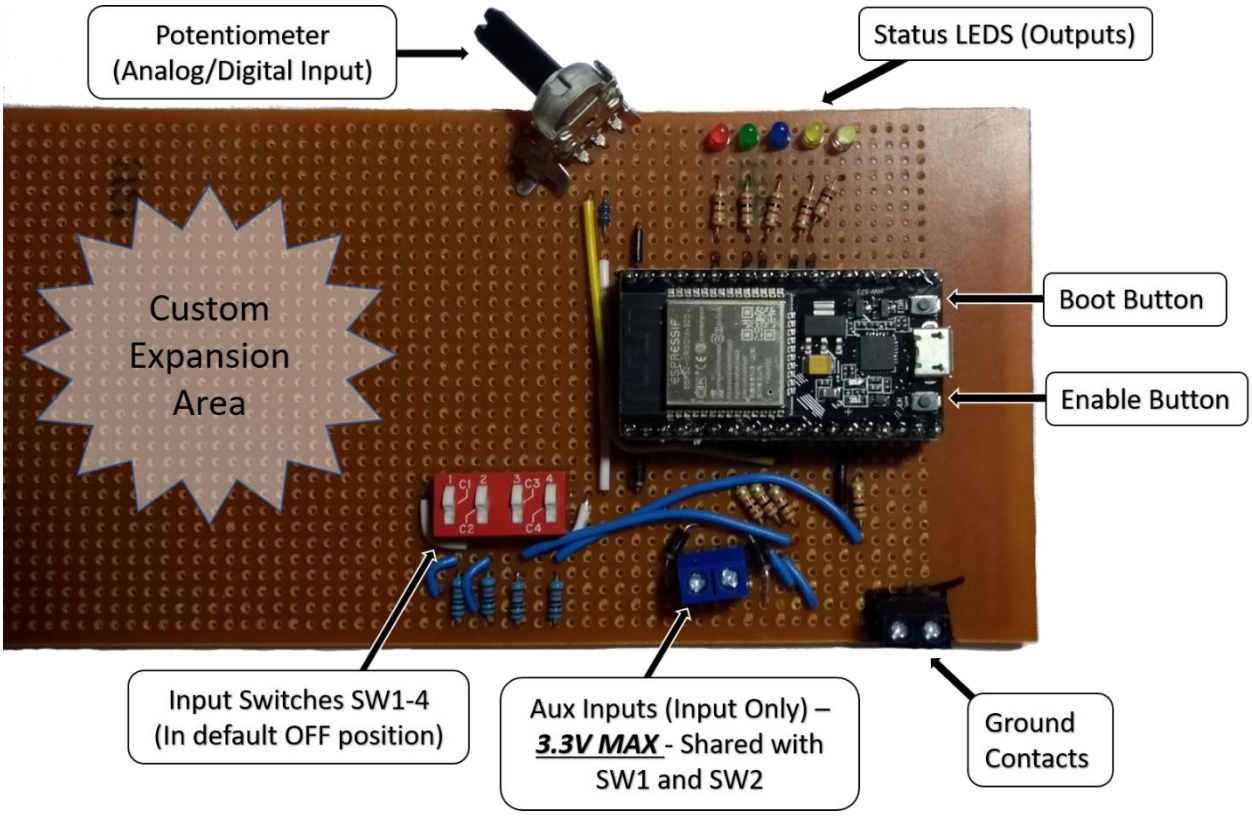
Title: Prototype Development Board

Author: Andrew Ward

Date: 3/14/2020

This document exists to give a brief overview of the ESPLC test boards that I have built for the group members. The following shows the features of the boards, the pinouts for each component, and lists a few “do” and “do nots”.



Shown above is the board, with all given inputs and outputs shown.

Components:

- ***Ground Contacts***
  - Serve as a way of easily connecting the grounds of auxiliary circuits that may be used for inputs/outputs.
- ***Inputs Switches***
  - Simple DIP switches that are compact, but still actuated by hand. Inputs are pulled low via 10k ohm resistors to prevent floating conditions.

- ***Potentiometer***
    - A simple 5K linear potentiometer, this is used primarily to test analog inputs.
        - In reality, a sensor would output a value between 0-3.3V (in our case), and that value would be translated to a number with 12-bit resolution (0-4095), and that number would then be translated into units that are useful. This is achieved via the logic program.
            - Example: Let's say an analog value of 0 represents a sensor monitoring a wheel turning at 500 RPM, but an analog value of 4095 represents a wheel RPM of 1000 or better. At some instance, the sensor gives a voltage of about 1.61V, which is about 2000 in 12-bit resolution. From this, some conversion from analog to RPM must occur in the program before we can perform logic operations with the data.
            - This is done with the following formula:
                - $(\frac{A_{IN}}{A_{HIGH}-A_{LOW}}) * (RPM_{HIGH} - RPM_{LOW}) + RPM_{LOW}$
                - → $(\frac{2000}{4095-0}) * (1000 - 500) + 500 \cong 744RPM$
                - Thus, we now have a value in RPM with which to make decisions with in a program, whether by comparison blocks or otherwise.

- ***Auxiliary Inputs***
    - Simple auxiliary inputs with diodes allowing the current to flow inwards to the input pins. The input pathways are shared with SW1 and SW2 respectively, so you must remember to set this switches OFF when using the auxiliary inputs. This was done to reduce the number of components that I needed to build each circuit. ***DO NOT EXCEED 3.3V WHEN INPUTTING TO THESE PINS; doing so will damage or destroy the ESP-32.***

- ***Status LEDs***
    - Simple, individually colored, low power LEDs, with 2.2k ohm current limiting resistors placed in series. I realize the green LED is not very bright, but it works well enough.
    - These outputs can easily be soldered into at the contact points before the resistors, but keep in mind that the pins from the ESP-32 are ***not designed to output more than 12mA of current***. Take these considerations into account if you play around with these.

