MQTT Turnout Controller Instructions

Description

The MQTT Turnout Controller is designed to provide everything required to operate a turnout, whether it's operated by servo or if it's a snap switch, such as Atlas or Kato type. Its small size makes it easier to hide under the track.

- A 3.3v servo output on a 3-pin header. Speed can be adjusted in the sketch.
- Two 3.3v input/output pins available on a 3-pin header.
- A red/green LED output available on a 2-pin header.
- Two 3.3v closed and thrown outputs available on a 3-pin header.
- ESP8266-based Wemos D1 Mini providing WiFi connection to JMRI.
- User-programmable addresses for inputs and outputs.
- Can be powered by 5-12VDC. The higher the input voltage, the more power is wasted in heat. Above 9V should be avoided if possible.

Assembly

Surface Mount Parts

The surface mount parts need to be soldered onto the board before the through-hole parts since the top surface needs to be laid flat on the "cookie sheet". My cookie sheet is a piece of 1/16" thick aluminum big enough to cover the hot plate, with a wooden handle attached. My hot plate is actually a heater section from a semiconductor wafer oven with a digital controller.

Set the hot plate to 250°C. Having the hot plate too hot will cause some of the parts to dance around when the flux boils. Check the profile included with the solder paste for details.

The surface mount parts chosen for this project are large enough to allow for manual solder paste application and part placement, even if you're not an expert. The only exception is the driver chip on the shield.

Apply solder paste to all the pads, then place each part, being careful to observe the orientation of the L293DD. If you only want to use a servo, you don't need the L293DD driver chip.

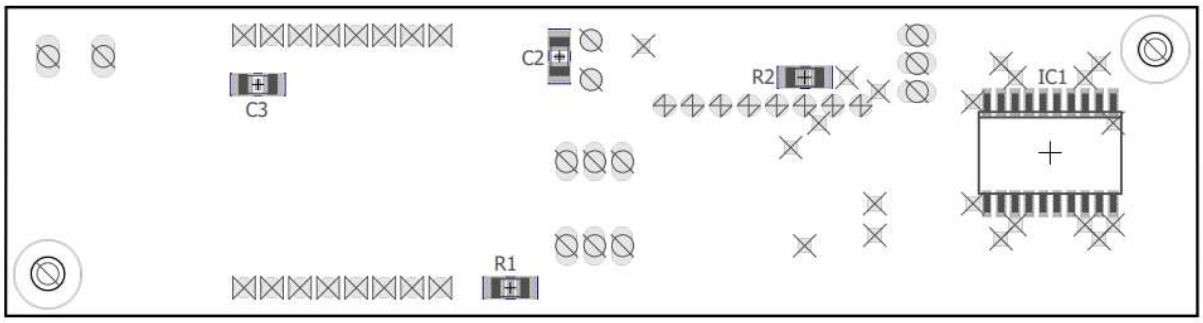

Figure 1: Surface Mount Parts Placement

Table 1: Surface Mount Parts Needed

Qty	Part	Value	DigiKey #	Device
2	C2,3	.1uF	399-C1206C104K5RAC7800CT-ND	C-USC1206
1	IC2	L293DD	497-1390-5-ND	L293DD
1	R1	182K 1%	311-182KFRCT-ND	R-US_M1206
1	R2	2K	311-2.0KERCT-ND	R-US_M1206

Place the board(s) on the cookie sheet then place the cookie sheet on the hot plate. Watch the boards carefully so you can nudge parts back into place if they move. Carefully remove and let cool when everything flows nicely.

After the board(s) cool, use a magnifying glass to check for bridging between the pins of the driver chip (if you have a driver chip) and if any is found, remove it with solder wick or a fine point soldering iron. Also check for parts that may be missing solder.

Through-Hole Parts

Prepare the D1 Mini first. Hopefully you bought your minis with the headers included. Solder the male headers onto the D1 Mini, tacking only one pin at first to make sure the header is perpendicular to the board, then solder the rest of the pins.

Figure 2: Mini with male headers installed

To ensure good alignment of the short female headers on the circuit board, install them onto the pins of the Mini, then insert the headers and Mini into the top of the block controller board.

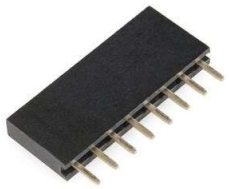

Figure 3: Short female header.

Solder the remaining pins of the headers.

