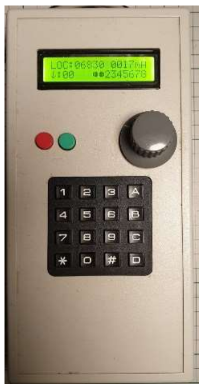
ESP DCC Controller

ESP8266 based DCC model railroad controller

DISCLAIMER: Build at your own risk. Every care has been taken in the design to handle short-circuits and over-voltages to avoid potential damage to your loco decoders, however I cannot control your build quality. I accept no responsibility for damage/loss on your part.

There are various build options. First choose a power option. Next you need to choose a display-keyboard DSKY option (or none for mobile/laptop/tablet control only). It is recommended you use the system PCB designed to integrate all the components.

Component	Value	Purpose
R3	13k	Jogwheel (rotary encoder) support. Pushbutton on encoder is connected across pins 1+2. Pin 4 is the center of the encoder and 3,5 are the two encoder outputs.
R4	Not used	
R5	1k	
R7	1k	
JP5	Header	
D1	1N914	Negative voltage charge pump. Not required if you run the LCD display at 5v. D1/D2 any small signal diode is ok.
D2	1N914	
C5	4.7uF 16v	
C2	0.1uF	
D3	1N914	D3/4 are I2C bus clamping diodes. Any small signal diode is ok. JP4 is the I2C bus itself and there is a 3-pin header with jumper to select 3v3 or 5v operation. R1/2 are bus pull-ups. Not required because the INA module has on-board bus pull-ups.
D4	1N914	
JP4	Header	
R1	4k7	
R2	4k7	
3.3/5V selector	Header	
IC1	7805	7805 regulator or equivalent. Required for I2C bus power and to drop supply to 5V for the nodeMCU.
JP1	Wire link	
C7	0.1uF	
IC2	LMD18200T	On-board LMD H bridge. C1 should ideally be a low ESR electrolytic. C6 should be a good quality 1uF non polarised cap. R6 is the sense resistor but is not required. If using any of the off-board H bridge options, R6 is a wire link.
R6	680 ohm*	
C1	470uF 25v	
C3	0.01uF	
C4	0.01uF	
C6	1uF non p	
JP6	Wire link	

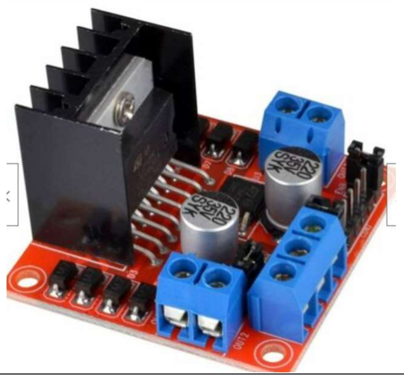


Complete unit with on-board LMD18200 and DSKY option built into a Hammond case. It’s a squeeze inside. Red button is emergency stop, green is mode.

The 12V DC regulated power supply is external, and is an ex -laptop supply.

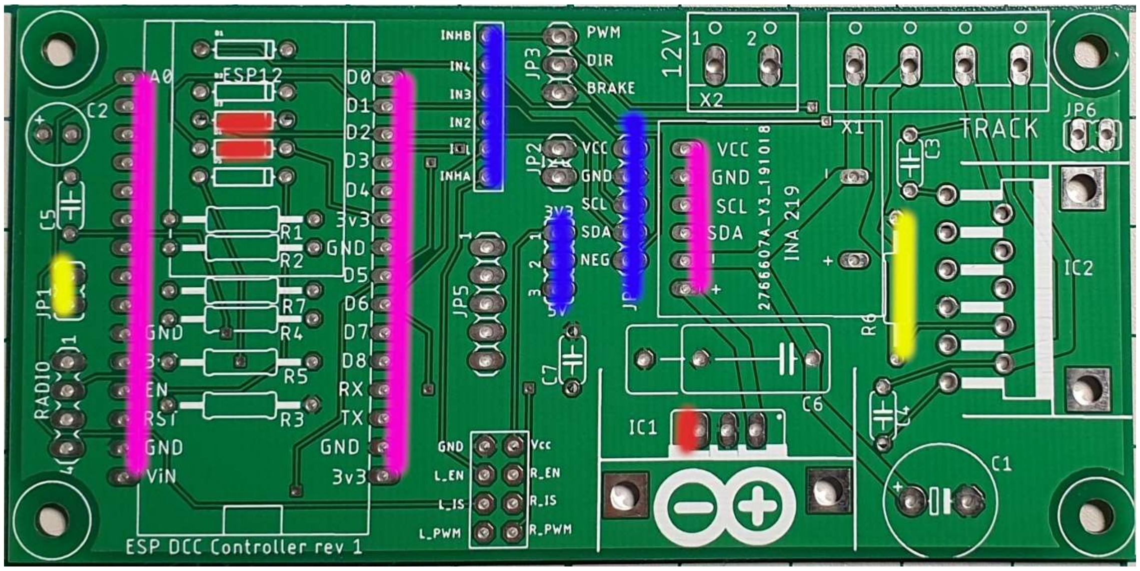
ESP DCC Controller

External L298 power option.

