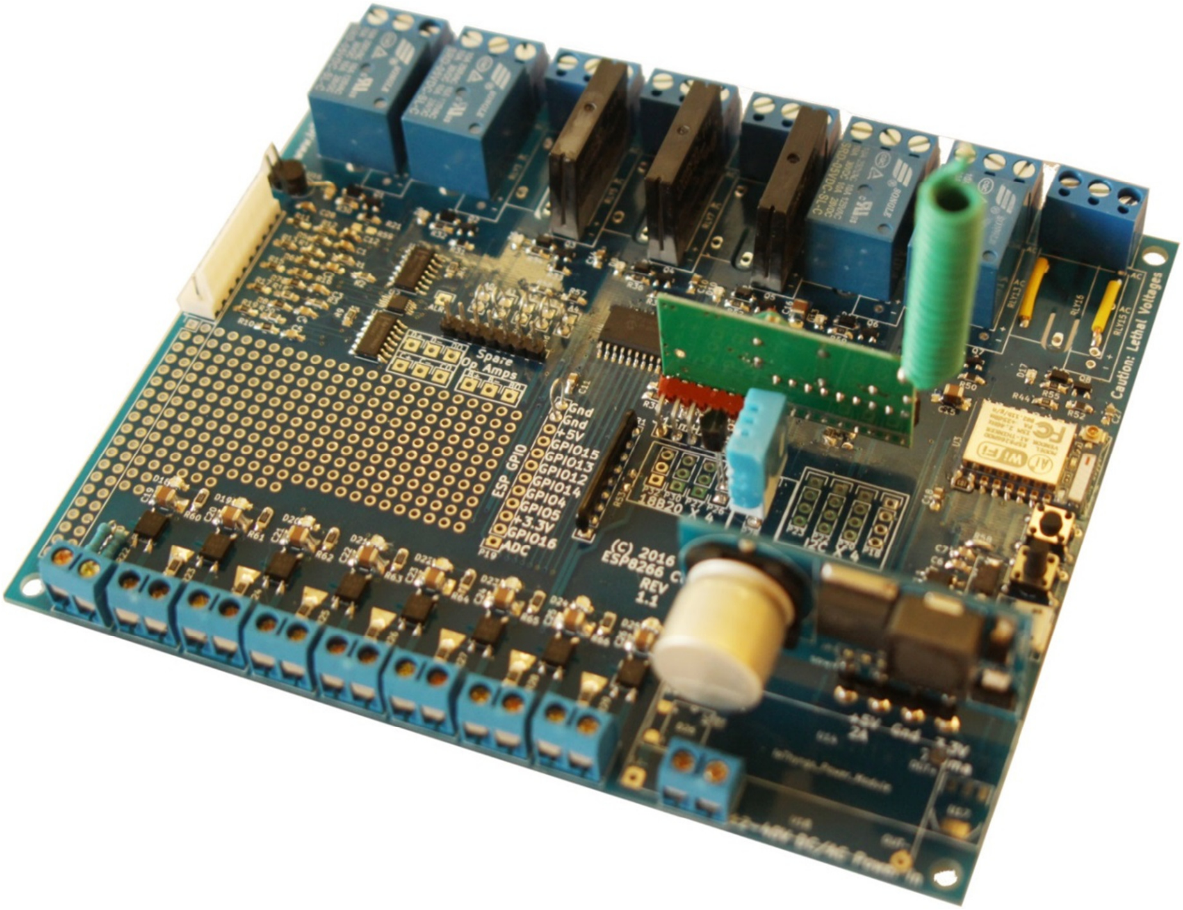
ESPEveryThyng-885 Controller Board

User Manual

Thyngs to Rule Your World!



Rev 1.0 July 2016

ESPEveryThyng-885

Table of Contents

Table of Figures

Thanks

Thanks for purchasing the IoThyngs ESPEveryThyng-885 board! This board is one of the first boards on the market to integrate a Wifi capable processor with opto-isolated inputs, configurable relay outputs, signal conditioned analog inputs, wide range power input and generous expansion capability all on one small board.

This board is compatible with any ESP8266 firmware currently available or it can be easily programmed using an ESP configured Arduino Integrated Development Environment, LUA programming and BASIC code through a web interface.

This board is ideal for:

- Pool filter/pump system control
- Expandable sprinkler systems
- Hot-house and hydroponics controller
- Outdoor lighting control
- Solar system monitoring and control
- Alarm systems
- Gate / Garage and entry system control
- HVAC control with multiple zone monitoring
- Wine cellar and Humidor controller
- And Much More…

Depending on the installed firmware the ESPEveryThyng is compatible with Thingspeak, OpenHab, Domoticz, NodeRed, MQTT and many other Home control software products.

Introduction

The ESPEveryThyng-885 is a general purpose Wifi enabled controller board. It has 8 optoisolated, High voltage capable inputs, 8 configurable relay outputs, 5 signal conditioned analog inputs and a wide range power input (9-40 VDC or VAC) using the IoThyngs Power Module. The individual sections below will describe each section of the board in detail.

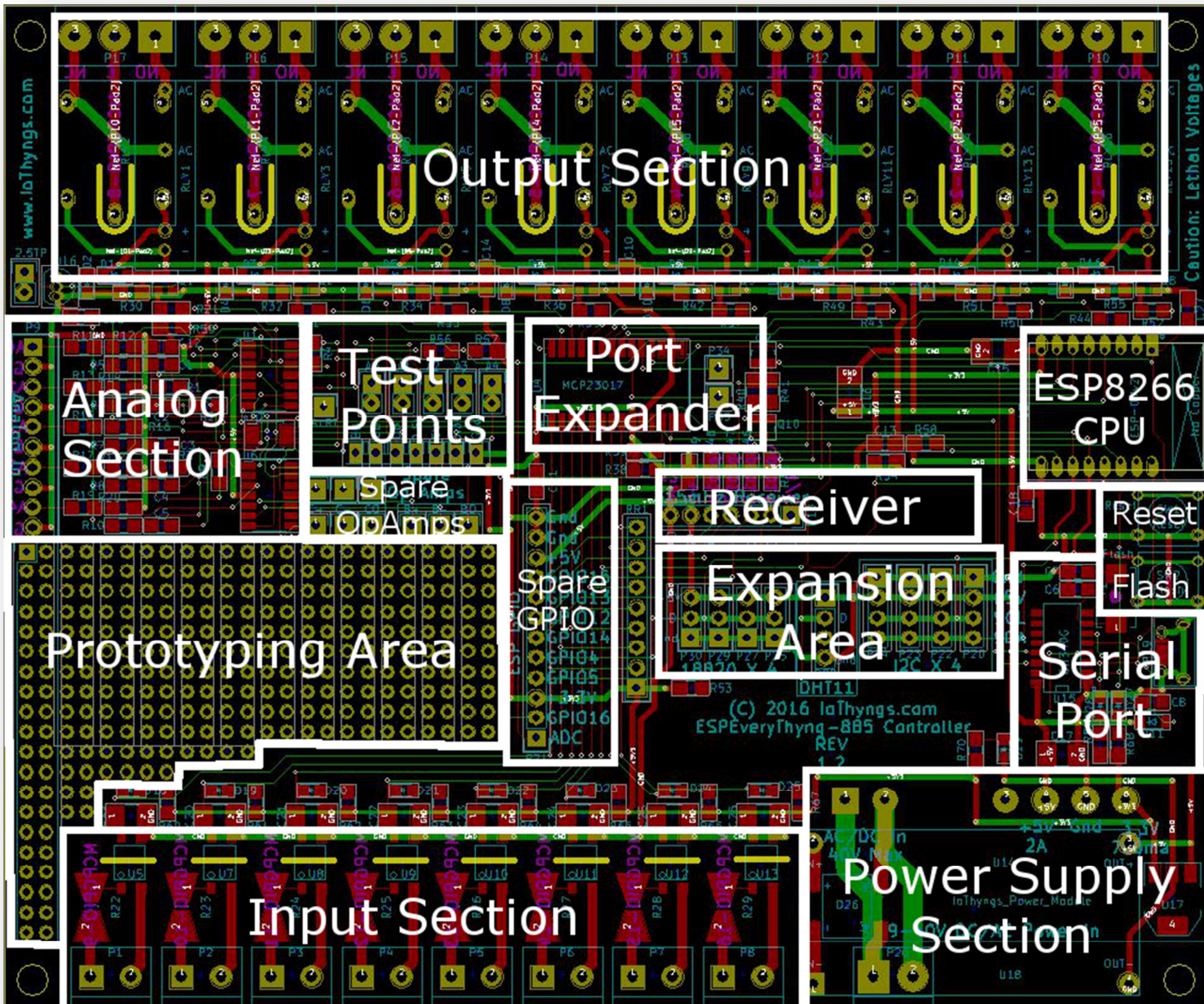


Figure 1 ESPEveryThyng-885 Controller board

Hardware

Power Supply

The power supply can be one of two types. The preferred power supply is the IoThyngs Power Module. This module guarantees a wide range input of 9-40 volts and can be supplied either AC as from an inexpensive 24VAC sprinkler valve transformer or DC from a wall transformer or power supply. The Power module supplies 5VDC at 2 Amps and 3.3VDC at 600mA.