Remove the Mini and install the remaining through-hole parts. Make sure to observe the polarity of C1.

Table 2: Through-Hole Parts Needed

Qty	Part	Value	DigiKey #	Device
1	C1	100uF	493-1107-ND	CPOL-USE3.5-8
1	TS1	8 pos term block	ED10566-ND	TERMBLK05
3	SV1,2,3	3 pos male header	Snap from S1011EC-40-ND	MA03-1
1	U$1	WEMOS-D1-MINI	(Amazon or Ebay)	See Instructions
1	X1	2 pos term block	277-12547-ND	W237-102

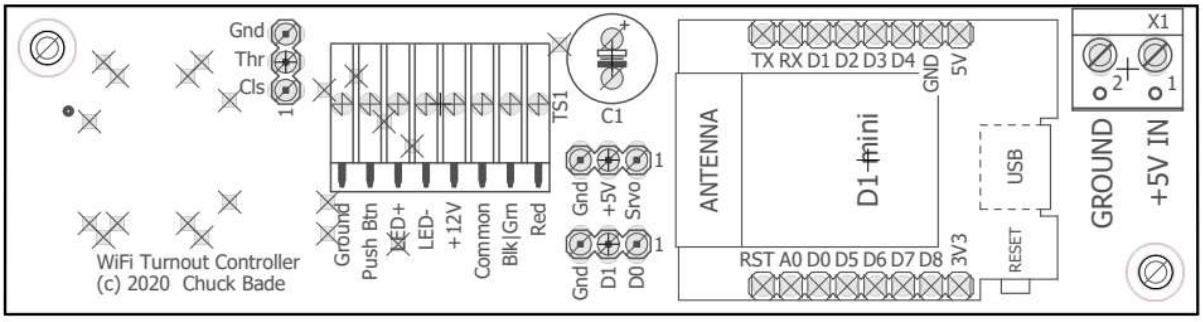

Figure 4: Through-Hole Parts Placement

Testing

1. Program a mini and plug it in to the turnout controller. See the Programming section below.
2. Apply power to X1.
3. The easiest way to test the module is to put a combination red/green LED in TS1 in the positions marked LED+ and LED-. If you don’t have one, you can connect two different colored LEDs in parallel but in opposite polarity to each other.
4. Solder up a testing push button with two wires or a two pin male header.
5. Insert the push button into TS1 between Ground and Push Btn.
6. Push the button. The LED should change.
7. If you’re going to use a servo, plug it into the 3-pin header in the middle of the board. Make sure the black wire goes to the pin marked Gnd, red to +5V, and white to Srvo.
8. Push the button. The servo should cycle. If it doesn’t on the first push, it may already in the commanded position; just hit it again.

Programming

Numbering Inputs and Outputs

The numbers for inputs in JMRI are independent of the output numbers. They could have the same numbers as inputs or completely separate. I prefer to have my turnouts at low numbers, leave enough room for expansion, then have my signal light outputs at some higher range. The reason for this is that internally JMRI treats all outputs as turnouts, and when you want to throw a turnout with WiThrottle, having the turnouts first in the list will help you avoid doing a lot of scrolling to get to your turnouts.

A note about turntables and staging yard transfer tables. Each ray in a turntable is treated as a turnout in JMRI, and "throwing" the turnout sends a command to move the table to that ray. You need to reserve enough turnout numbers for however many rays you have. Again, these need to be lower numbers so you can select the rays with WiThrottle.

Since lights are generally set automatically, the numbers for them can be set to high values, out of the way of outputs that need more frequent access.

Each light in a signal head is one output, so remember to leave space. I have an extra output on my block controller, called "aux", so I leave an extra number for that. Since the D1 Mini that is on the block controller has up to 6 outputs, it wouldn't hurt to leave space for 6 total. For example, my first light (green) on my first block controller is at 800, and the green light on the second block controller is at 806.

So my output numbering scheme is a follows:

- 1-99 Turnouts
- 100-199 Turntables/Transfer Tables
- 800-805 1st Block (green, yellow, red, aux, future1 future2)
- 806-811 2nd Block
  …
- 944-949 25th Block

Inputs can have the same numbers as outputs, but to avoid confusion, I've chosen a different series of numbers.