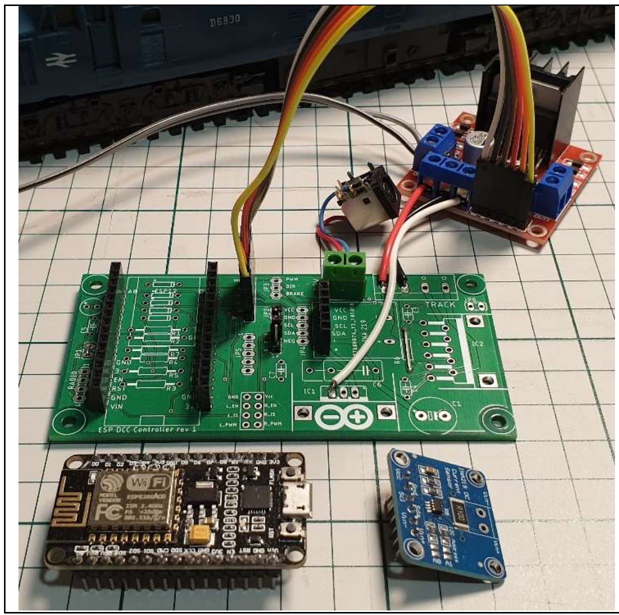
	Uses an external L298 module. These are available on Ebay. They also have an on-board 5v regulator which can be used to power the nodeMCU and DSKY option you chose, if any, thus avoiding need for IC1 + C7. This build is also known as the bare-bones build because it uses the absolute minimum of parts. Bonus, it also comes with 2 pin jumpers which need to be removed. These can be used on the main board jumper positions.

On the system PCB, you need to instal;

- wire link in JP1 and a wire link across R6. Shown yellow.
- Female PCB headers for the nodeMCU and INA 219. Pink.
- Install male PCB headers for the L298 control, the I2C bus power selector, and the I2C bus itself (if using DSKY) shown blue.
- Fit a screw terminal block in X2 for the incoming 12V DC.
- X1 leftmost positions are soldered wires going off to the L298 and 5V is picked up from the L298 and fed into the IC1 regulator position (red blob). A regulator does not need to be fitted on the main PCB.
- If you don't intend to add any DSKY devices and wish to control the unit through a mobile and laptop only, then select 3v3 for the I2C bus and you can avoid the need for clamping diodes D3/4. (shown red)



ESP DCC Controller



Here's the built circuit board prior to fitting the INA219 and node MCU into their sockets.

The red wire from X1 is 12V toward the L298, black is power ground for the L298 and white is 5V from the L298 board to the motherboard and will supply the node MCU and INA. Grey wires to the track.

Note the I2C bus power jumper is in the 3v3 position and no clamping diodes are fitted. Careful. Do not jumper to 5v or you risk damaging the nodeMCU

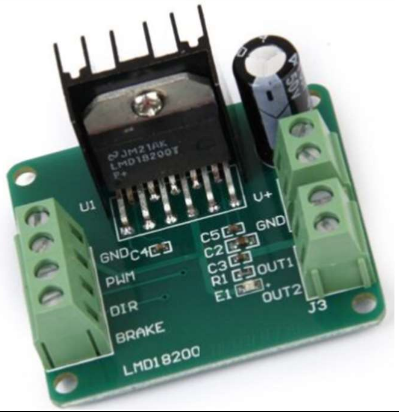
A Dell power socket with red/blue wires is the power supply and the grey wires go to the track.

This is all that is required for the bare-bones build. I would recommend fitting D3/4 clamping diodes to protect the nodeMCU in case you jumper the I2C bus to 5V supply by mistake. The INA has on-board I2C bus pull ups, which saves the need to fit R1/R2.

I also recommend a hardware emergency stop button. This is a pushbutton wired across JP2. The FLASH button next to the USB socket on the node MCU also acts as emergency stop.