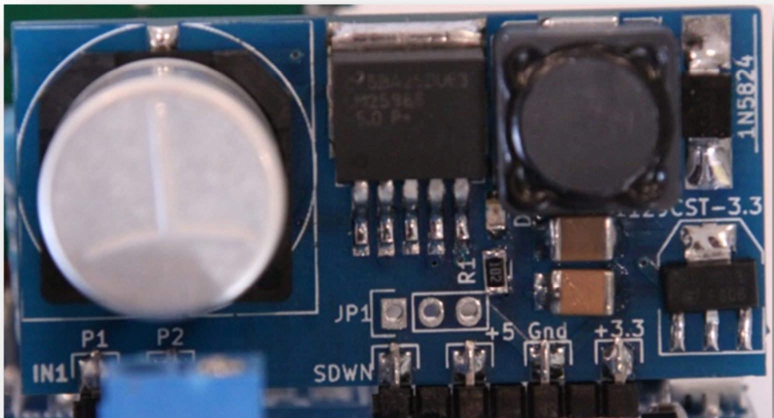


Figure 2 IoThyngs Power Module. 9-40 VDC or VAC input 5V and 3.3V 10 Watts output

The ESPEveryThyng-885 also supports the less expensive LM2596 power modules from China. These modules are good as well but they do suffer from a few negatives. The power output from these modules is adjustable and must be set to 5V +/- .5V before installing and applying power. Another downside is the size of the filter capacitors supplied may be insufficient to eliminate ripple from noisy power sources. This may also reduce the input supply range. If using these modules the power input should be narrowed to 12-36VDC input. If using a Chinese power module then pads are provided on the ESPEveryThyng-885 PCB for a bridge rectifier and 3.3V regulator to allow AC input and to provide the required voltage to run the CPU.

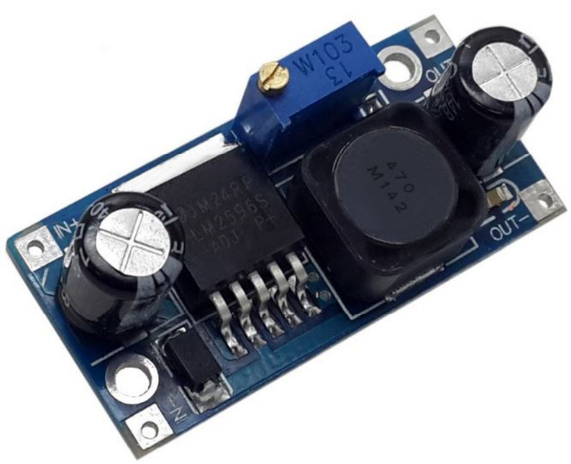


Figure 3 Adjustable LM2596 Module. 9-36VDC or VAC input

Processor

The ESPEveryThyng-885 board uses one of two ESP8266 modules for a processor. The board will use either an ESP8266-12 (Type E or F should work as well) or an ESP8266-07 Module. The ESP8266-07 module has 1024 Kbytes of flash memory and both internal and external WIFI antenna. The ESP8266-12 module has 4096 Kbytes of flash memory and a few extra connections that are not used by the ESPEveryThyng-885 board. Both modules will run most of the available firmware without problem. The external antenna connector on the ESP8266-07 may be required if the controller is put into an all metal junction box.

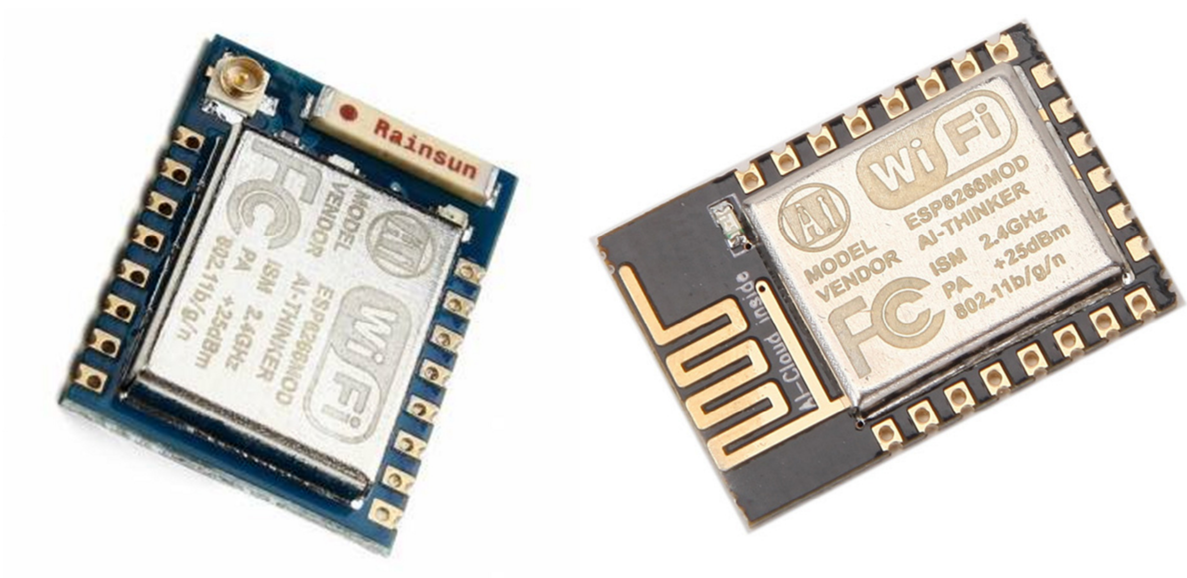


Figure 4 ESP8266-07 (left) ESP8266-12E (right)

Digital Input Section

The input section of the ESPEveryThyng-885 is an 8 bit optically isolated digital input. It can accept a signal from 3.3VDC to 240VAC with proper resistor selection (See Resistor Selection section). P1 – P8 are the input connections for bits GPB0 – GPB7 of the MCP23017 port expander chip. The GPB port pins will be normally high and will toggle low when an input is true (LED of the optoisolator is lit). There is a capacitor on the output of each optoisolator to smooth AC inputs.

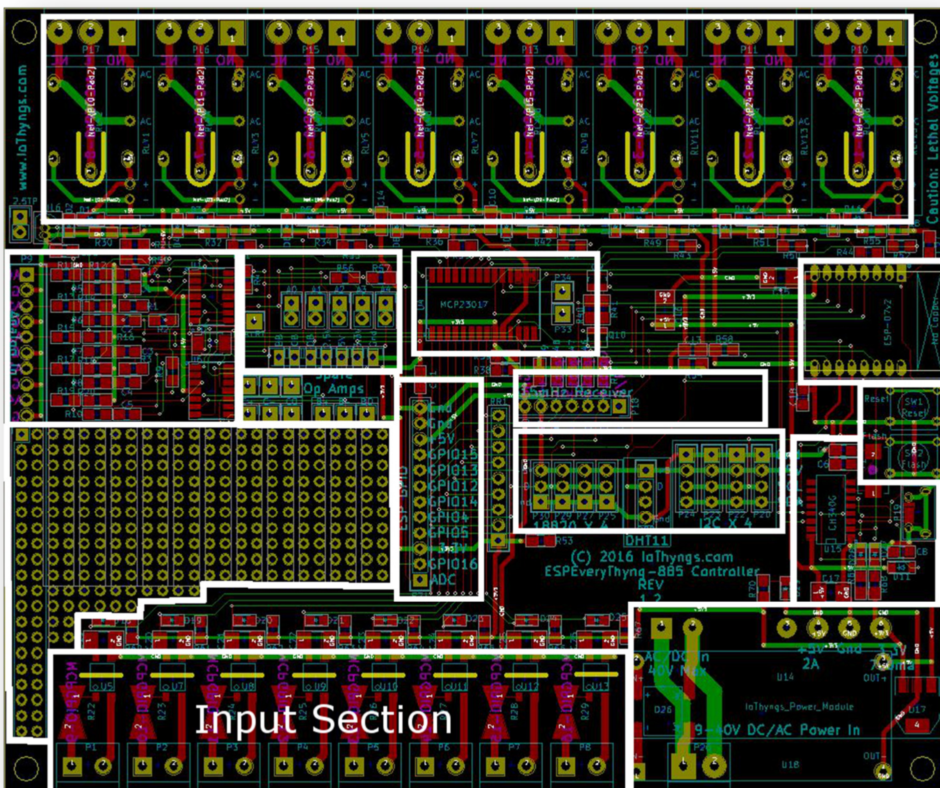


Figure 5 Input Section. 3.3V - 240V AC or DC input. Input resistors must be calculated and installed for selected voltage range

