MQTT Robocut Instructions

Description

The MQTT Robocut is designed to provide a way to uncouple cars remotely with a servo. Its small size makes it easier to hide in a boxcar.

- A 3.3v servo output on a 3-pin header. Speed can be adjusted in the sketch.
- A bidirectional LED output available on a 2-pin header.
- ESP8266-based Wemos D1 Mini providing WiFi connection to JMRI.
- User-programmable addresses for 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.

This device is designed to be used with a battery shield that plugs on top of, or under the D1 mini.

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^{o}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.

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

Figure 1: Parts Placement

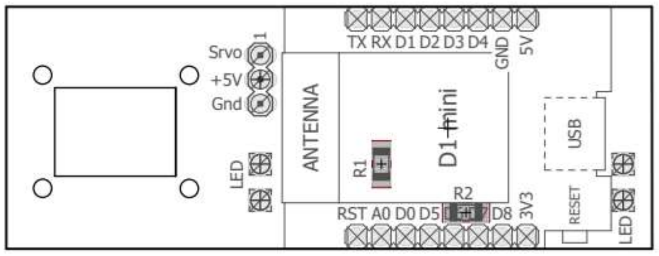

Table 1: Surface Mount Parts Needed

Qty	Part	Value	DigiKey #	Device
1	R1	182K 1%	311-182KFRCT-ND	R-US_M1206
1	R2	1K	311-1.0KERCT-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.

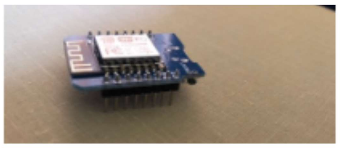
Figure 2: Mini with male headers installed

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.

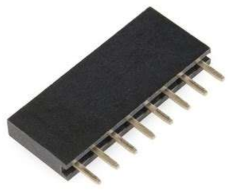

Figure 3: Short female header.

Solder the remaining pins of the headers.

Table 2: Through-Hole Parts Needed

Qty	Part	Value	DigiKey #	Device
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

Testing

1. Program a mini and plug it in to the robocut board. See the Programming section below.
2. Apply power to the battery shield.
3. Connect two LEDs to the led outputs but reverse the polarity of one of them.
4. Connect the servo to the 3-pin header. Make sure the black wire goes to the pin marked Gnd, red to +5V, and white to Srvo.
5. Send a command to the address you've configured in the sketch. The servo should cycle.

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
- PubSubClient by Nick O'Leary

D1 Mini Configuration

1. Go to https://github.com/chuckbade/WiFi-MQTT-Modules
2. Download the latest ino file from the MQTT Robocut directory.
3. Open it in Arduino IDE.
4. If you don’t have a password file, create one in Documents/Arduino/libraries/Personal/SSIDPASSWD.h
5. In this file, place the following definitions:

   ```
   #define MYSSID "myssid"
   #define PASSWD "mypasswd"
   #define MQTTIP "192.168.1.13"
   ```
6. You will need to set up an MQTT broker somewhere on your network and have it start automatically. (See my document **MQTT Instructions).** The IP address should be listed in the file above for the sake of the other sketches. The MQTT Monitor uses its own configuration for this IP address.Change the **Ssid** and **Pswd** values near the top of the sketch to match your network.
7. 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.
8. Change the **JMRITurnoutNumber[], JRMISensorNumber[], and ServoAngle[]** for the “turnout” being configured, which is the address of your robocut car.
9. The **ServoDelay** variable slows the servo down for more realistic operation. For this application, a value of 2 seems adequate.
10. Connect the Mini to the USB cable and compile the sketch.

11. 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 10 degrees to 170 degrees with 90 degrees being the off position. Change the ServoAngle[] values to achieve the travel you need.