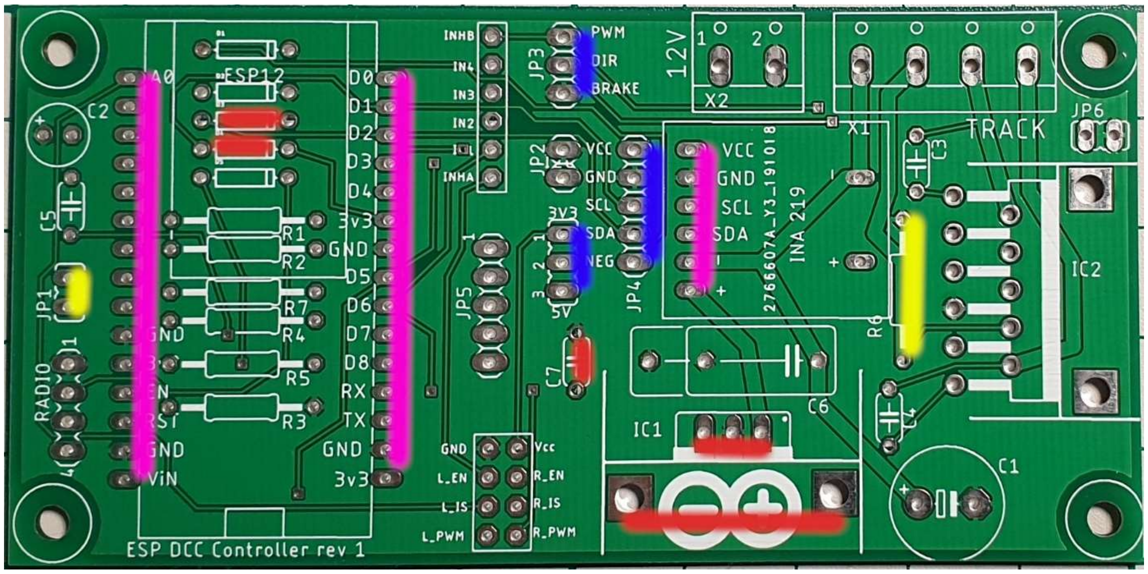
ESP DCC Controller

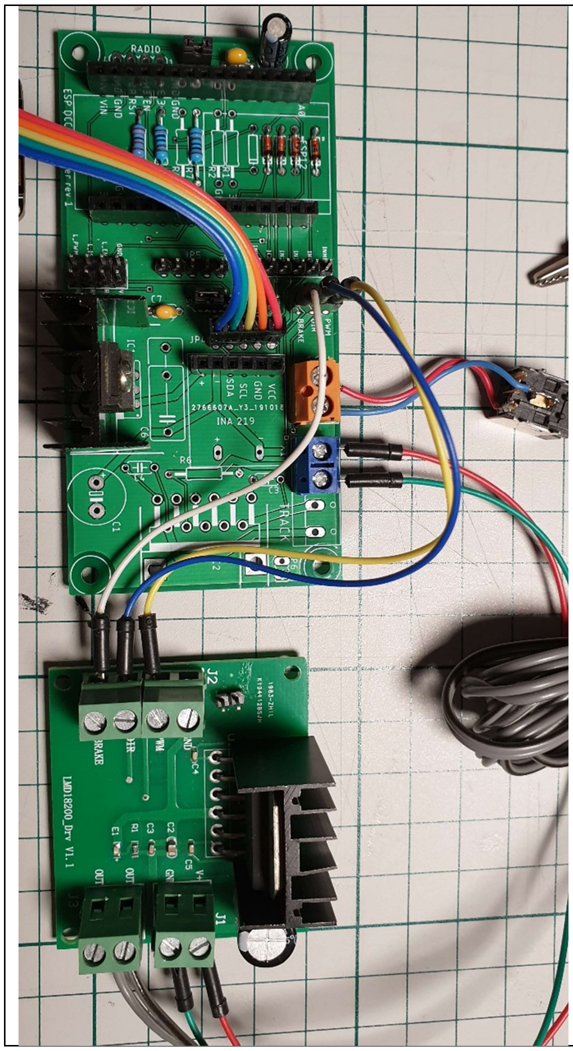
External LMD18200 option

	Uses an external LMD18200T module, available on eBay. The LMD18200 supports 3A, the LMD18200T supports 4A. With this build, you need to fit a 5V regulator to the system board.

On the system PCB instal;

- wire link in JP1 and a wire link across R6. Shown yellow.
- Female PCB headers for the nodeMCU and INA 219. Pink.
- Install male PCB headers for the LMD18200 control, the I2C bus power selector, and the I2C bus itself (if using DSKY) shown blue.
- Fit a screw terminal block in X2 for the incoming 12V DC.
- Fit a screw terminal block in X1 leftmost 2 holes. This is power/ground for the LMD.
- If you don't intend to add any DSKY devices and wish to control the unit through a mobile and laptop only, then select 3v3 for the I2C bus and you can avoid the need for clamping diodes D3/4. (shown red)
- Fit 5v regulator IC1 with heatsink and C7 0.1uF bypass capacitor. Shown red.





Built up PCB. The orange block is 12v/gnd power in, the blue block is 12v/gnd to the LMD module. The grey wires are from the LMD to the track. <u>The R6 wire link is missing in this photo, it should be installed</u>.

INA and nodeMCU yet to be fitted to their sockets. The ribbon cable is the I2C bus for DSKY.

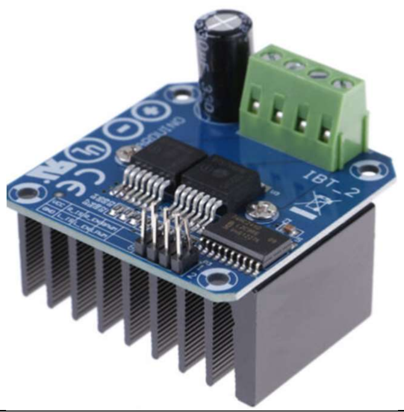
You can see the LMD control lines for pwm/dir/brake linked up.

The I2C power is set to 5v (for the DSKY) and the two I2C clamp diodes D3/4 are present. If not using the DSKY, you can set the I2C power to 3v3.

Also present but optional and related to the DSKY are;
D1/2 and C2/C5 which are the negative voltage charge pump (see DSKY options).

