MSD Team P19345 User Guide

This document is intended to serve as a guide / outline for users in the RIT Model Railroad Club using our project's produced system. The information included in this document is relevant to the status of the project at the end of the Senior Design II Fall 2019 semester.

Table of Contents

Hardware

Inputs / Outputs

This board is intended to serve as a platform to interface with a Raspberry Pi Zero. All of the I/O communications is performed via MCP23017 I2C GPIO expanders. The I/O is as follows.

J3

This connector is used to power the board with an external DC source. The minimum and maximum voltages that may be applied are 10VDC and 17VDC max. **See J2 for important information regarding this connector.**

J1

This connector is used for providing power to the board with a 5VDC barrel jack. This uses a barrel jack with a 2.1 mm ID/5.5 mm OD center positive connector. An example of this type of connector is provided below. **See J2 for important information regarding this connector.**

https://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?langId=-1&storeId=10001&catalogId=10001&productId=2225529&

J2

If J1 is used to power the board, J2 **must not** be populated (must not be shorted together with a jumper). If J3 is used to power the board, J2 **must be** populated. Failure to comply may result in damage to the board.

J4

J4 must not be populated if the Raspberry Pi is powered (PWR IN) from an external USB source. Under normal operation, J4 will be populated. If the Raspberry Pi is connected to a USB source for power via the “PWR IN” port on the Raspberry Pi, do not short J4.

RS-1

This connector is used for board-board RS-485 communication. This is a differential pair (A and B), and works best when the 3 wires (A, B, GND) are a twisted pair.

SW1 - Address: 0x20

This device is dedicated to up to 16 user-accessible inputs from control panel buttons. The buttons on the control panel are either route-based, or toggle-based commands. Pullup resistors for the buttons are included on the PCB.

HYBRID1 - Address: 0x21

This device is versatile and user-configurable, allowing for the device to be used for either inputs (via the control panel buttons) or outputs (LEDs for indication and signaling). The **hardware configuration** for using this device as an input or output is seen below. **There are additional software configurations that need to be changed, depending on whether a connection (HA0 - HB7) is an input or output.** DNP = “Do not place/populate”

Connection	Input (switch)	Output (LED)	Notes
HA0	R60: 10K R89: 0R	R60: DNP R89: 470R	R60 on bottom
HA1	R61: 10K R85: 0R	R61: DNP R85: 470R	R61 on bottom
HA2	R62: 10K R81: 0R	R62: DNP R81:470R	R62 on bottom
HA3	R63: 10K R77: 0R	R63: DNP R77: 470R	R63 on bottom
HA4	R64: 10K R75: 0R	R64: DNP R75: 470R	R64 on bottom
HA5	R65: 10K R73: 0R	R65: DNP R73: 470R	R65 on bottom
HA6	R66: 10K R71: 0R	R66: DNP R71: 470R	R66 on bottom
HA7	R67: 10K R69: 0R	R67: DNP R69: 470R	R67 on bottom
HB0	R52: 10K R68: 0R	R52: DNP R68: 470R	R52 on bottom
HB1	R53: 10K R70: 0R	R53: DNP R70: 470R	R53 on bottom
HB2	R54: 10K R72: 0R	R54: DNP R72: 470R	R54 on bottom
HB3	R55: 10K R74: 0R	R55: DNP R74: 470R	R55 on bottom
HB4	R56: 10K R76: 0R	R56: DNP R76: 470R	R56 on bottom

HB5	R57: 10K R80: 0R	R57: DNP R80: 470R	R57 on bottom
HB6	R58: 10K R84: 0R	R58: DNP R84: 470R	R58 on bottom
HB7	R59: 10K R88: 0R	R59: DNP R88: 470R	R59 on bottom

LED1 - Address: 0x23

This device is dedicated to driving LEDs to indicate switch position, block occupancy, and signaling. The connections associated with these devices are strictly outputs. Current limiting resistors are present on the PCB, and do not need to be added externally.