- ***Boot Button***
  - When programming the ESP-32, you may need to hold this button down until the firmware flashing tool makes a connection to the device prior to uploading new firmware.

- ***Enable Button***
  - I'm honestly not 100% sure the enable button dues for these. It may function as a reset after flashing new firmware, but that's about all I can guess right now. When I find out, I'll update this documentation.

- ***Expansion Area*** (For the more motivated and enterprising among you)
  - I designed these boards to function well enough for the purposes of this project, but I also tried my best to leave plenty of area for those of you who would like to play with your own designs, and I would encourage you to do so.

Pinouts:

The figure and corresponding table can be used to reference the capabilities of each ESP-32 pin, and its viability as an input or output.

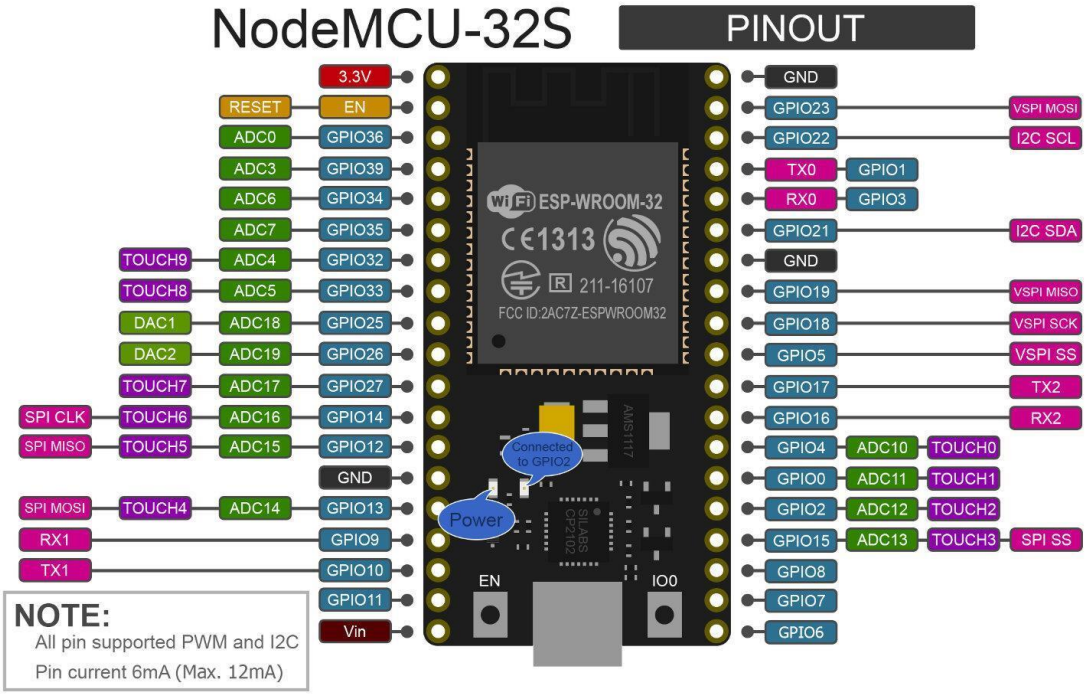


Figure 1 - Physical Pinout Diagram

GPIO	Input	Output	Notes
0	pulled up	OK	outputs PWM signal at boot
1	TX pin	OK	debug output at boot
2	OK	OK	connected to on-board LED
3	OK	RX pin	HIGH at boot
4	OK	OK	
5	OK	OK	outputs PWM signal at boot
6	x	x	Connected to the integrated SPI flash
7	x	x	Connected to the integrated SPI flash
8	x	x	Connected to the integrated SPI flash
9	x	x	Connected to the integrated SPI flash
10	x	x	Connected to the integrated SPI flash
11	x	x	Connected to the integrated SPI flash
12	OK	OK	boot fail if pulled high
13	OK	OK	
14	OK	OK	outputs PWM signal at boot
15	OK	OK	outputs PWM signal at boot
16	OK	OK	
17	OK	OK	
18	OK	OK	
19	OK	OK	
21	OK	OK	
22	OK	OK	
23	OK	OK	
25	OK	OK	
26	OK	OK	
27	OK	OK	
32	OK	OK	
33	OK	OK	
34	OK		input only
35	OK		input only
36	OK		input only
39	OK		input only

Table 1 - Pin Compatibility Chart

DEVICE	GPIO PIN (ESP-32)	STATE
WHITE LED	16	Active High
YELLOW LED	17	Active High
BLUE LED	18	Active High
GREEN LED	19	Active High
RED LED	21	Active High
POTENTIOMETER	26	Analog IN
SWITCH 1	32	Digital IN
SWITCH 2	33	Digital IN
SWITCH 3	25	Digital IN
SWITCH 4	13	Digital IN
AUX IN 1	32*	Digital/Analog IN
AUX IN 2	33*	Digital/Analog IN

Table 2 - Prototype Board Pins and Device States

* = Corresponding shared switch must be OFF for AUX inputs to function, otherwise input is always HIGH.

References:

- Table 1 compiled from data located here:
  - https://randomnerdtutorials.com/esp32-pinout-reference-gpios/
- Figure 1 was taken from this site:
  - http://modtronix.com.au/product/nodemcu-32s/