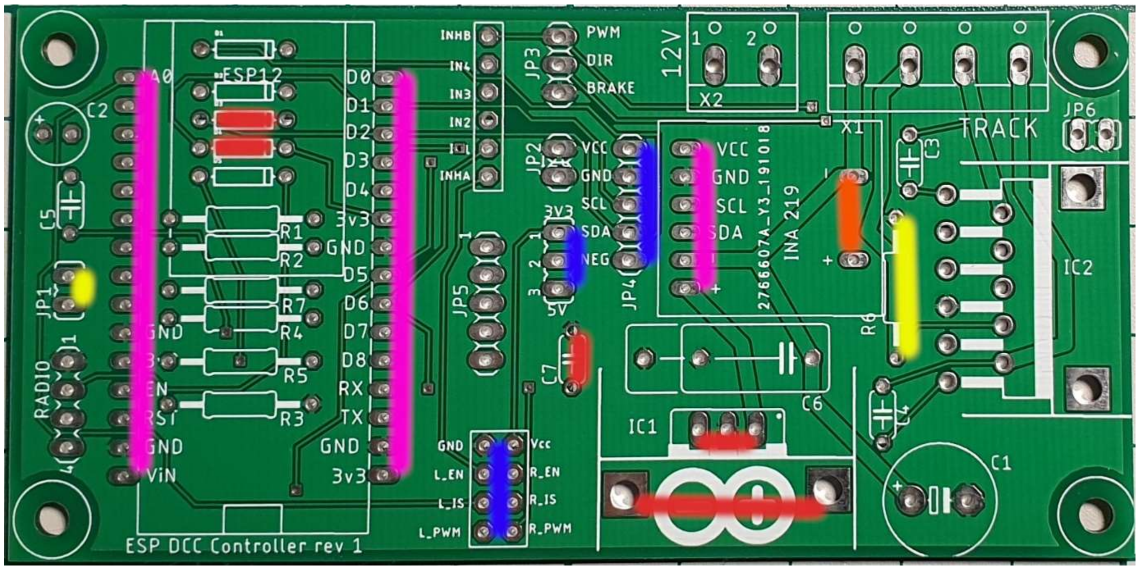
Resistor R3/5/7. These are for the Jogwheel option.

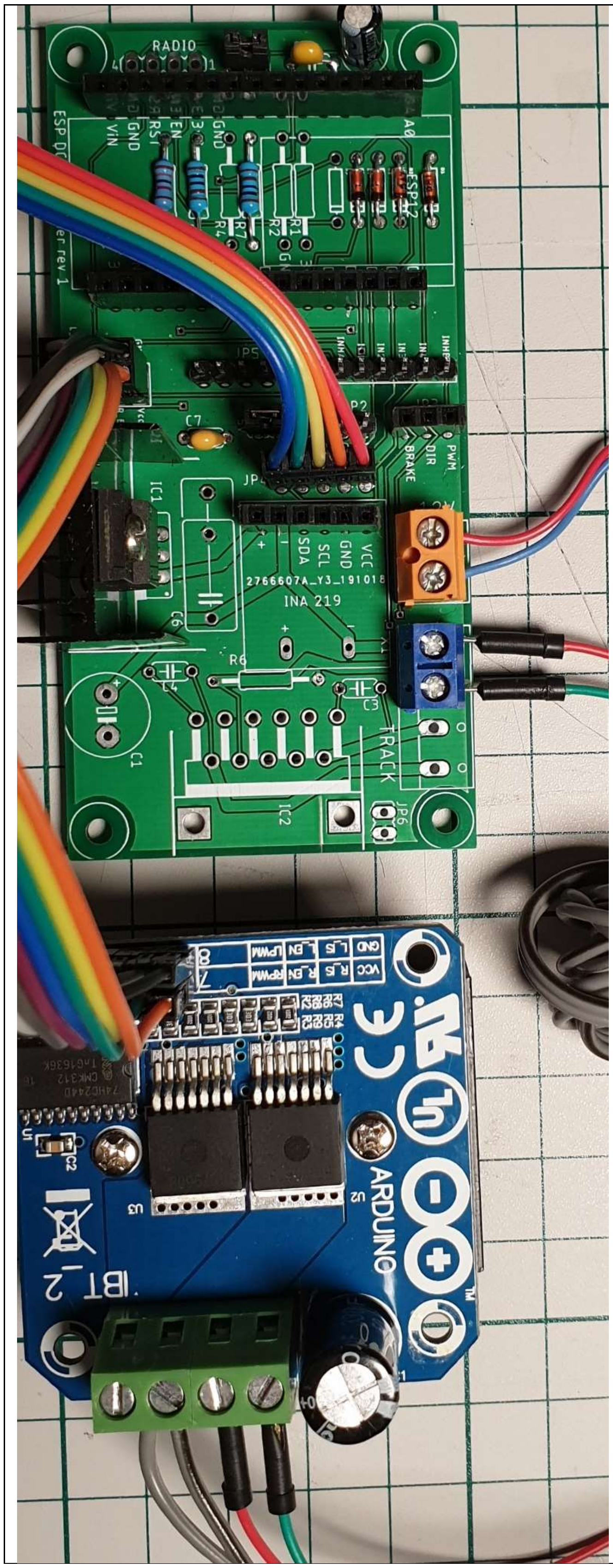
ESP DCC Controller

	IBT2 board, incorporates two BTS7960B Half-H bridge chips. The board can handle 20Amps+ The System PCB and INA are designed to handle a max 4A. It would be possible to add a shunt resistor in the current measurement circuit to allow the unit to operate at 8A. If you want this much power, then it makes sense to use an IBT2. Otherwise stick with the LMD.