LED2 - Address: 0x24

This device is dedicated to driving LEDs to indicate switch position, block occupancy, and signaling. The connections associated with these devices are strictly outputs. Current limiting resistors are present on the PCB, and do not need to be added externally.

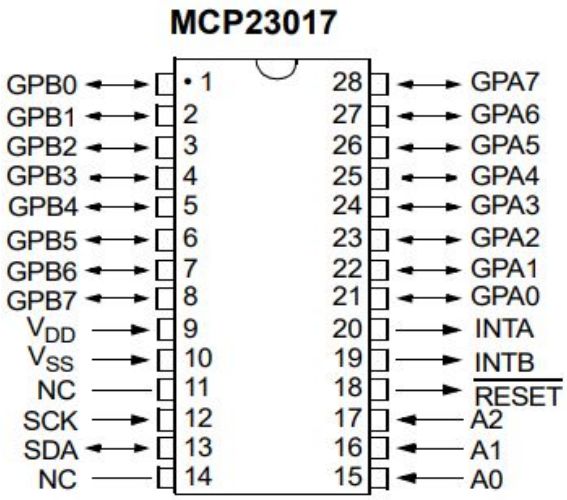
LED3 - Address: 0x22

This device is dedicated to driving LEDs to indicate switch position, block occupancy, and signaling. The connections associated with these devices are strictly outputs. Current limiting resistors are present on the PCB, and do not need to be added externally.

Off-board connections

The 80 off-board connections (buttons, LEDs) are described as follows. “Address” indicates the I2C device address corresponding to SW1, HYBRID1, LED1, LED2, LED3. “Port” indicates the port on the MCP23017 GPIO expander that corresponds to that connection. Further technical and necessary information can be seen in the MCP23017 datasheet accessible through the following link.

MCP23017 Datasheet: http://ww1.microchip.com/downloads/en/DeviceDoc/20001952C.pdf



MCP23017 pinout

SW1 - Address: 0x20

Connection	Port
SA0	GPA0
SA1	GPA1
SA2	GPA2
SA3	GPA3
SA4	GPA4
SA5	GPA5
SA6	GPA6
SA7	GPA7
SB0	GPB0
SB1	GPB1
SB2	GPB2
SB3	GPB3
SB4	GPB4
SB5	GPB5

SB6	GPB6
SB7	GPB7

HYBRID1 - Address: 0x21

Connection	Port
HA0	GPA0
HA1	GPA1
HA2	GPA2
HA3	GPA3
HA4	GPA4
HA5	GPA5
HA6	GPA6
HA7	GPA7
HB0	GPB0
HB1	GPB1
HB2	GPB2
HB3	GPB3
HB4	GPB4
HB5	GPB5

HB6	GPB6
HB7	GPB7

LED1 - Address: 0x23

Connection	Port
D19	GPA0
D17	GPA1
D15	GPA2
D13	GPA3
D11	GPA4
D9	GPA5
D7	GPA6
D5	GPA7
D4	GPB0
D6	GPB1
D8	GPB2
D10	GPB3
D12	GPB4
D14	GPB5

D16	GPB6
D18	GPB7

LED2 - Address: 0x24

Connection	Port
D35	GPA0
D33	GPA1
D31	GPA2
D29	GPA3
D27	GPA4
D25	GPA5
D23	GPA6
D21	GPA7
D20	GPB0
D22	GPB1
D24	GPB2
D26	GPB3
D28	GPB4
D30	GPB5

D32	GPB6
D34	GPB7

LED3 - Address: 0x22

Connection	Port
D51	GPA0
D49	GPA1
D47	GPA2
D45	GPA3
D43	GPA4
D41	GPA5
D39	GPA6
D37	GPA7
D36	GPB0
D38	GPB1
D40	GPB2
D42	GPB3
D44	GPB4
D46	GPB5