- 400-499 Turnout Feedback Sensors
- 500-599 Block Occupancy Detectors
- 500 1st Block Detector
- 501 (future use, unused input on block controller)
- 502 2nd Block Detector
- 503 (future use)

Setting up Arduino IDE

1. Install Arduino IDE.
2. In Arduino IDE, go to File->Preferences and enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into the “Additional Boards Manager URLs” field.
3. Go to Tools->Board->Boards Manager…, search for ESP8266 and press the install button for the “ESP8266 by ESP8266 Community“.

You will probably need the following libraries:

- Servo by Michael Margolis
- Stepper by Arduino
- WiFi by Arduino
- PubSubClient by Nick O'Leary
- Ethernet by various

D1 Mini Configuration

1. Go to https://github.com/chuckbade/MQTT
2. Download the latest ino file from the MQTT Turnout Controller directory.
3. Open it in Arduino IDE.
4. Change the **Ssid** and **Pswd** values near the top of the sketch to match your network.
5. Change **MQTTServer** to the IP address of your MQTT broker. You will need to set up an MQTT broker somewhere on your network and have it start automatically.
6. Change the **JMRITurnoutNumber** and the **JRMISensorNumber** for the turnout being configured.
7. The **ServoDelay** variable slows the servo down for more realistic operation. Set it to 0 if no servo is being used.
8. Change **RESTORE_LAST_STATE** to **true** if you want the controller to remember the last state upon startup.
9. Connect the Mini to the USB cable and compile the sketch.
10. After it says it is resetting the device, disconnect the Mini and connect it to the turnout controller. **Be very careful installing the Mini on the board, making sure it is properly oriented and aligned with the correct pins. Mini's will self-destruct if off by just one pin.**

JMRI Configuration

Adding Turnouts

NOTE: You must have JMRI version **4.21.1** or newer for MQTT to work with sensors.

1. From the main PanelPro menu, go to **Tools → Tables → Sensors**.
2. Click **Add…**. The **Add New Sensor** dialog will appear.
3. If **System Connection** is not showing **MQTT**, select it in the pull-down. If MQTT is not listed in the pull-down, then the installed version of JMRI is too old. See the note above.
4. Enter your desired sensor number in the **Hardware Address** field.
5. Enter a name for the turnout sensor in the **User Name** field.
6. Click **Create**. The new sensor should now be listed in the sensor table.
7. In the list on the left side of the window, select **Turnouts**.

8. Click **Add…**. The **Add New Turnout** dialog will appear.
9. If **System Connection** is not showing **MQTT**, select it in the pull-down.
10. Enter your desired turnout number in the **Hardware Address** field.
11. Enter a name for the turnout in the **User Name** field.
12. Click **Create**. The new turnout should now be listed in the turnout table.
13. If not already checked, check the box at the bottom of the window for **Show Feedback information**.
14. Under **Mode** for your new turnout, select **ONESENSOR**.
15. Under **Sensor1**, select the new turnout's sensor name.
16. If you have the turnout controller connected, click on the **State** for your new turnout. The turnout should change states and the state shown under **Feedback** should change to the selected state.
17. In the list on the left side of the window, select **Sensors**.
18. Repeat steps 2 through 17 for each new turnout.
19. Don't forget to save your changes. (File->Store->Store Configuration and Panels to File...)

Adjusting the Servo Travel

The sketch for the mini is written with default travel from 30 degrees to 120 degrees. To adjust the travel, you can connect a temporary toggle switch to the D0,D1 header and follow this procedure.

You will need to build the following switch assembly. The diode is important to make it work. (There is no internal pullup for D0, so it has to be pulled up by D1's pullup resistor.) Make sure the band on the diode is on the D0 side of the switch. The switch must be momentary. You can use two push buttons instead of the DPDT if that's what you have on hand.

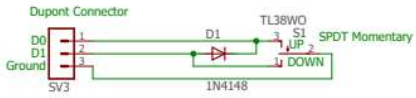

Figure 5: Programming Switch Assembly

The "main" button in this procedure is the one connected to the terminal strip between Push Btn and Ground. It is normally used to switch the turnout.

1. Connect your switch assembly to the D0,D1 header, making sure it is oriented correctly.
2. Hold the main button for about 5 seconds until the servo moves.

3. Use the switch you made to move the servo CW or CCW until you're happy.
4. Hit the main button once. The servo will move to the other end.
5. Repeat step 2.
6. Hit the main button to exit programming mode. The settings will be saved in EEPROM.
7. Reset the mini to test that the values were saved.

Variation

If you only want to run a servo and want to save a little money and time installing the driver chip, just leave it off.