On the system PCB instal;

- wire link in JP1 and a wire link across R6. Shown yellow.
- Female PCB headers for the nodeMCU and INA 219. Pink.
- Install male PCB headers for the IBT2 control, the I2C bus power selector, and the I2C bus itself (if using a DSKY) shown blue. You do not need to connect the L_IS and R_IS lines.
- Fit a screw terminal block in X2 for the incoming 12V DC.
- Fit a screw terminal block in X1 leftmost 2 holes. This is power/ground for the LMD.
- If you don't intend to add a DSKY and only control the unit through a mobile and laptop, then select 3v3 for the I2C bus and you can avoid the need for clamping diodes D3/4. (shown red)
- Fit 5v regulator IC1 with heatsink and C7 0.1uF bypass capacitor. Shown red.
- Optionally, you can fit a 0.1R shunt across the INA, shown orange. This needs to be a 2W resistor. You will need to modify the CURRENT_SCALING value in the software. This will support 8A. CAUTION: Things (your locomotives, track) can easily melt because the unit will not trip even if 7 Amps are flowing into a short circuit. Your DC supply must also be capable of pushing out >8A.





Built up PCB. The orange block is 12v-gnd power in, the blue block is 12v-gnd to the LMD. The grey wires are from the LMD to the track. The R6 wire link is missing in this photo, it should be present.

INA and nodeMCU yet to be fitted to their sockets. The ribbon cable is the I2C bus (for DSKY).

You can see the IBT2 control lines hooked up via a ribbon cable on left. Note you do not need to fit the IS lines (current sense) as they are not used.

The I2C power is set to 5v (for the DSKY) and the two I2C clamp diodes D3/4 are present. If not using the DSKY, you can set the I2C power to 3v3.

Also present but optional and related to the DSKY are;
D1/2 and C2/C5 which are the negative voltage charge pump (see DSKY options).

Resistor R3/5/7. These are for the Jogwheel option.

ERRATA: the system PCB (rev 1) has a mistake. L_EN and R_EN should be connected together. They are not, so solder a wire between them.

L_IS and R_IS should not be connected. You only need a 6-wire ribbon over to the IBT2.

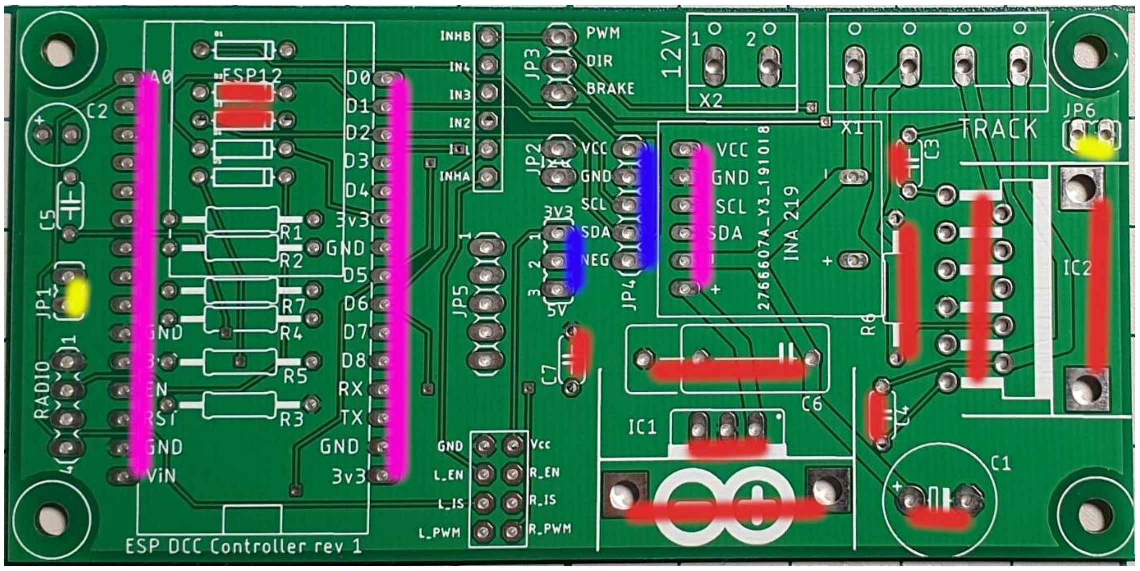
Pay attention to the sequence, as the PCB has EN and IS out of sequence compared to the IBT2 board. Oops.