Input Optoisolators

The inputs are optically isolated through 8 LTV354V optoisolator chips. These optoisolators are unique in that they have two opposing LEDs on their inputs, which allow the optoisolators to be used with DC as well as AC signals. This permits the use of commonly available sensors such as low cost light sensors available from home and garden stores or high voltage motor travel limit switches from gate controllers etc. There is a routed cut in the PC board between the input and output sides of the optoisolator which provides increased dielectric strength to prevent arcing for high voltage inputs.

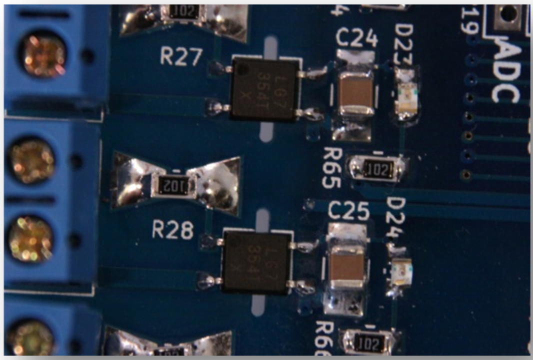


Figure 6 Input resistor to select input voltage range.

Resistor Selection

R22 – R29 are used to select the desired voltage input range. Before an input can be used it must be determined what type of sensor will be connected and within what voltage-range the sensor will operate. An appropriate current limiting resistor can then be selected and installed. The maximum current input for the LTV354V optoisolator is 50 milliamps. The minimum operating current is 3 milliamps. A resistor value must be picked so the operating current falls between these two values. Once the resistor value is determined then the correct power dissipation must be determined. The controller PCB will accept up to 1 watt resistors. To make resistor value selection easy a table is provided for common values. The formula to calculate a resistor value is: $\frac{Voltage}{Current} = Resistance$. Current is in Amps and resistance is in Ohms. Example: A light sensor from a garden store has a 120 VAC output when the sun is up. We would like the lowest power dissipation to reduce heating on the controller. The lowest usable current is 3 milliamps. This works out to: $\frac{120V}{.003A} = 40{,}000\ Ohms$ or 40K ohm. To calculate the required power dissipation we use the formula: $Current^2 * Resistance$ or $.003^2 * 40K\ Ohms = .36\ Watts$. For this example we will use a 40K ½ Watt resistor.

Sensor Voltage	Optoisolator LED current	Resistor Value	Resistor wattage
3.3	10 ma	330 ohm	1/8 watt
5	10 ma	500 ohm	1/8 watt
12	10 ma	1200 ohm	¼ watt
24	10 ma	2400 ohm	¼ watt
48	5 ma	9600 ohm	¼ watt
120VAC	3 ma	40K ohm	½ watt
240VAC	3 ma	80K ohm	1 watt

Table 1 Common Input Resistor Values

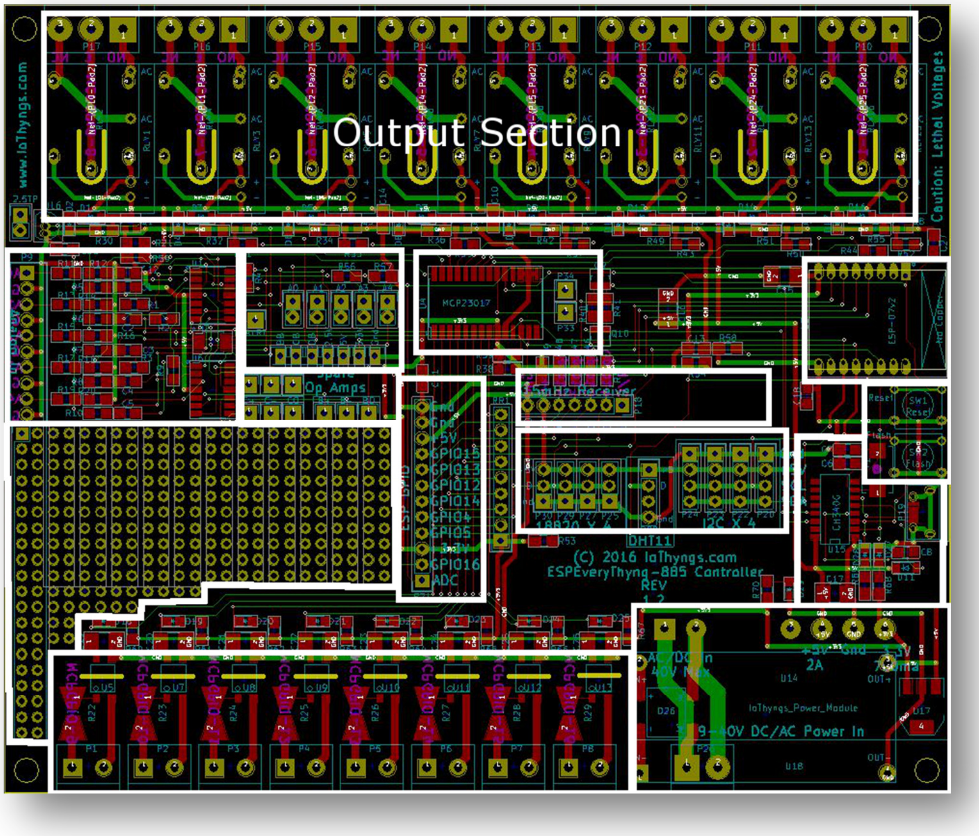


Figure 7 Output section. Can install mechanical, Solid State or jumper for external relay

Output Section

The output section of the ESPEveryThyng-885 supports 8 relay outputs. The board can support 2 different relay types directly. Pads are provided for both an SRD-5VDC-SL-C mechanical relay and an Omron G3MB-202P Solid State AC relay. Only one type of relay can be installed in any output port position. The controller supports 8 output channels that correspond to GPA0 – GPA7 (Connectors P10 – P17 respectively) of the MCP23017 port expander chip.

Relay Selection

The SRD-5VDC-SL-C relays can be used to switch 10 amps, AC or DC max. The G3MB-202P relays can switch 2 amps max AC only. External large current solid state relays can be used with the ESPEveryThyng-885 by adding jumpers as shown in Figure 8. External contactors can be used by using an installed relay to drive the contactor coil. In this manner the ESP controller board can switch very large currents which may be required for pumps, motors or HVAC applications. Notice the semi-circular routed slot around the common pin of the mechanical relay footprint. This is used to increase the dielectric strength of the PCB to prevent arcing between the low voltage DC control signals and the switched side of the relay. This allows the design to handle 10 Amps at 240 VAC.

