GETTING STARTED

PREREQUISITES

In addition to the Digital Transceiver for the Raspberry Pi, you'll need these components:

- Raspberry Pi 3 Model B+, Raspberry Pi 3 Model B, or Raspberry Pi 2 Model B. The Digital Transceiver for the Raspberry Pi will likely work with more recent Raspberry Pi models as they are released. Check the support page at https://www.brandenburgtech.com for information on using the Digital Transceiver for the Raspberry Pi with new Raspberry Pi models. Visit https://www.raspberrypi.org/products/ for more information on Raspberry Pi models.
- Power supply capable of delivering at least 2.5 amps. Visit the Raspberry Pi Foundation page at https://www.raspberrypi.org/learning/hardware-guide/components/power-supply/ for information on power supplies. While the page indicates a 2-amp supply is required, a little more power is desirable.
- 16 GB or larger micro SD card. Be sure to acquire a high-quality card.
- Some way to work with the Raspberry Pi. This is often a monitor with HDMI cable, USB keyboard, and USB mouse all connected to the Raspberry Pi.

INSTALL RASPIAN

https://www.raspberrypi.org/learning/software-guide/quickstart/ has information on the initial installation of Raspbian. If you acquired a micro SD card with NOOBS already installed, you don't have to format the card, download the NOOBS image, or install the NOOBS image on the micro SD card. You can just insert the micro SD card into the Raspberry Pi, supply power to the Raspberry Pi, then elect to install Raspbian when the NOOBS menu appears.

UPDATE RASPBIAN

It is a good idea to update the Raspbian operating system before you attach new hardware or install new software on the Raspberry Pi. Follow the instructions on https://www.raspberrypi.org/learning/software-guide/update-sd-card/ to update the Raspbian operating system.

CONFIGURE RASPBIAN

Use raspi-config, as described at https://www.raspberrypi.org/documentation/configuration/raspi-config.md, to configure your Raspberry Pi. Suggested settings to consider and configure are:

- Change User Password – always a good idea, especially if SSH is enabled
- Boot Options → Desktop / CLI – set to your preferences
- Boot Options → Wait for Network at Boot – turning this on will slow the boot process but ensures the network is available for any applications that start
- Localisation Options → Change Locale – set to en_US.UTF-8 UTF-8
- Localisation Options → Change Timezone – set to desired timezone
- Localisation Options → Change Keyboard Layout – set to Generic 104-key PC → Other → English (US) → English (US) then accept the default keyboard settings
- Localisation Options → Change Wi-fi Country – set to desired location

- Interfacing Options → SSH – enable
- Interfacing Options → SPI – enable
- Interfacing Options → I2C – enable

CONFIGURE I2C

The Digital Transceiver for the Raspberry Pi uses one I2C bus to communicate with the transceiver chip and another I2C bus to communicate to the temperature sensor. By default, the Raspberry Pi is configured with only one I2C bus. To active the second I2C bus and communicate with the temperature sensor add a line to the /boot/config.txt file.

```
cd /boot
sudo cp config.txt config.txt.0
# create a backup before changing the file
sudo vi config.txt
# you may use nano or another text editor
```