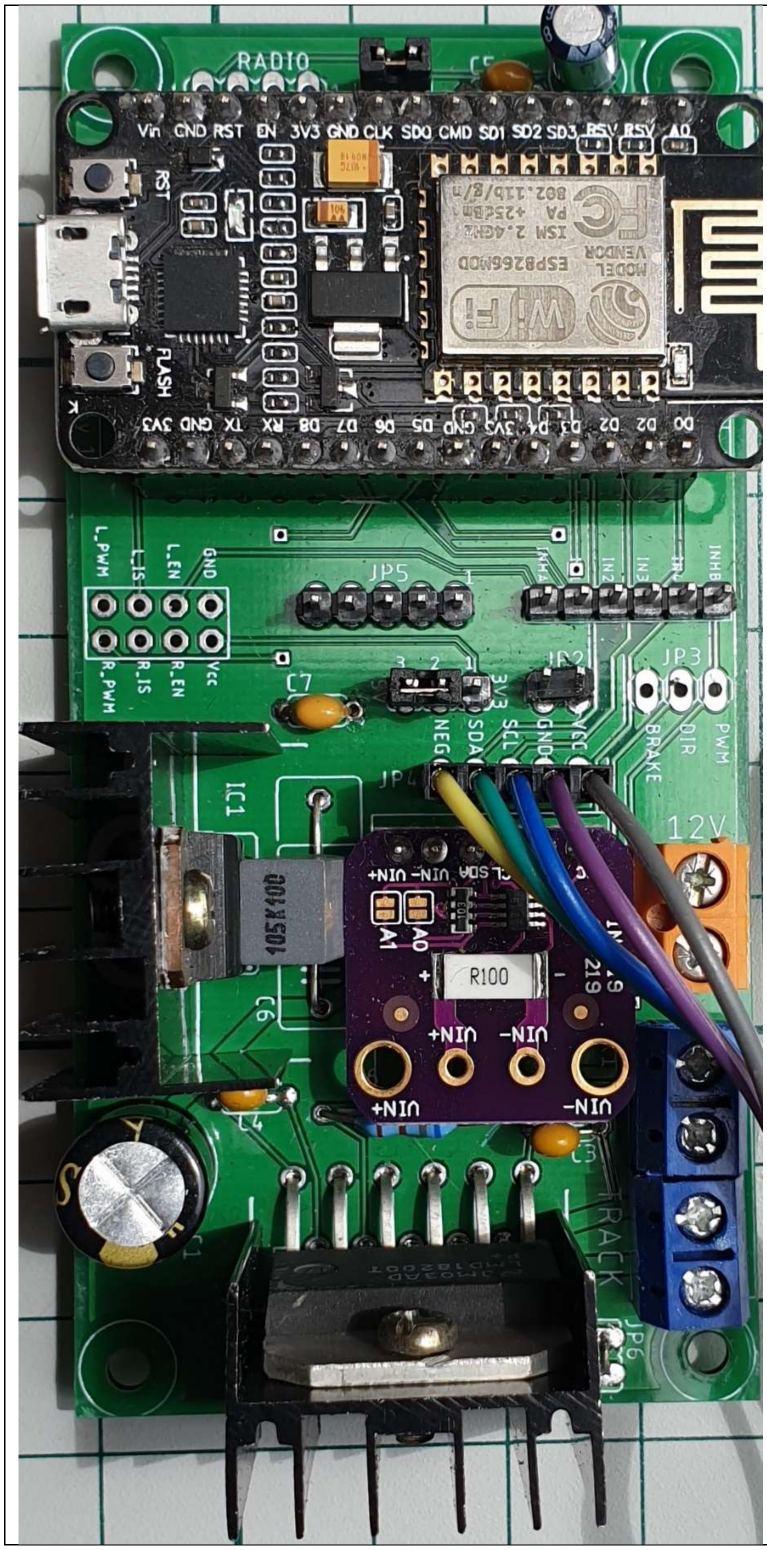
Fully integrated LMD18200 option

This option builds an LMD18200 onto the system board along with its supporting discrete components. It is the most compact of the build options, but does require you source a discrete LMD18200 and heatsink. The heatsink can be found on Ebay as an "L298 heatsink."

On the system PCB instal;

- wire link in JP1 and also JP6. Shown yellow.
- Fit R6, or omit entirely. Shown red. This is a current sense resistor, but we are using the INA.
- Female PCB headers for the nodeMCU and INA 219. Pink.
- Install male PCB headers for the I2C bus power selector, and the I2C bus itself (if using a DSKY) shown blue.
- Fit a screw terminal block in X2 for the incoming 12V DC.
- Fit a screw terminal block in X1 rightmost 2 holes. This is power to the track. The left most 2 holes of X1 are left blank.
- If you don't intend to add a DSKY and so control the unit through a mobile and laptop only, then select 3v3 for the I2C bus and you can avoid the need for clamping diodes D3/4. (shown red)
- Fit 5v regulator IC1 with heatsink and C7 0.1uF bypass capacitor. Shown red.
- Fit C1, C3, C4 and the LMD18200T IC2 with heatsink.





Built up PCB. The orange block is 12v-gnd power in. The lower blue block is power to the track. The upper blue block is unused in this configuration.