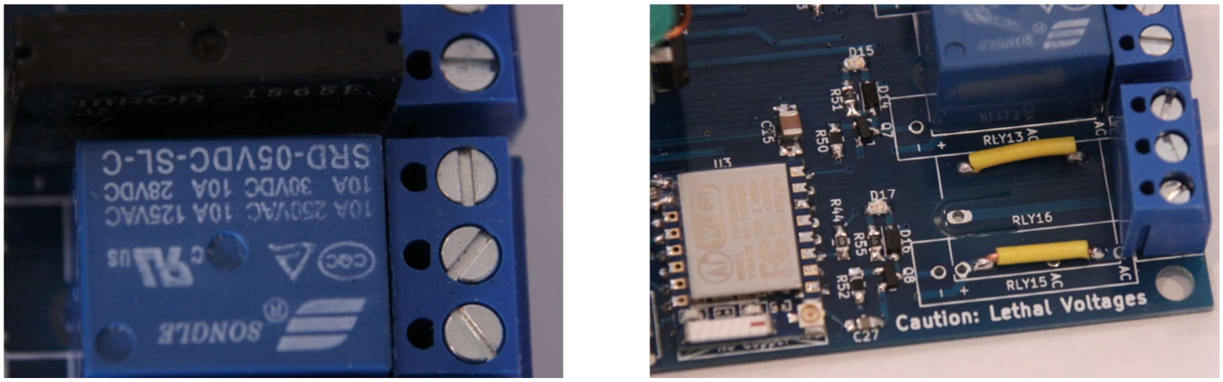


Figure 8 Mechanical and 2A SSR (left) Jumpered for external SSR (right)

Analog Section

The analog section of the ESPEveryThyng-885 supports 5 analog input channels. Each channel can accept up to 5 volts input signal and all channels have a voltage divider on the input to scale sensor signals that may exceed 5 volts. An opamp signal conditioner is provided for all channels with resistor programmable gain. A capacitor in the feedback allows for a simple anti-aliasing filter. The first channel (CH0) is 10 bit and corresponds to AD0 of the ESP8266 module. AD0 of the ESP8266 can only support 0-1 VCD input. There is an additional voltage divider on the output of the opamp for channel 0 to enable scaling to limit the output to 1 volt. The tables below show common configurations. Common configurations for the analog inputs are: 0-1 VDC, 0-3.3 VDC, 0-5 VDC, 0-10 VDC and 4-20 ma type signals.

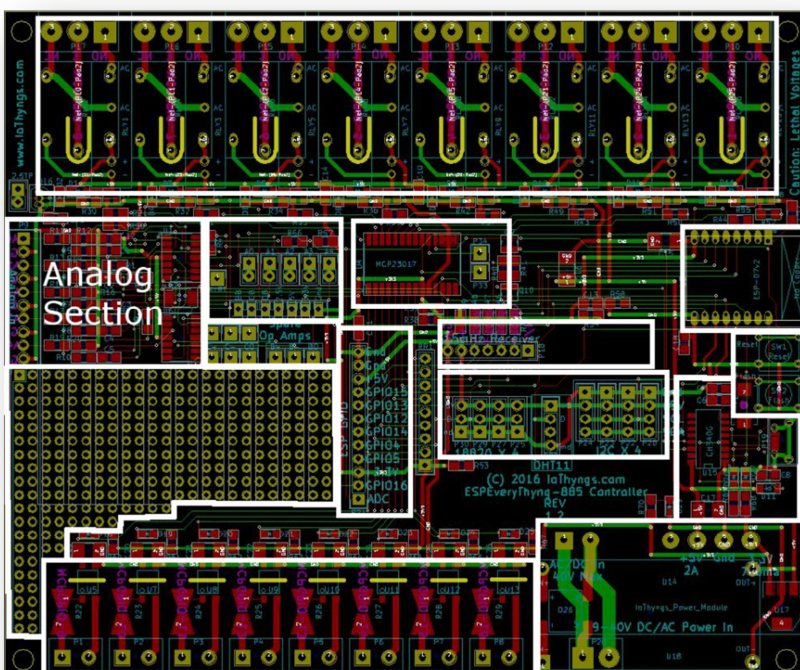


Figure 9 Analog section

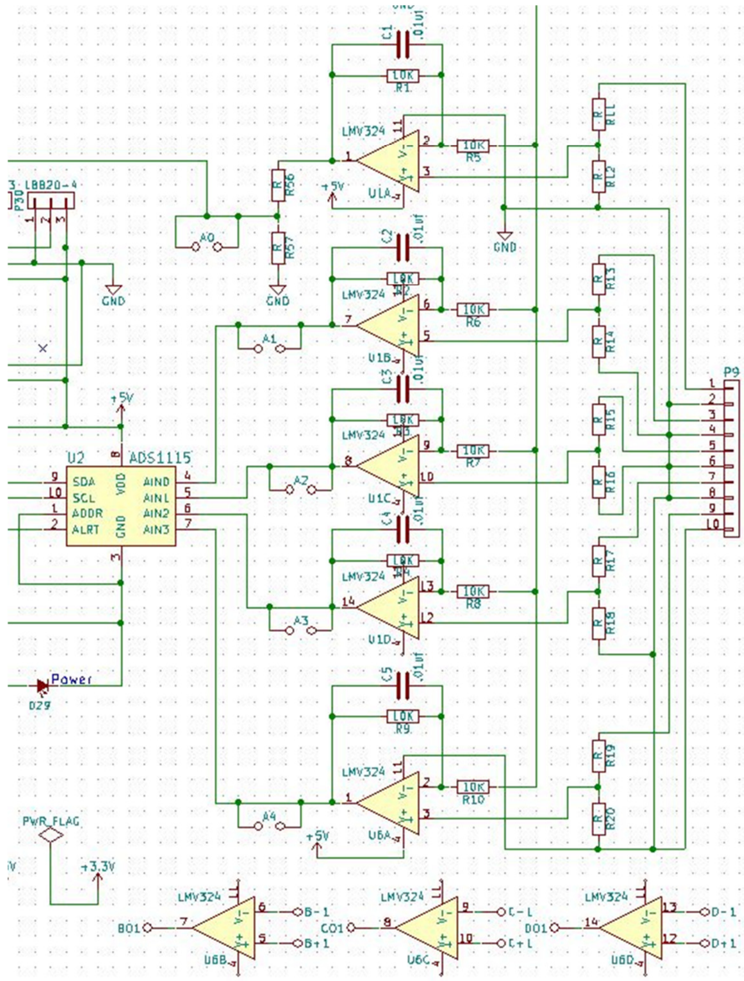


Figure 10 Analog section schematic. Unity gain configuration shown

Amplifier Section Resistor and Capacitor Selection

The ESPEverythyng-885 uses a non-inverting amplifier configuration to condition and scale the analog inputs. A single amplifier section for channel 1 is shown in Figure 11.

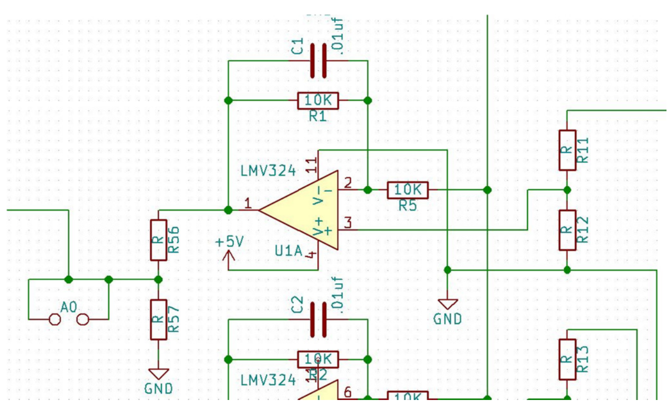


Figure 11 Non-inverting amp for channel 1. R11 & R12 are input scaling. R1 & R5 set the gain. R56 & R57 are output scaling.

Input Scaling

R11 and R12 are used to scale the input signal. The input to the amplifier should be scaled to be no more than 5 volts. If no scaling is required (signal is less than 5 volts) R12 may be omitted and R11 should be 1K to 10K. The formula for the input scaling voltage divider is:

$$Vout = Vin\,\frac{R12}{R11 + R12}$$

Remember that Vout is 5 volts max. If the input signal is 0-10 V and we need 0-5 V then the divide ratio is ½. This makes it easy. If both R11 and R12 are set to 10K the input signal will be divided by 2. The input Scaling should also take into account the Amplifier gain setting.

Amplifier Gain

R1 and R5 set the gain of the amplifier. Under most conditions the amplifier should be operated at a gain of 2 (output = 2X the input). If the input signals are small then the gain resistors can be calculated to amplify the signal to provide a 0-5 V signal to the analog to digital converter. The formula for the gain resistors is:

$$Vgain = 1 + \frac{R1}{R5}$$

As can be seen from the equation above if R1 and R5 are the same value, 10K for example, the total gain will be 2. Unity gain can be achieved by setting R1 to 0 ohms and eliminating R5 altogether.

Example

If we have an input signal that is 0 – 1 V (after the input scaling) and we want an output signal that is 0-5 V then the ratio of R1 to R5 will be 1/5. R1 could be a 2K ohms and R5 could be a 10K ohms. It is best to keep the values of R1 and R5 between 1K and 100K. Values that are too low or too high could excessively load the amplifier output or cause the amplifier to be unstable.

