MQTT Turntable Controller Instructions

Description

The MQTT Turntable Controller is designed to provide everything required to operate a turntable or staging yard transfer table, including the stepper motor driver. Its small size makes it easy to hide under the track.

- ESP8266-based Wemos D1 Mini providing WiFi connection to JMRI.
- User-programmable addresses for the rays.
- MQTT messages from JMRI are used for ray selection.
- Base address is set in the sketch.
- Uses AccelStepper library which provides acceleration mode movements.
- Compatible with the A4988-type stepper motor driver, commonly used in 3D printers.
- Up to 32 programmable rays, possibly more if the sketch is modified.
- Maximum stepper motor speed can be set in the sketch.
- Two push button inputs for ray selection and programming.
- LED output to indicate moving and programming status.
- Sends thrown (ACTIVE) and closed (INACTIVE) feedback to JMRI.
- Movement commands can be overridden while moving.

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 250ºC. 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.

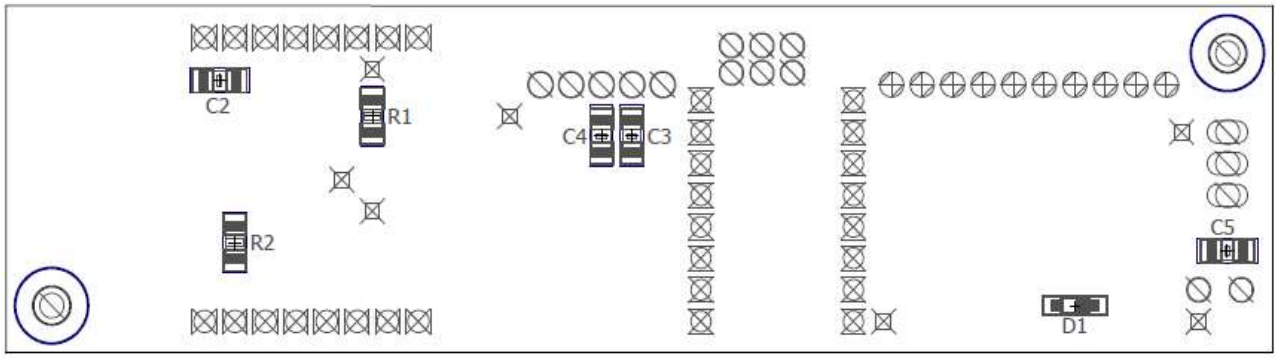
Apply solder paste to all the pads, then place each part.

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.

Table 1: Surface Mount Parts Needed

Qty	Part	Value	DigiKey #	Device
4	C2,3,4,5	.1uF	399-C1206C104K5RAC7800CT-ND	C-USC1206
1	D1	1N4148	MMSD4148T1GOSCT-ND	SOD123
1	R1	182K 1%	311-182KFRCT-ND	R-US_M1206
1	R2	1K	311-1.0KERCT-ND	R-US_M1206

Figure 1: Surface Mount Parts Placement



After the board(s) cool, use a magnifying glass to check for bridging between any parts 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.

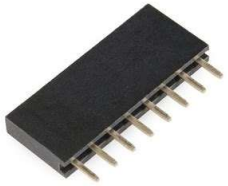


Figure 3: Short female header.

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 board. I use a dead mini for this step.

Solder the remaining pins of the headers and remove the mini.

Install the headers for the stepper driver the same way you did for the mini.