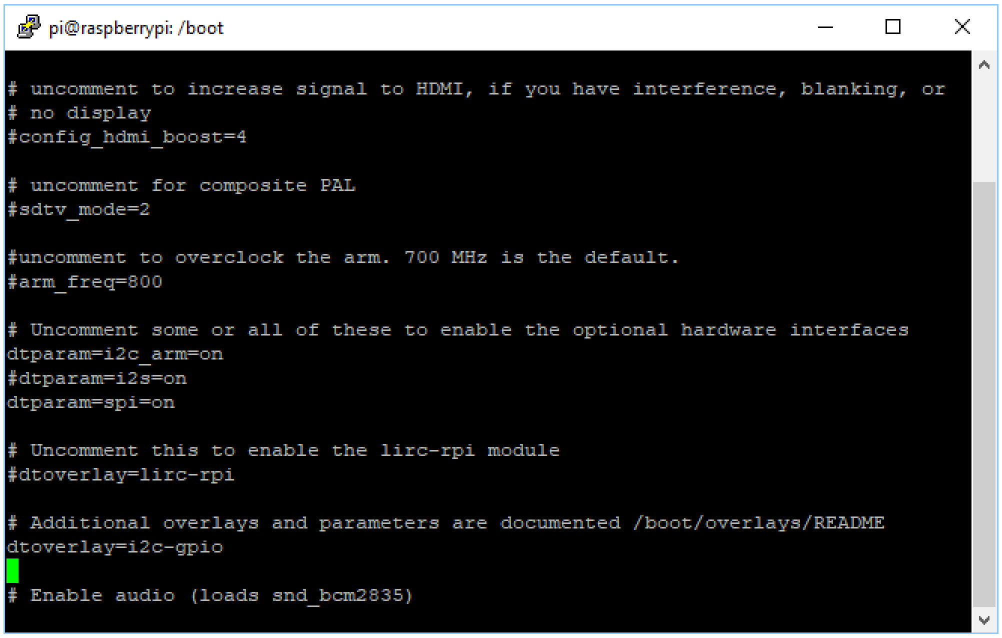
Find the following line.

```
# Additional overlays and parameters are documented /boot/overlays/README
```

Add the following line.

```
dtoverlay=i2c-gpio
```

You file should look like the following image.



```
# uncomment to increase signal to HDMI, if you have interference, blanking, or
# no display
#config_hdmi_boost=4

# uncomment for composite PAL
#sdtv_mode=2

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
dtparam=i2c_arm=on
#dtparam=i2s=on
dtparam=spi=on

# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README
dtoverlay=i2c-gpio

# Enable audio (loads snd_bcm2835)
```

Save the file and reboot the Raspberry Pi.

```
sudo reboot
```

After the Raspberry Pi reboots, log in and confirm the second I2C bus is present.

```
cd /dev
ls i2c*
```

You should see two entries, like the following.

```
i2c-1  i2c-3
```

ACCESS THE TEMPERATURE SENSOR

Ensure utilities are installed to work with the I2C bus and access the sensor.

```
sudo apt-get install -y i2c-tools
```

If the utilities are already installed, you'll see output like the following. Otherwise, you should see the packages downloaded and installed.

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version (3.1.2-3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Run i2cdetect to find the temperature sensor on the I2C bus. The second I2C bus on the Raspberry Pi, the bus with the temperature sensor is usually bus 3. This bus number corresponds to the second I2C entry seen above as "i2c-3".

```
i2cdetect 3
```

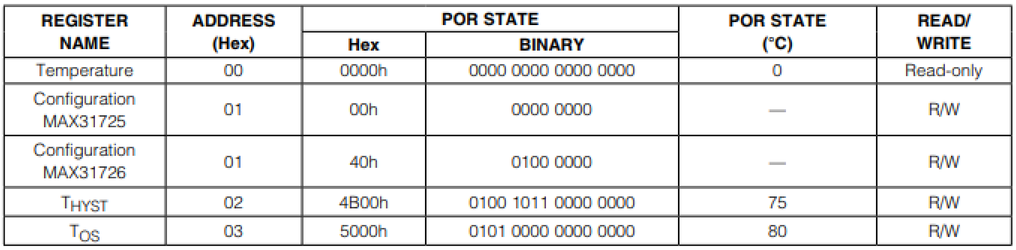
The temperature sensor may be the only device present. In the following result, the temperature sensor is at address 0x48.

```

WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will probe file /dev/i2c-3.
I will probe address range 0x03-0x77.
Continue? [Y/n]
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

To read the temperature, use the i2cget command. We need to know the register and number of bytes for the temperature. Referring to Table 2 in the MAX31725 data sheet, we see the temperature is contained in register 0x00 and consists of two bytes, so it is appropriate to read a word.

Table 2. Register Functions and POR State



REGISTER NAME	ADDRESS (Hex)	POR STATE		POR STATE (°C)	READ/ WRITE
		Hex	BINARY		
Temperature	00	0000h	0000 0000 0000 0000	0	Read-only
Configuration MAX31725	01	00h	0000 0000	—	R/W
Configuration MAX31726	01	40h	0100 0000	—	R/W
T_{HYST}	02	4B00h	0100 1011 0000 0000	75	R/W
T_{OS}	03	5000h	0101 0000 0000 0000	80	R/W

The following command reads the temperature from device 0x48 on I2C bus 3.

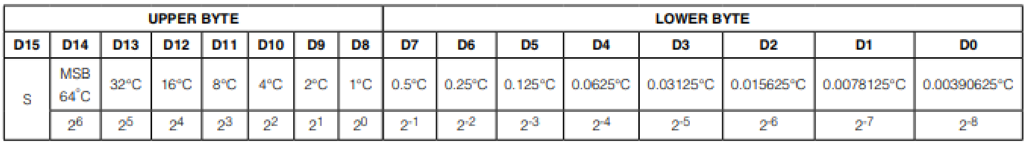
```
i2cget 3 0x48 0x00 w
```

Your value will differ based on the temperature in your environment, but the value will be something like 0x8524.