Signal Filtering

Capacitor C1 is used to add a simple low pass filter to the amplified signal. This capacitor may be eliminated if no filtering is desired. The filter can help to smooth a noisy signal. The value of C1 sets the cutoff frequency of the filter. The formula for setting the filter frequency is:

$$C = \frac{1}{2\pi\, fc\, R1}$$

C is the value of the capacitor in Farads, fc is the desired cut-off frequency in Hertz and R1 is the value of the feedback resistor that was calculated in the gain equation above. If we want to filter out AC hum noise we could make the cut-off frequency below the line frequency (60Hz for the US and 50Hz for Europe etc.). Let's pick 15Hertz as a cut-off frequency well below the normal AC hum frequency. The formula would be:

$C = \frac{1}{2\pi * 15 * 2000}$ = .00000530516477 Farads. Multiply by 1000000 to get uf (micro farads) or 5.3uf. A common capacitor value is 4.7uf and would work fine here (at 4.7uf cut-off frequency would actually be 17 Hz. Close enough!)

See www.IoThyngs.com in the TOOLS section for online calculators to assist selecting input scale, gain and output scaling resistors as well as a calculator to help with filter frequency and capacitor values.

Channel 1 Configuration Differences

Channel 1 is the only channel with a voltage divider after the amplifier. This is because channel 1 is sent to the Tout (A0) pin of the ESP8266 processer. This is a 10 bit analog input with the capability for 0 – 1 volt full scale. It is important to keep this signal within the 0 – 1 volt range. Because OpAmp U1A is 0-5 V out the channel 1 output divider resistors are set to divide by 5. R56 is 10K and R57 is 51K. These resistor values are standard values and give a divide value of 5.1 or .98 volts max output. This is close enough to full scale.

Channel 2 – 4 Configuration

Setting the resistor and capacitor values for channels 2 – 4 is exactly the same as the calculations for channel 1.

Resistor Tolerance

Common chip resistors have a tolerance of +/- 5%. This may be OK for most things. If more precise measurements need to be made then +/- 1% metal film resistors are best. The 1% tolerance will assure that channel calibrations will behave similarly. A metal film resistor is more stable over a wide operating temperature. If common carbon resistors are used their values may drift more as the operating temperature changes. If the ESPEveryThyng-885 is mounted in an weather proof junction box, a connected sensor could then read differently in the cold of winter verses the heat of summer.

Spare OpAmps

U6-B, C and D on the schematic in Figure 10 are spare LMV324 opamps. These opamps are made available to the user for the prototyping area via headers B-, B+, BO, C-, C+, CO, D-, D+ and DO.

Biasing Unused OpAmps

Unused inputs should be biased properly to prevent the amplifiers from oscillating or driving to the power rails. This could cause unnecessary power supply drain or opamp over-heating. To bias unused opamps, jumper the output pin (BO, CO or DO) to its inverting input pin (B-, C- or D- respectfully) and then jumper the non-inverting input (B+, C+ or D+ respectfully) to 2.5 Vref. Pads are provided near the test points to do this. The biasing pins are labeled BB, CB and DB. These pins are connected to the 2.5 V amplifier reference voltage.

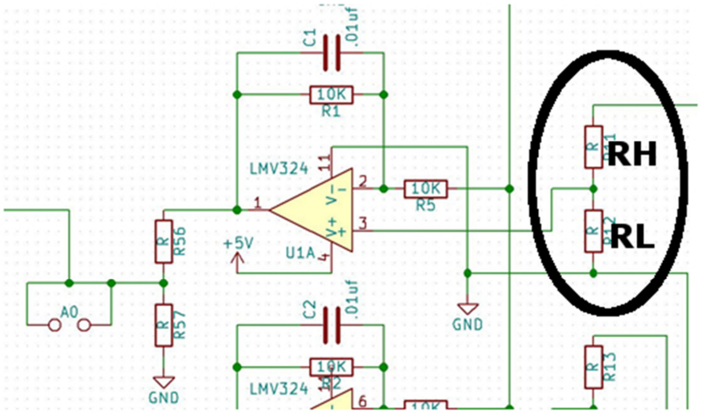


Figure 12 Analog Input Resistor Divider. Same for all 5 Channels

Signal Type	*RH*	*RL*
0 – 1 V	10K	None
0 - 3.3 V	30K	10K
0 – 5 V	10K	10K
0 – 10 V	20K	10K
4-20 ma	0 Ohms (Jumpered)	250 Ohms

Table 2 Input Voltage Divider Resistor Selection for Common Inputs

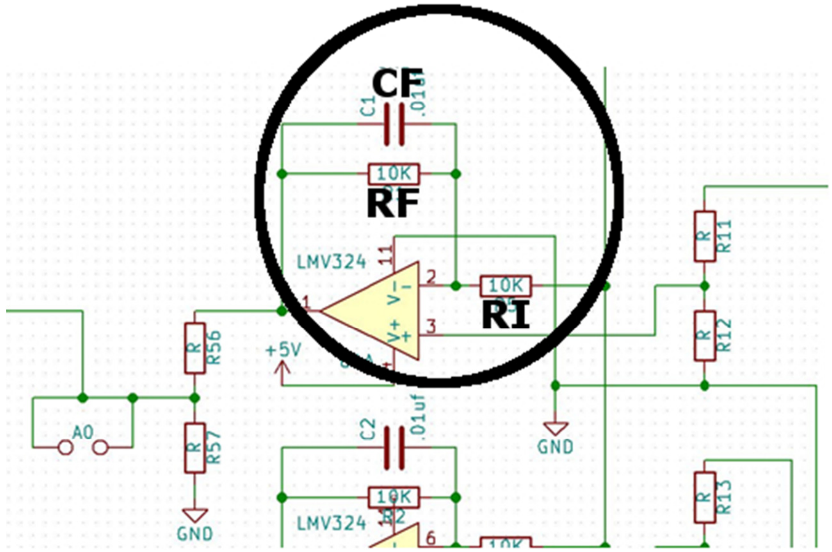


Figure 13 Gain Resistors and Filter Capacitor. Same for all Channels

Signal Type	*RF*	*RI*	*Gain*
0 – 1 V	39K	10K	4.9
0 - 3.3 V	39K	10K	4.9
0 – 5 V	10K	10K	2
0 – 10 V	10K	10K	2
4-20 ma	10K	10K	2

Figure 14 Gain Resistors for Common Signal Inputs

Cut-Off Freq	*RF*	*CF*
1Hz	39K	3.9uf
1Hz	10K	15uf
25Hz	39K	.15uf
25Hz	10K	.68uf
50Hz	39K	.082uf
50Hz	10K	.33uf
60Hz	39K	.068uf
60Hz	10K	.27uf
100Hz	39K	.039uf
100Hz	10K	.15uf

Figure 15 Filter Capacitor Selection for common Cut-Off Frequencies

Analog Channel Calibration

Calibration of each channel is important to get an accurate measurement. The analog-to-digital converter (ADC) will convert the voltage presented to its input and return a count value. The count value will be an integer number between 0 and the maximum resolution of the ADC. For the ESP8266 ADC this is 10 bits or 1024. For the ADS1115 channels this is 16 bits or 65535. The goal is to convert the ADC counts (ADACs) into a meaningful engineering unit (Volts, current, pressure, temperature, flow etc.). There are several ways of performing a channel calibration. 1) A simple theoretical calibration, 2) A linear calibration based on several measurements, and 3) A polynomial fit calibration for sensors that are non-linear.

The type of calibration will be determined by the required accuracy. If a precise measurement is required or the calibration of a particular sensor is not known then the theoretical calibration may not be good enough. Many types of sensors are factory calibrated and will come with a certificate describing their precise output. Many sensors are factory adjusted to perform to a standard output. Commonly available pressure sensors are a good example of the latter. A 0 – 100 PSI sensor might have an output signal of .5 to 4.5 volts.

Theoretical Calibration

A theoretical calibration can be done by taking the full scale input voltage and dividing by the ADC resolution. If the input voltage is 5 volts max and the ADC has 16 bit resolution then the theoretical calibration is $\frac{5V}{65535} = .000076\ volts\ per\ bit$. If we have a pressure sensor that has a calibration certificate that says 0 – 100 psi = .5 to 4.5 volts then 100 psi will be 58982 ADC counts and 0 psi will be 6554 counts. Since 0 psi does not equal 0 volts we need to compute the total ADC count spread. This is: $CountHi - CountLo$. In this case: 58982 – 6554 = 52428. PSI per ADC count is calculated by $\frac{Maxpsi}{ADCCountspread}$ or $\frac{100psi}{52428} = 0.001907$ ADCcounts per Psi. As mentioned the signal does not start at 0 ADC counts. The signal begins at 6554 or .5 volts. 6554 * 0.001907 psi per ADCcount = 12.5 psi. If we simply multiply ADC counts by 0.001907 our value in PSI will be off by 12.5 PSI. The final formula to convert ADC counts to PSI is (ADCcounts * 0.001907)-12.5 = PSI.