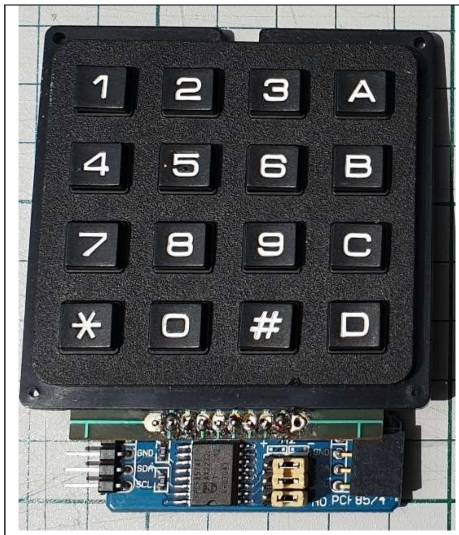
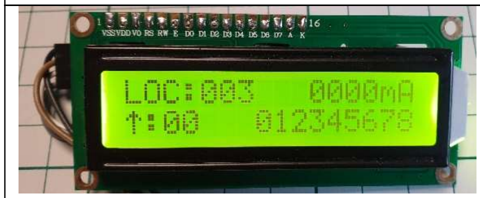
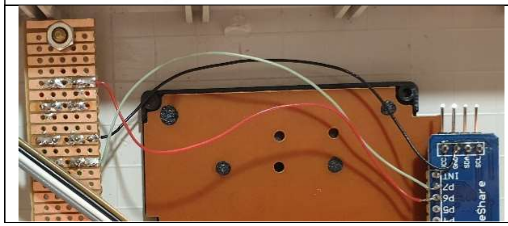
R6 is fitted as a 680ohm resistor. C1/3/4/6 and the LMD are fitted.

<u>Note the wire link for JP6.</u> JP1 can also be a wire link.

The ribbon cable is the I2C bus (for DSKY).

The I2C power is set to 5v (for the DSKY) and the two I2C clamp diodes D3/4 are present but hidden beneath the nodeMCU.

DSKY. Display, Keyboard and Jogwheel options

	The keypad is a 4x4 telephone matrix. You need to solder on a PCF8574 I2C GPIO expander board (Ebay) to this, the left most INT pin is not required, but there is a convenient hole for it on the keypad PCB. The next 8 pins (P0-P7) pick up the columns and rows of the keypad. Buy an expander with a plug one side and socket the other, this allows you to daisy chain the I2C bus onward to the 1602 display. The board comes as PCF8574 or PCF8574A which has a different address range. Read the chip ID. Either will work. They have jumpers A0-2 to set the address. Terrible soldering in this pic. These keypads have a cheap paper/resin PCB and its too easy to cook the pads off. Clean the pads with isopropyl, use a flux, solder on and clean afterward with isopropyl.
	LCD1602 display. These are available with a I2C backpack already fitted. These are based on the PCF8574(A) and there are several types which map their pins differently. The software needs to be configured at compile time to match your board. These boards usually run off 5V.
	Jogwheel, also known as a rotary encoder. You need the vanilla 5 terminal type, where 2 terminals are the pushbutton, the other 3 are the encoded pulses. They have a thread + nut to secure them to a panel. You need to find a knob to suit. Remember to space the knob off the panel slightly so you can push down to activate the button. IMHO this is by far the best way to control your loco, especially for shunting.
	If using the DSKY, the red emergency stop button is wired to P6 of the GPIO expander. The mode button is wired to P7. These buttons pull P6 or P7 down to ground (black wire) when pressed and are picked up by the GPIO scanner.

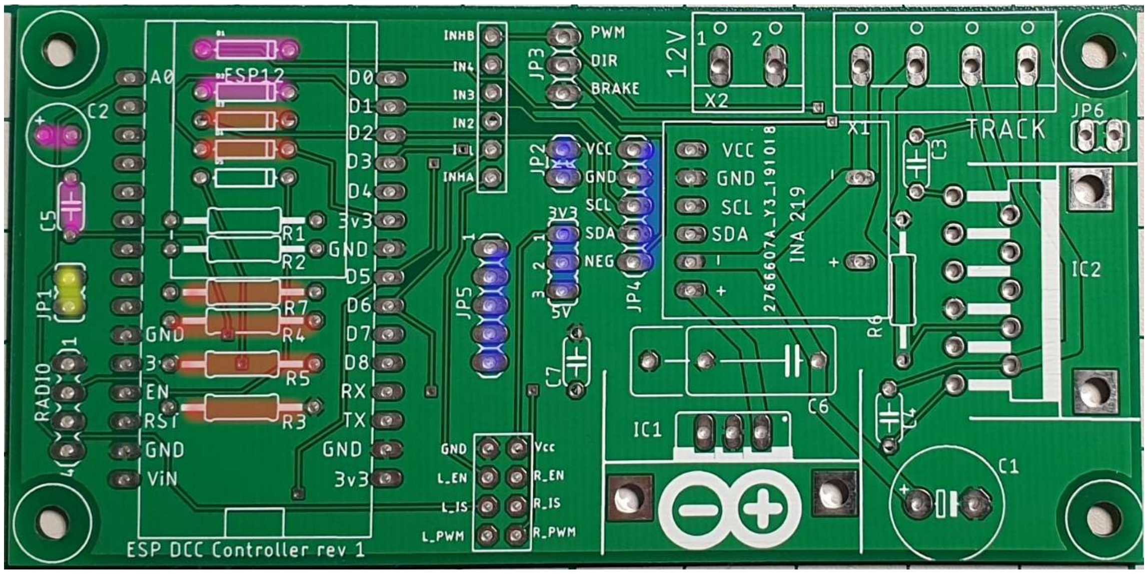
Whichever power option you chose, you have the same DSKY options. For bare-bones systems, control is via mobile phone, however it is also useful to have a hardware emergency stop button on the unit. This can be wired across JP2 shown blue below.

ESP DCC Controller

The I2C bus JP4 is required for the DSKY, but not the NEG pin. (See 3v3 display, charge pump).