```
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will read from device file /dev/i2c-3, chip address 0x48, data address
0x00, using read word data.
Continue? [Y/n]
0x8524
```

The Raspberry Pi is “little endian” meaning the least significant byte of a multi-byte value is displayed first. To interpret this value, refer to Table 3 in the MAX31725 data sheet.

Table 3. Temperature, T_{HYST}, and T_{OS} Register Definition



UPPER BYTE								LOWER BYTE							
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
S	MSB 64°C	32°C	16°C	8°C	4°C	2°C	1°C	0.5°C	0.25°C	0.125°C	0.0625°C	0.03125°C	0.015625°C	0.0078125°C	0.00390625°C
	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}

The whole number portion of the temperature is 0x24 or 36°C. The fractional portion of the temperature is $\frac{0x85}{0x100}$ or 0.51953125°C. The temperature is this example is 36. 51953125°C or 97.73515625°F.

ACCESS THE DIGITAL TRANSCEIVER

Follow the instructions at http://wiringpi.com/download-and-install/ to install WiringPi, a GPIO interface library for the Raspberry Pi used by the sample projects.

```
gpio -v
# this command resulted in the following output so I need to remove
# the automatically installed version of WiringPi so I can install
# a newer version
# gpio version: 2.44
# Copyright (c) 2012-2017 Gordon Henderson
# This is free software with ABSOLUTELY NO WARRANTY.
# For details type: gpio -warranty
#
# Raspberry Pi Details:
#   Type: Unknown13, Revision: 03, Memory: 1024MB, Maker: Sony
#   * Device tree is enabled.
#   *--> Raspberry Pi 3 Model B Plus Rev 1.3
#   * This Raspberry Pi supports user-level GPIO access.
sudo apt-get purge wiringpi
hash -r
mkdir -p ~/src
cd ~/src
git clone git://git.drogon.net/wiringPi
cd ~/src/wiringPi
./build
```

Obtain the sample projects from GitHub.

```
mkdir -p ~/src
cd ~/src
git clone https://github.com/BrandenburgTech/DigitalTxRxRPi.git
```

Build the sample project to probe for the AX5043 transceiver.

```
cd ~/src/DigitalTxRxRPi/probeTxRx
make rebuild
./probeTxRx
```

If successful, the output will resemble the following.

```
INFO: Initializing SPI
INFO: Finished initializing SPI
INFO: Resetting AX5043
INFO: Successfully initialized the AX5043
```

NEXT STEPS

If you have more than one Digital Transceiver for the Raspberry Pi, or a friend has a Digital Transceiver for the Raspberry Pi, check out the Digital Transceiver Chat Application for the Raspberry Pi in the Support section at https://www.brandenburgtech.com.

You can also develop and share your own applications. Refer to the data sheets and the programming guides in the Support section at https://www.brandenburgtech.com.