Linear Calibration

A linear calibration is performed by injecting a series of signals and recording the results. 5 or 10 data points over the full scale input range will provide a good idea of linearity and calibration. Applying the formula Y=mX+b will allow ADC count data to be converted into engineering units. If the formula looks familiar, it is because it is taught in every beginning geometry class as the equation for a line on a Cartesian coordinate system. 'm' is the slope and 'b' is the Y intercept of the line.

The easiest way to compute Y=mx+b is to use a spreadsheet program such as Microsoft Excel or the free Apache Openoffice Calc. Both of these programs have functions to compute Slope, Intercept, r^2, and Standard Error. The OpenOffice Calc spreadsheet in Figure 16 shows data points taken from a calibration of the pressure sensor mentioned above. There were 11 data points taken from 0 to 100 PSI. A calibration like this would is done by comparing the measured ADC counts to a calibrated reference instrument. The pressure would start at 0 and then increased to 10 PSI as indicated on the reference sensor. The ADC count for that pressure is recorded in the spreadsheet. The pressure is then increased to 20PSI etc. In fact the exact pressure is not important here as long as the value on the reference instrument and the ADC counts are recorded in the spreadsheet.

PSI	ADC Count	Verification
0	6550	-1.6353073563
9.54	11801	8.41617378
19.26	17102	18.5633650776
29.76	22295	28.5038224268
39.55	27578	38.6165580664
49.78	32826	48.662296593
59.87	38005	58.5759550971
69.67	43206	68.5317260721
80.05	48492	78.6502043213
89.77	53734	88.6844576286
99.93	58971	98.7091399198

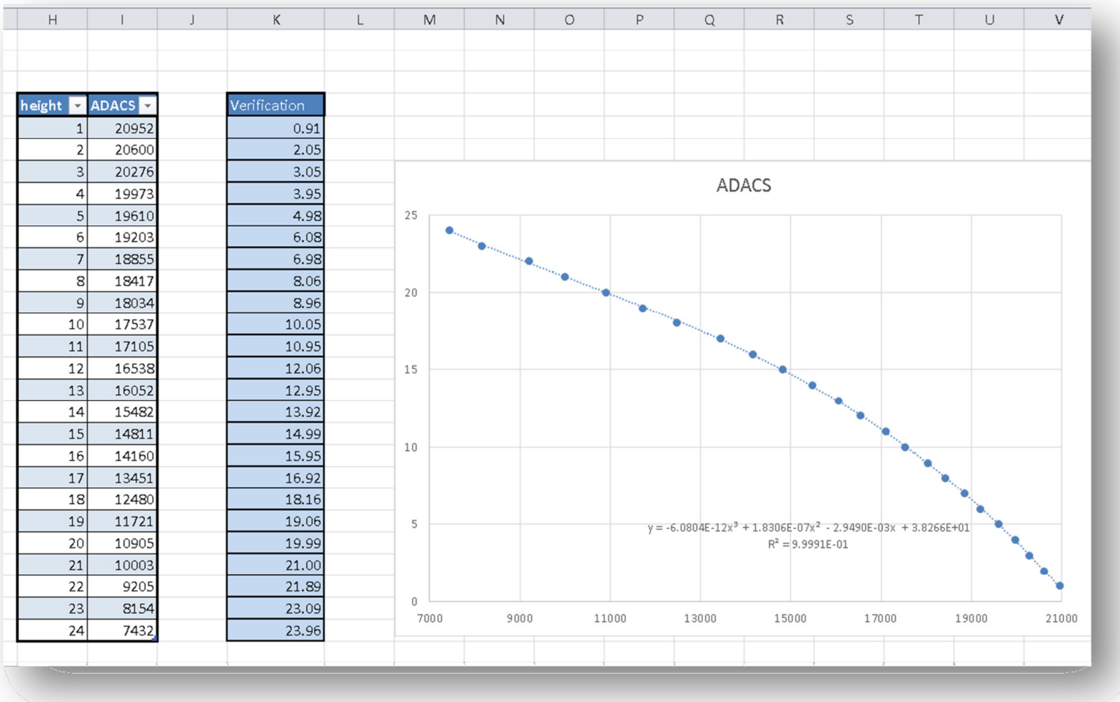
Slope	0.0019142032
Intercept	-14.1733384823
R^2	0.9999468757
Std Error	0.26%

Figure 16 Linear Calibration Using a Spreadsheet Program

The spreadsheet calculates the slope (m) and the intercept (b) by using the SLOPE(dataY,dataX) and INTERCEPT(dataY,dataX) functions. It also gives us 2 other valuable pieces of information. R^2 is a statistical way of determining how linear our data set is. The RSQ(dataY,dataX) function is used to get the R^2 value. This value should be as close to 1.0 as possible. The further away from 1.0 this value is, the less linear the results. The next value shown is the Standard Error. This is calculated using the STEYX(dataY,dataX) function. Divide this value by maximum Y value – minimum Y value (Max(Y)-Min(Y)) and set the data type of the cell to percentage and this gives us a percentage error over full scale. This gives an idea of how accurate the output can be expected to be, given the line that was regressed though the data set. The larger this number the less accurate the results will be. Finally as a verification of accuracy the right hand column labeled ***Verification***, shows the actual engineering unit we are calculating given the calculated slope and intercept. This verification value is simply the ADC count value multiplied by the computed slope value added to the intercept.

Polynomial Fit Calibration

If the sensor data is non-linear then it may be necessary to use a polynomial fit function to get an accurate calibration. As of this writing Openoffice does not perform this automatically. Excel does do this when plotting a function. One can enable display of the function formula. A value is given by the user to specify the order of the function. The example below is for a resistive tank level sensor. Just as was done in the linear calibration section, the sensor was lowered into a tank of water at many height levels. The ADC count data was recorded for each level. This particular sensor is 24 inches long so the sensor was lowered in 1 inch increments.



height	ADACS	Verification
1	20952	0.91
2	20600	2.05
3	20276	3.05
4	19973	3.95
5	19610	4.98
6	19203	6.08
7	18855	6.98
8	18417	8.06
9	18034	8.96
10	17537	10.05
11	17105	10.95
12	16538	12.06
13	16052	12.95
14	15482	13.92
15	14811	14.99
16	14160	15.95
17	13451	16.92
18	12480	18.16
19	11721	19.06
20	10905	19.99
21	10003	21.00
22	9205	21.89
23	8154	23.09
24	7432	23.96

Figure 17 Excel spreadsheet showing Polynomial Fit of a Level Sensor

Figure 17 shows the polynomial fit function for the calibration performed on the resistive tank level sensor. Excel will display the formula on the graph. In the above example the formula is Y= $(-0.0000000000060804 * \text{ADCcount})^3 + (0.00000018306 * \text{ADCcount})^2 - (0.002949 * \text{ADCcount}) + 38.266$.

The order of the equation can be increased or decreased to get the most accurate fit. This is determined by the R^2 just as it was in the Linear Calibration section. Typically a 3rd order polynomial is sufficient to approximate most sensor outputs. Also the higher the order of the function, the longer the value will be to enter into a user program. Some easy to use firmware may not be able to accept polynomial fit functions. If a 6th order polynomial function is necessary to describe a particular sensor output then the user may wish to use a programming language such as ESPLUA, ESPPYTHON or ESPBASIC to write a custom application to accept such a function.

Calculation Tools

At www.IoThyngs.com there is an extensive array of calculators that can help to calculate the resistors for scaling and gain, Capacitor for low pass filter cut-off frequencies as well as determine slope and intercept for linear calibrations and polynomial curve fits. There are also many on-line calculators to help in setting up your ESPEveryThyng-885.

Serial Communication and Programming

The ESPEveryThyng-885 has a micro USB port (P19) that is used for programming the ESP8266 module as well as general serial communications. The board has a CH340G USB serial port chip to facilitate this. LEDs D27 and D28 are serial transmit and receive indicators. They will flash upon serial data flow. Transistors Q9 and Q10 make up a circuit to put the ESP8266 module into programming mode. These transistors are controlled via the serial handshaking control lines RTS and DTR. Most ESP8266 flashing software will control these lines to automatically place the ESP8266 in programming mode just prior to flashing the memory.