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	SV1	3 pin male header	na	na
1	TS1	10 pos term block	ED10566-ND	na
1	TS2	5 pos term block	A98336-ND	na
1	U$1	WEMOS-D1-MINI	na	See Instructions
1	U$2	A4988	na	See Instructions
4	na	8-pin short female header	included with D1 Mini	

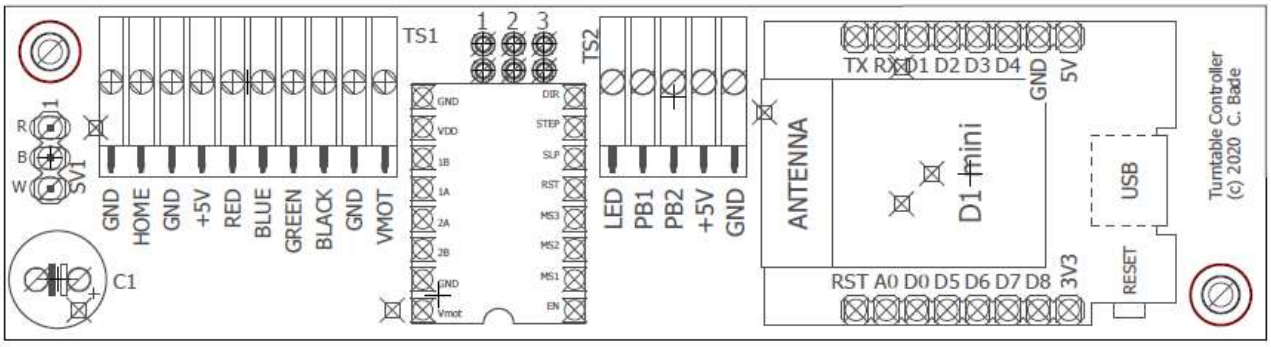


Figure 4: Through-Hole Parts Placement

The jumpers marked 1, 2, and 3 are for setting the stepper motor speed. You can install header pins and use some old PC or disk drive jumpers, or you can just solder wires across the ones you think you need.

Driver Jumper Settings

The A4988 stepper motor driver module has three jumpers to control motor step resolution. On the A4988 module, these jumpers are called MS1, MS2, and MS3. On my circuit board these are connected to pins labeled 1, 2, and 3 respectively.

A Nema 17 stepper motor has 200 steps per revolution. The A4988 Driver generates an analog signal that makes it possible to realize up to 3200 steps per revolution. If the motor was connected to the table directly, 3200 steps would allow 32 rays with 100 counts per ray. For indirect drive, a lower microstep setting, like 1/8 or 1/16 could be used but this results in noisier operation. Acceptable results can be achieved by leaving the jumpers set for the 1/16 setting and increasing the speeds. Since the AccelStepper library uses a long integer to store the step value, you could track the position on a T8 lead screw almost a mile long! Obviously a lead screw that long would be too heavy to turn. I'm using a 16" long lead screw for my staging yard with the speeds turned up high and I get very acceptable results.

My point is don't use the other jumper settings because you won't be happy with the performance. Just install wire jumpers in all three positions. The other settings are available just in case someone needs to do something out of the ordinary.

Table 3: Stepper Resolution Jumpers

Jumper Number				
1	**2**	**3**	**Steps Per Revolution**	**Resolution**
N	N	N	200	Full Step
Y	N	N	400	Half Step
N	Y	N	800	Quarter Step

Y	Y	N	1600	Eighth Step
Y	Y	Y	3200 (Recommended)	Sixteenth Step

Control Box Wiring

You will need 2 push buttons and a panel-mount LED for your control box.

1. Wire one side of each of the push buttons and the negative side of the LED together. Label this wire GND.
2. Label one of the push buttons "+" and connect a wire to it, labeled PB1.
3. Label one of the push buttons "-" and connect a wire to it, labeled PB2.
4. Connect a wire to the positive side of the LED and label it LED.

Table 4: Button Functions

Button	Normal Operation	Ray Programming
+	Forward	Forward
-	Reverse	Save

Testing

Initial testing should be done with the Mini removed.