The I2C bus voltage jumper should be set to 5v. Clamping diodes D3/4 (red) are required. The INA and LCD1602 are 5v tolerant, but the nodeMCU is not. The clamping diodes do not allow the I2C voltage on the SCL and SDA lines to rise above 3v3 and so protect the nodeMCU inputs.

3v3 display with charge pump: D1/2, C2/5 are a negative voltage charge pump (pink). This generates -3v3 on the NEG pin of JP4. If you want to run the I2C bus at 3v3 including the 1602 LCD display, then fit these components. To run a 1602 on 3v3 can be done but it will result in a lower brightness LED backlight. You also need to disconnect the gnd side of the contrast pot, and bring the NEG voltage to this pin. The LCD requires a bias voltage 4V below supply voltage in order for the contrast to work and the display characters to actually be visible. This is easy with a 5v supply, you have potentiometer across 5V and gnd. If the supply voltage is reduced to 3v3, the bias must now be -1v. You need to unsolder the ground end of the pot and hook it up to the NEG output. This allows a negative bias and the display will be visible. It works but is not worth the bother unless you also intend to put some other I2C devices on the bus (and write code for them) that are not 5V tolerant.



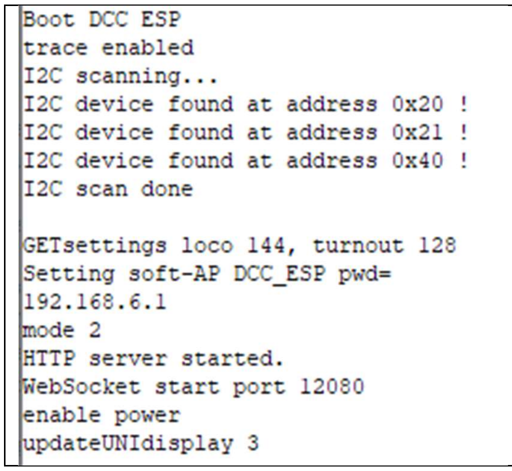
JP5, and R3/5/7 relate to the jogwheel. R4 is not required.

Integrate and test the I2C devices.

First the software Global.h needs to be configured to match the I2C addresses you have for your 3 devices. The INA typically comes with a default address of 0x40. The PFC on the 1602 may be the same device type as the one on the GPIO, so you will need to set the jumpers on the GPIO board (keypad) so there is no address clash.

After you compile and upload to the nodeMCU, you can test.

If you connect to the nodeMCU with a USB cable and monitor this through the Arduino IDE serial monitor port, when you press the RST button it will force the nodeMCU to re-boot and one of the first routines will scan the I2C bus and declare all devices found. You expect to see 3 on separate addresses.



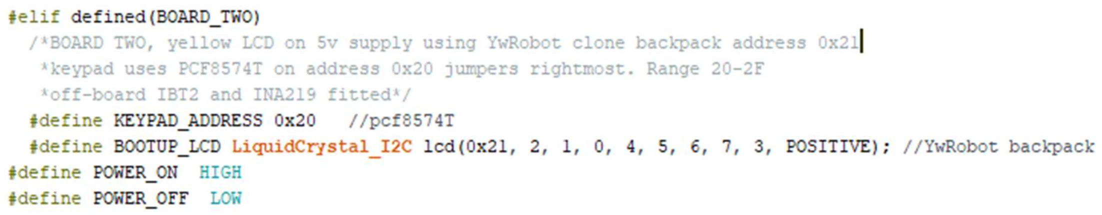
```
Boot DCC ESP
trace enabled
I2C scanning...
I2C device found at address 0x20 !
I2C device found at address 0x21 !
I2C device found at address 0x40 !
I2C scan done

GETsettings loco 144, turnout 128
Setting soft-AP DCC_ESP pwd=
192.168.6.1
mode 2
HTTP server started.
WebSocket start port 12080
enable power
updateUNIdisplay 3
```

Here the boot sequence has found 3 different I2C devices on separate addresses. If you have an address clash, e.g. two devices on 0x20, then you will only see 0x40 and 0x20 displayed.

Change the address jumpers, reboot and see what you now have.

You need to set Global.h to match your addresses and the backpack configuration on your 1602 LCD.

Example Global.h entry



```
#elif defined(BOARD_TWO)
  /*BOARD TWO, yellow LCD on 5v supply using YwRobot clone backpack address 0x21
   *keypad uses PCF8574T on address 0x20 jumpers rightmost. Range 20-2F
   *off-board IBT2 and INA219 fitted*/
  #define KEYPAD_ADDRESS 0x20   //pcf8574T
  #define BOOTUP_LCD LiquidCrystal_I2C lcd(0x21, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //YwRobot backpack
#define POWER_ON  HIGH
#define POWER_OFF  LOW
```

Note that the software defaults to assuming the INA is on 0x40

For the jogwheel, this is independent of the I2C bus, but it can only be used with the DSKY because it is assigned to the last loco selected in the DSKY. Clockwise rotation should result in a speed increase, counter clockwise in a speed decrease. If this is the wrong way around, you can swap the pin positions in the header cable, or hack the software.

If you are in shunter-mode, a CCW rotation will decrease speed in the forward direction, through zero and continual CCW rotation will then start to increase speed but in the opposite direction.