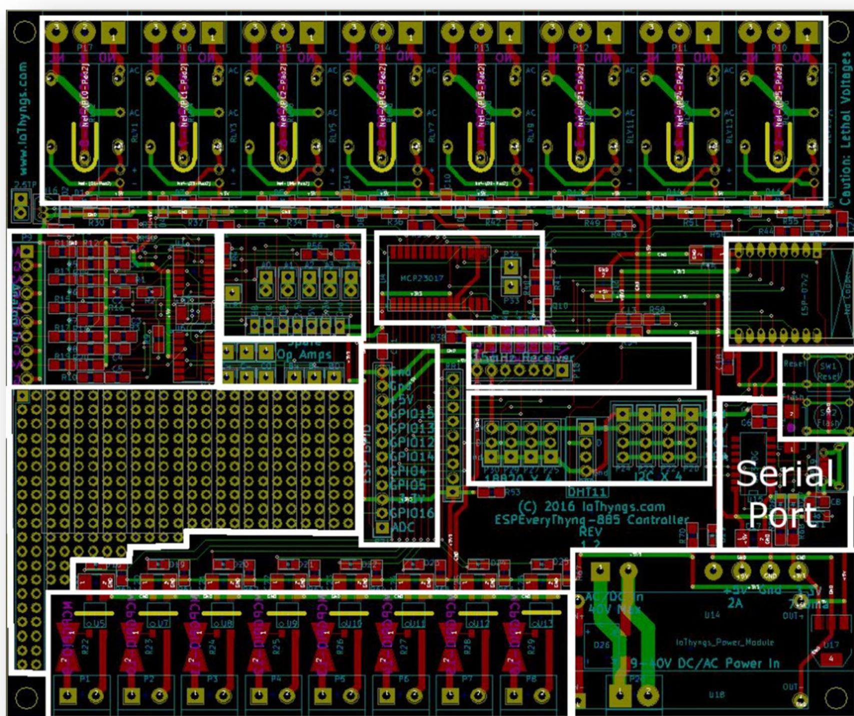


Figure 18 USB connector and Serial port Section

The ESP8266 can be programmed to use any standard baud rate, but several baud rates are common to most firmware. The bootloader firmware will offer its initial boot message at 74880 baud. Most firmware will then switch to 115200 baud. Common baud rates are 1200, 2400, 4800, 9600, 19200, 38400, 57600, 74880 and 115200.

Besides programming the ESP8266, the most common use for the serial port is diagnostic logging, where messages about software state or events are made available.

It is not recommended to use a USB cable longer than 15 feet.

Reset and Flash Buttons

If your ESP8266 flashing software does not control the RTS and DTR lines for automatic programming modes you may enter the programming mode manually by pressing and holding the FLASH button while briefly pressing the reset button.

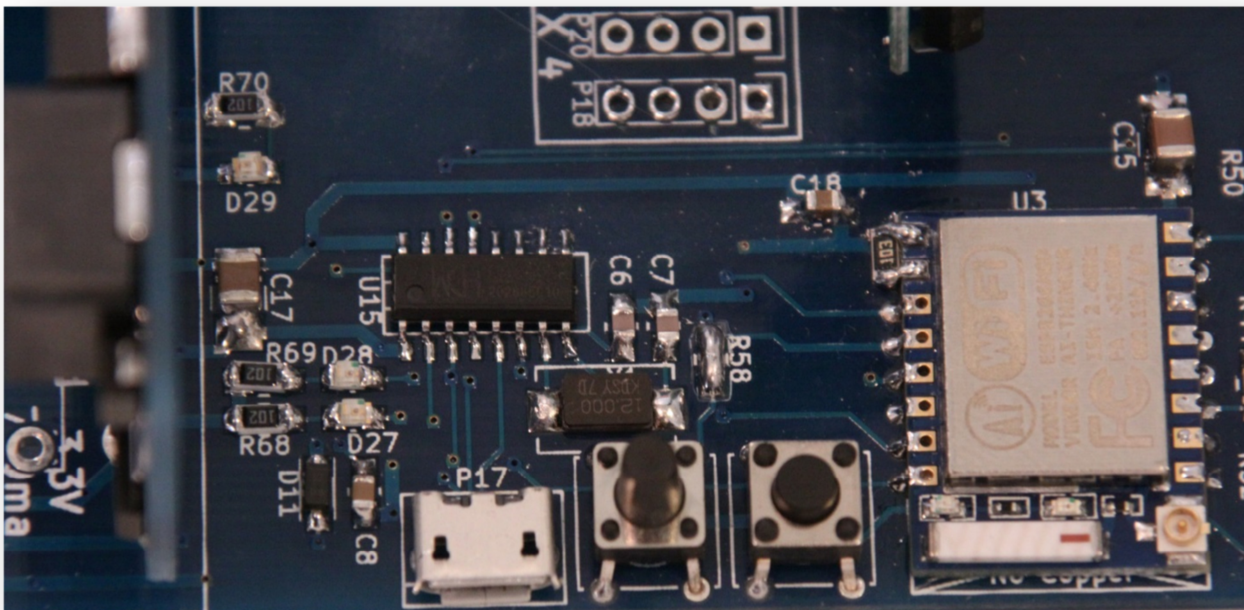
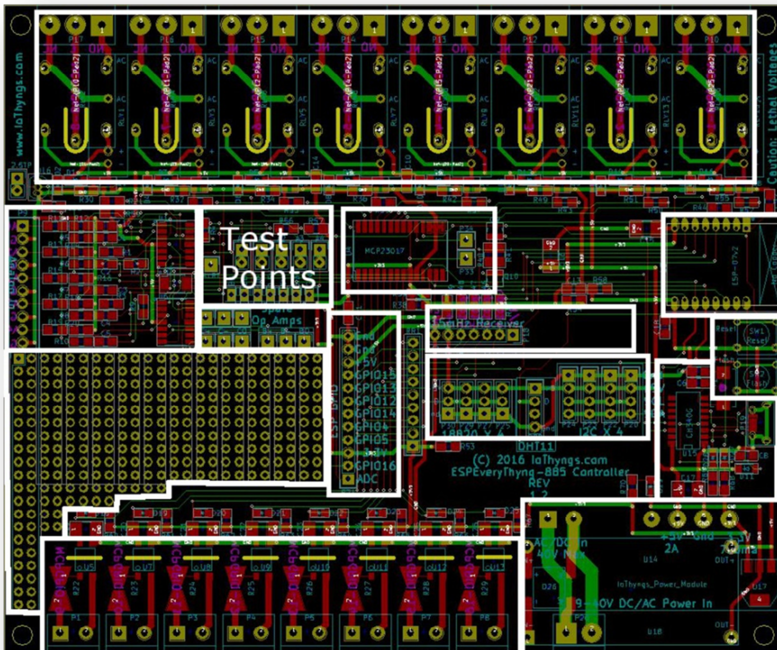


Figure 19 Reset (closest to the CPU) and Flash buttons (Next to the USB connector).

Test Points

Several test points are made available around the controller board. The test points are labeled and provide an easy way to check power supplies and analog voltages. Test points are provided for analog voltages on all channels (A0 – A4), ground, +3.3 volts, +5 volts and 2.5 volt opamp reference voltage. Figure 20 shows the locations of the test points.



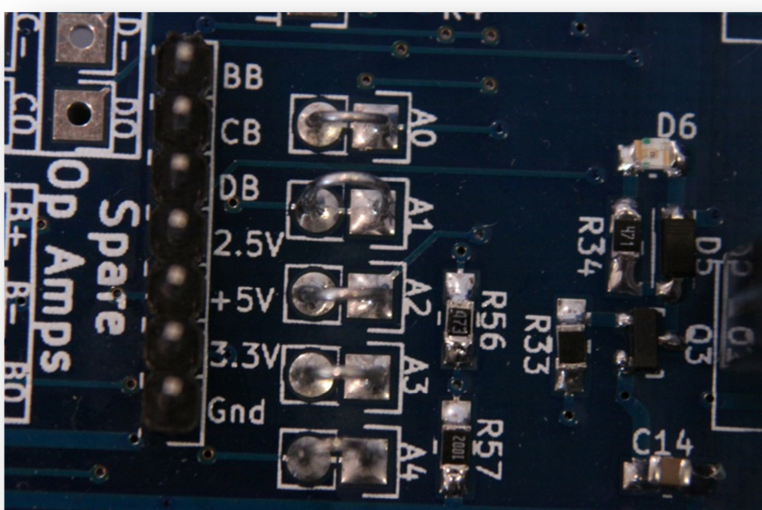


Figure 20 Test Points and Unused Opamp Bias Connections

Prototyping Section

The ESPEveryThyng-885 board has a generous prototyping area. This area can be used to add special or experimental functionality to the controller board. Situated around the prototyping area are headers that provide easy access to the +3.3 and +5 volt power supplies as well as the 2.5 V reference voltage. P21 is a connection point for ground, power supplies and spare ESP8266 GPIOs. Also just above the prototyping area are 3 headers labeled B-, B+, BO, C-, C+, CO, D-, D+, and DO. These connection points allow easy use of the three spare opamps on U6. Along the left hand edge of the prototyping area is a long double row of thru-holes that can accommodate up to a 2X20 pin header. This can be used for a ribbon cable connection to user circuitry in the prototyping area.