D48	GPB6
D50	GPB7

Software

Setting up WiFi on a RaspberryPi

1. Open a terminal
2. Type command

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

3. Add to file

```
network={
      ssid=”RIT”
      scan_ssid=1
      key_mgmt=WPA-EAP
      group=CCMP TKIP
      eap=PEAP
      identity="RIT username"
      password="RIT password"
      phase1="peapver=0"
      phase2="MSCHAPV2"
}
```

   Text in red should be filled in with different values

4. Write to file and save it
5. Type command

```
sudo nano /etc/network/interfaces
```

6. Add to bottom of file

```
auto lo

iface lo inet loopback
iface eth0 inet manual
```

```
allow-hotplug wlan0

iface wlan0 inet manual
      pre-up    wpa_supplicant    -B    -Dwext    -i    wlan0    -c
      /etc/wpa_supplicant/wpa_supplicant  <  DO  NOT  INCLUDE  THIS
      CONTENT - this should be one line >
      post-down killall -q wpa_supplicant
```

7. Write to file and save it
8. Reboot Pi
9. Type command to check if worked or not

```
ping google.com
```

Packets should be transmitting when ENTER is pressed. Use CTRL+C to quit out of this command

Setting up wiringpi to work with Python 3

The steps listed below are assuming WiFi is set up on the RaspberryPi. Look at section Setting up WiFi on a RaspberryPi to set up WiFi.

Steps to getting wiringpi to work with Python3

1. Open a terminal on the Pi
2. Type the command

```
sudo apt-get install python3-pip
```

3. Type the command

```
sudo apt-get update
```

This is to make sure the python3-pip library is being used and is recognized as a library

4. Type the command

```
pip3 install wiringpi
```

Steps to get I2C working with Python 3 after installing wiringpi

There should be no difference with running this library in Python 2 and Python 3, but because we are using Python 3 for this project, this section is specified to work with Python 3.

1. Open a terminal on the Pi
2. Type the command

```
sudo raspi-config
```

3. Select `5 Interfacing Options`
4. Select `P5 I2C`
5. Select `<Yes>`
6. Select `<Ok>`
7. Select `<Finish>`

UART Communication via Terminal on RaspberryPis

The steps listed are assuming the proper hardware connections are made between two RaspberryPis for testing purposes.

Settings for UART use via the terminal

Steps for using the PL011 UART port

For all RaspberryPis:

1. Open a terminal
2. Use the command

```
sudo nano /boot/config.txt
```

3. Add to the bottom of the file

```
dtoverlay=pi3-miniuart-bt
```

4. Save changes and exit back to the terminal
5. Use the command

```
sudo reboot
```

Steps for disabling the serial console

For all RaspberryPis:

1. Open a terminal
2. Use the command

```
sudo nano /boot/cmdline.txt
```

3. Find the text below and remove it

```
console=serial0,115200
```

4. Save changes and exit back to the terminal
5. Use the command

```
sudo reboot
```

After all of these steps have been completed, the UART serial port at `/dev/ttyAMA0` can now be accessed.

Ensuring serial connections using the pins

1. Open a terminal
2. Use the command

```
sudo raspi-config
```

3. Select `Interfacing Options`
4. Select `Serial`
5. Select `<No>`
6. Select `<Yes>`
7. Select `<Finish>`
8. Reboot the RaspberryPi, if asked to

Testing if settings for UART use are correct

The steps below are to test if the settings were set correctly.

1. Have both pis turned on and ready to use
2. Have a terminal open on both pis
3. For both pis, use command

```
stty -F /dev/ttyAMA0 9600
```

9600 is the baud rate

4. For the pi that is receiving the information, use command

```
cat < /dev/ttyAMA0
```

This command should hang with no output until the transmitter command has been executed

The command below can be used if you would like to print out the content of a text file on the receiver

```
cat someTextFile.txt > dev/ttyAMA0
```

5. For the pi that is transmitting the information, use command

```
echo "Some text here" > /dev/ttyAMA0
```

Whatever is put in for "`Some text here`" or whatever content is in the file replacing the whole phrase, it should be printed out on the terminal of the receiving pi