1. Connect a 12VDC power supply to the VMOT and GND connections on the block controller, but leave the power off for now.
2. Connect the stepper motor to the appropriate color connections.
3. The board is designed to take advantage of an off-the-shelf optical endstop sensor, part number BIQU-3D0512, available on Amazon. Connect the home sensor to the 3-pin header on the end of the board using the supplied cable. If you don't have an optical flag sensor, you could use a micro-switch wired to HOME and GND, but the switch cannot be allowed to get in the way of the table so it needs to ride up on some sort of cam.
4. Connect the 4 wires from the control box to their respective connections on TS2.
5. You can either use a mini USB cable to power the unit, or you can connect a 5v power supply between +5V and GND on TS2.

NOTE: Carefully observe the orientation of the modules you are about to plug in.

6. Program a mini and plug it in to the block controller. See the Programming section below.

7. Install the stepper driver.
8. Turn on the power.
9. Press the “+” button. The table will attempt to go home. If it goes the wrong direction, change the value of ToHome and Away. After going home pressing the “+” button momentarily won’t move the table but the blue LED on the mini should come on while the button is pressed.
10. Set the motor current to the rated current of your motor. Refer to instructions you can find online for setting the current limit control for an A4988. It's pretty simple with the correct instructions but you need the instructions for your specific driver.
11. The sketch includes a routine to determine the number of steps for each revolution of the turntable. To start the routine, hold the “-” button while resetting the D1 Mini. After the 2nd revolution, the program will report the number of steps and an average for as many revolutions as it is allowed to take. After about a 100 revolutions, you will have a pretty solid average.
12. See the programming sections.

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 them.

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 more easily 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 (not working prior to JMRI version 4.21.1)
- 500-599 Block Occupancy Detectors
- 500 1st Block Detector
- 501 (future use, unused input on block controller)
- 502, 503 2nd Block Detector/Controller
- ...
- 548, 549 25th Block Detector/Controller
- 600-699 Turntable / Staging Yard Feedback

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
- AccelStepper by Mike McCauley
- WiFi by Arduino
- PubSubClient by Nick O'Leary
- Ethernet by various

D1 Mini Configuration

1. Go to https://github.com/chuckbade/MQTTtrains
2. Download the latest ino file from the MQTT Turntable Controller directory.
3. Open it in Arduino IDE.
4. Change the values near the top of the sketch to match your network. You can uncomment and set the SSID and PASSWD values or create a header file as instructed for better (a little better) security.
5. Change the JMRISensorNumber to the first sensor number in JMRI that will represent the rays of the turntable. When a particular ray is “thrown” the turntable controller will send messages to JMRI showing all the other rays as “closed” (INACTIVE).
6. Change the JMRITurnoutNumber to the first turnout in the series for the turntable being set up. The base address is the address of the first ray, then the other 31 rays are contiguous addresses above that. The best way I found to achieve this is to create a series of turnouts in JMRI.
7. Change “ToHome” and “Away” to 1 (CW) or -1 (CCW), based on your setup.
8. Change “SensorAtHome” to match the type of home sensor you’re using. If you’re using an optical flag sensor, this value should be HIGH. If you’re using a normally open switch, this value should be LOW.
9. The maximum motor speed can be changed in the sketch by changing the variable "MaximumSpeed". This maximum speed will only be seen on large movements.
10. MaxHomingSpeed is set lower than MaximumSpeed to prevent excessive overshoot after the home sensor is detected.
11. Connect the Mini to the USB cable and compile the sketch.
12. After is says it is resetting the device, disconnect the Mini and connect it to the turntable controller.

Table 5: Stepper Tuning Variables

Variable	Direct Drive	T8 Lead Screw
MaximumSpeed	500	4000
MaxHomingSpeed	200	600
AccelerationFactor	30	500
ProgStepSize	1	15
StepsPerRev	3200	9999999

JMRI Configuration

Connecting to MQTT

1. If you have not yet set up the MQTT connection in JMRI, from the main window, go to **Edit → Preferences**. If **Connections** is not selected in the list on the left side, select it.
2. Select the **MQTT** tab. Under **System Manufacturer** select **MQTT**.
3. Under **System Connection** select **MQTT Connection**.
4. Under **Settings** enter the **IP Address** or **Host Name** of the computer where you have your MQTT server/hub running. I run mosquitto (spelled with two t's) on one of my Linux boxes. I highly recommend setting up a static IP on your router for that computer.
5. Use the default **Connection Prefix** of **M** and **Connection Name** of **MQTT**.
6. Click **Save**.