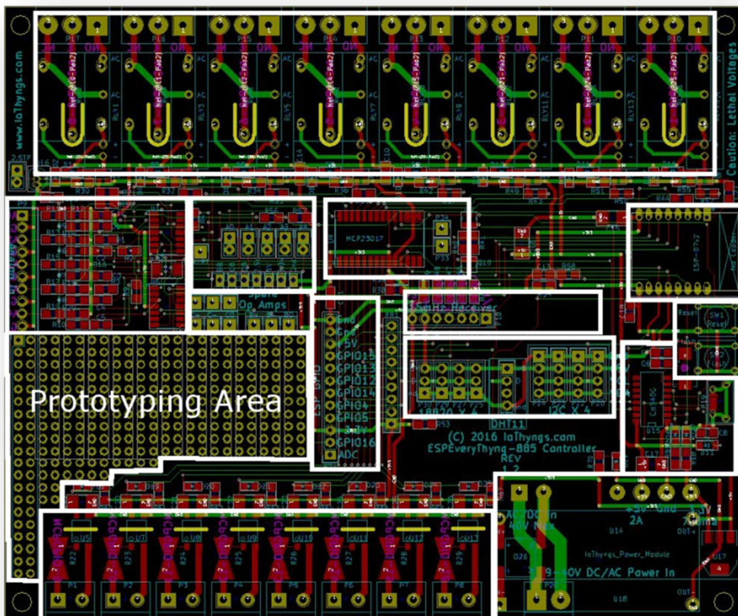


Figure 21 Prototyping Area

The thru-holes in the prototyping area are standard perf-board style .1 inch (2.54mm) hole-spacing double sided plated thru holes.

Expansion Ports

The ESPEverythyng-885 board has ample expansion capability. Headers P20, P22, P23 and P24 are I2C expansion slots. Headers P25, P27, P29 and P30 allow the connection of four DS18B20 temperature sensors header P28 allows the connection of a DHT11/22 Temperature Humidity sensor.

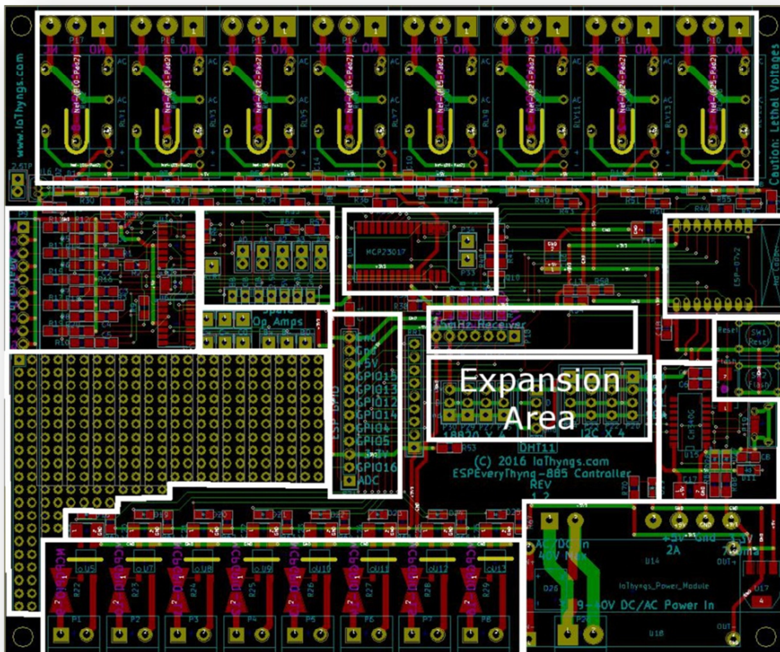


Figure 22 Expansion Section

DS18B20 Temperature Sensors

The DS18B20 is a common 1-wire temperature sensor with .5 degree C accuracy and a sensing range of -55 to +125 degrees C. The board can accommodate up to four DS18B20 sensors directly. This quantity of temperature sensors is very useful to track air temperature, water temperature, junction box internal temperature etc.

DHT11/22 Port

The DHT11 or DHT22 sensors are special temperature sensors with a built-in relative humidity sensor. The controller will accommodate one DHT11/22 sensor. The DHT22 sensor has better overall performance than the DHT11. The DHT11 has a humidity range of 20% - 80% 2% accuracy and a temperature range of 0 – 50 +/-2 degrees C. The DHT22 has a humidity range of 0% to 100% with an accuracy of 2%-5% and a temperature range of -40 to +125 +/- .5 degrees C. One other difference is the speed at which the sensors can be read. The DHT11 has a maximum update rate of 1Hz (Once per second) and the DHT22 has a maximum rate of .5 Hz (twice per second).

I2C Expansion Ports

The controller board will support directly 4 I2C devices. These can be LCD displays, port expanders, analog to digital converters and many more. Each device on the I2C bus will need a unique address. This is usually set on the device being connected (usually by jumpers or solder pads etc)

315mHz Receiver

Header P18 provides a place to connect a 315mHz four channel radio receiver. This is useful for remote activation of electric gates, garage doors, landscape lighting, manual override of pumps or irrigation, or light activation. The header is labeled. D0 – D3 from the module correspond to ESP8266 GPIO15, GPIO13, GPIO12, and GPIO14 respectively. The receiver will require the addition of a tuned antenna for maximum range. Line of sight reception can be as high as 60 feet (20 meters) with optimum conditions.

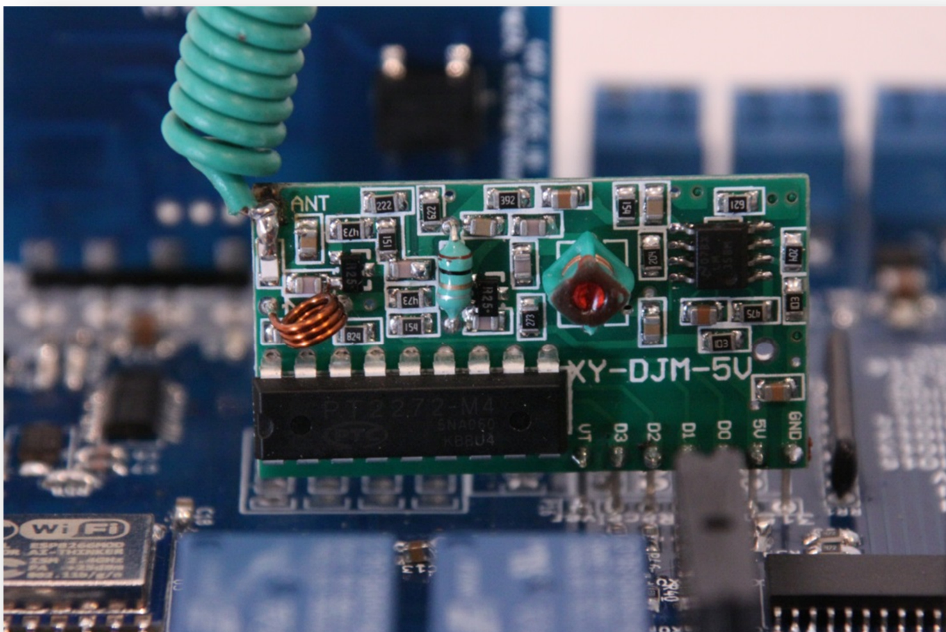


Figure 23 315mHz Receiver Module installed in P18

The 315 mHz receiver, antennas and other accessories, parts and modules may be purchased at www.IoThyngs.com.