Adding Inputs

1. From the main JMRI window, go to **Tools → Tables → Sensors**. Click the **Add…** button (bottom left).
2. Select **System Connection: MQTT**
3. Enter a **Hardware Address** i.e. 600 and a **User Name** i.e. RaySensor0.
4. Click the **Add a sequential range** check box and set **Number of items** to the number of rays on the turntable or transfer table.
5. Click **Create**. You should now see all the sensors you just created in the table.
6. Save your changes by going to **File → Store → Store Configuration And Panels To File…** and choose a file in which to save the JMRI configuration.

Adding Outputs

1. From the main JMRI window, go to **Tools → Tables → Turnouts**.
2. Re-size the window to allow you to see the columns labeled **Mode** and **Sensor1**.
3. Click the **Add…** button (bottom left).
4. Select **System Connection: MQTT**
5. Enter a **Hardware Address** i.e. 100 and a **User Name** i.e. Ray0.

Note: Normally, I click the **Add a sequential range** check box and set **Number of items** to the number of rays on the turntable or transfer table, but it doesn't currently work for MQTT.

6. Click **Create**. Your new turnout should show up in the Turnout Table.
7. Re-size the window to allow you to see the columns labeled **Mode** and **Sensor1**.
8. On the row of the sensor you just created, click on the **Mode** selection list and select **ONESENSOR**.
9. Click on the **Sensor1** selection list and select the sensor that corresponds with the new turnout.
10. Repeat steps 3 thru 9 for additional rays as needed.
11. Make sure you save your changes by going to **File → Store → Store Configuration And Panels To File…** and choose a file in which to save the JMRI configuration.

Programming Ray Positions

The first ray position is the distance from to Ray 1, then from home to 2, and so on.

When you start programming rays, you start from whatever ray the controller was positioned on and program rays from the current ray on up. If those rays have already been programmed, their old position values will be deleted.

1. To start ray programming, hold the "+" button for about 5 seconds until the LED turns off.
2. Press and hold or momentarily hit the "+" button to move to the desired position.
3. Press the "-" button to set the position in memory. The LED will turn on.
4. Repeat steps 2 and 3 for additional rays.
5. Hold the "-" button for about 5 seconds to exit programming mode. The LED will turn off then back on.
6. Cycle power or reset the mini. After the table goes home, hit the "+" button to check the ray positions.

To clear all rays and start over, press and hold the "+" button and while holding it, press and hold the "-" button until the LED turns off. All rays will be cleared and the table will go home.

Normal Operation

On power-up, the table will move to the home sensor, which is also track zero. The + button needs to be hit to move to track one. If the – button is hit while sitting on the first track, the stepper will move to the last track.

A newly programmed module will not move until the rays are programmed.

To move to the next highest ray, hit the "+" button. If the + button is hit while sitting on the last track, the stepper will move to the first track.

To move to the previous ray, hit the "-" button. If the – button is hit while sitting on the first track, the stepper will move to the last track.

NOTE: To avoid sudden stops, known as earthquakes to the occupants of your trains, don’t hit the + or – button while the table is moving. Massive derailments can occur especially on a staging table.

Selecting Rays via DCC/JMRI

To select a ray with a throttle, set the turnout corresponding to the desired ray to "thrown". The turntable will only respond to "thrown" commands.

To select a ray with JMRI, click on the circle or other indicator for the desired ray. This assumes the rays have been assigned to the correct turnouts. You can do this in the turnout table or on the panel.

If a ray is selected while the table is still moving, the previous selection will be canceled in progress and the new new selection will be executed. If a reverse direction is required, the stepper will slow to a stop then start movement toward the new destination.

The LED will turn off while the table is moving.