Enabling I2C interface in Raspberry Pi.

e-Yantra Team

July 9, 2016

Contents

1 Objective

In this tutorial we will learn how to enable I2C Interface in Raspberry Pi.

2 Prerequisites

One should have:

- To know to do interconnections on bread board.
- Knowledge of how to do SSH connection.
- Some basic information regarding different serial communication protocols

3 Hardware Requirement

1. Raspberry Pi (I will be using Version 2 Model B+)
2. Power adapter
3. Connecting wires
4. Bread board

4 Software Requirement

MobaXterm (for windows user)

5 Theory and Description

The no. of GPIO pins in an R-Pi is less(merely 26 in R-Pi 2) and so interfacing sensors and other modules would consume most of the pins thus setting a limit for I/O access. Therefore in order to increase the no. of GPIO pins we can interface a port expander to an R-Pi.

A port expander is a hardware device designed to allow a user to utilize more than one device on a single port at one time. For example, the device may allow seven devices to connect to one serial port.[1]

5.1 I2C

I2C (Inter-Integrated Circuit), pronounced I-squared-C, is a multi-master, multi-slave, single-ended, serial computer bus used for attaching lower-speed peripheral ICs to processors and micro controllers.[2]

The two I2C signals are called 'serial data (SDA) and serial clock (SCL). There is no need of chip select (slave select) or arbitration logic in this. Virtually any number of slaves and any number of masters can be connected onto these 2 signal lines and communicate between each other using a protocol that defines:

- 7-bits slave addresses: each device connected to the bus has got such a unique address
- data divided into 8-bit bytes
- a few control bits for controlling the communication start, end, direction and for an acknowledgement mechanism.

The data rate has to be chosen between 100 kbps, 400 kbps and 3.4 Mbps, respectively called standard mode, fast mode and high speed mode. Some IC variants include 10 kbps (low speed mode) and 1 Mbps (fast mode +) as valid speeds.

Physically, the I2C bus consists of the 2 active wires SDA and SCL and a ground connection. The active wires are both bi-directional.[3]

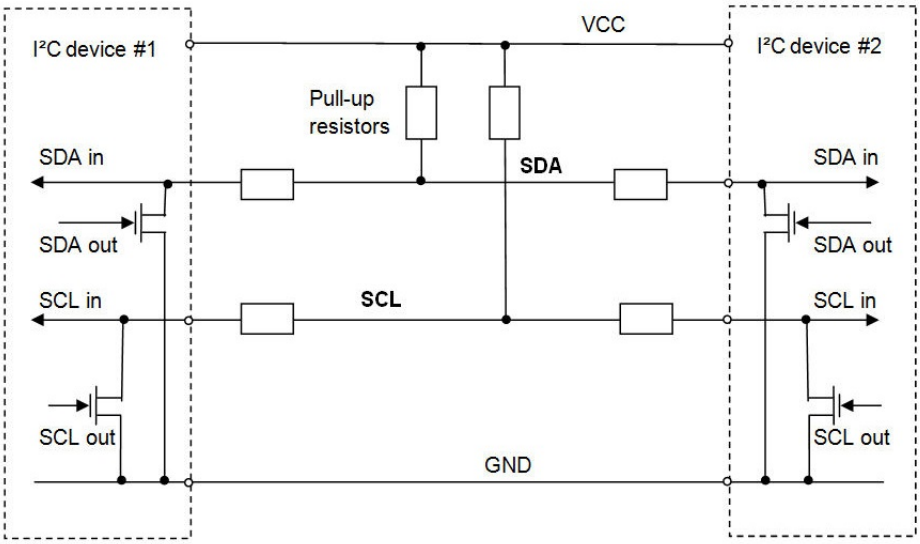

Figure 1: [3]

The I2C protocol specification states that the IC that initiates a data transfer on the bus is considered the Bus Master. Consequently, at that time, all the other ICs are regarded to be Bus Slaves.

Data transfer using I2C

1. First, the master will issue a START condition. This acts as an Attention signal to all of the connected devices. All ICs on the bus will listen to the bus for incoming data.

2. Then the master sends the ADDRESS of the device it wants to access, along with an indication whether the access is a Read or Write operation. Having received the address, all ICs will compare it with their own address. If it doesnt match, they simply wait until the bus is released by the stop condition. If the address matches, however, the chip will produce a response called the ACKNOWLEDGE signal.

3. Once the master receives the acknowledgement, it can start transmitting or receiving DATA.When all is done, the master will issue the STOP condition. This is a signal that states the bus has been released and that the connected ICs may expect another transmission to start any moment.

4. When a master wants to receive data from a slave, it proceeds the same way, but sets the R/W bit at a logical one. Once the slave has acknowledged the address, it starts sending the requested data, byte

by byte. After each data byte, it is up to the master to acknowledge the received data.[3]

5.2 Enabling I2C interface on R-Pi

1. Open MobaXterm.
2. Establish SSH connection to R-Pi.
3. Under Advanced SSH settings.

   If in remote environment you have chosen Interactive shell.

   If in remote environment you have chosen LXDE desktop then open LXTerminal.
4. Type *sudo raspi-config*. This will launch the raspi-config utility.

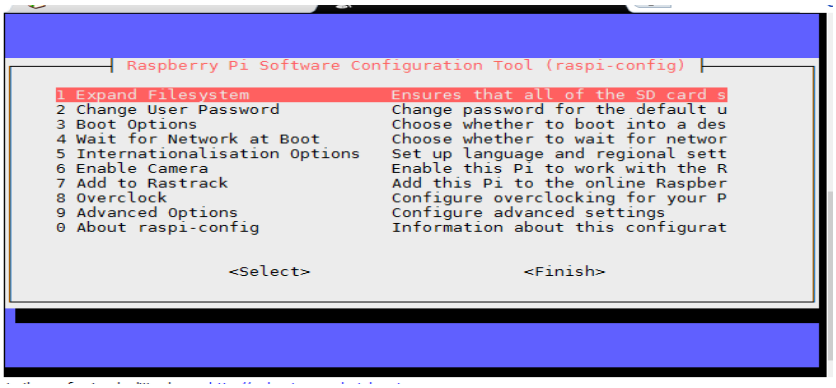

Figure 2: [4]

5. Select the *Advanced options*

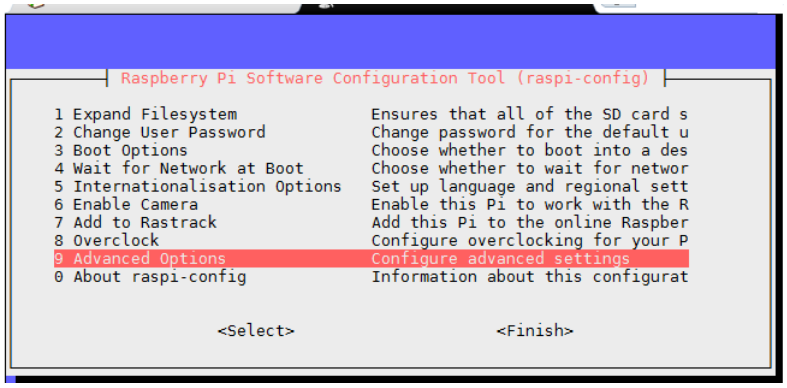

Figure 3: [4]

6. Then select *option A7 I2C*

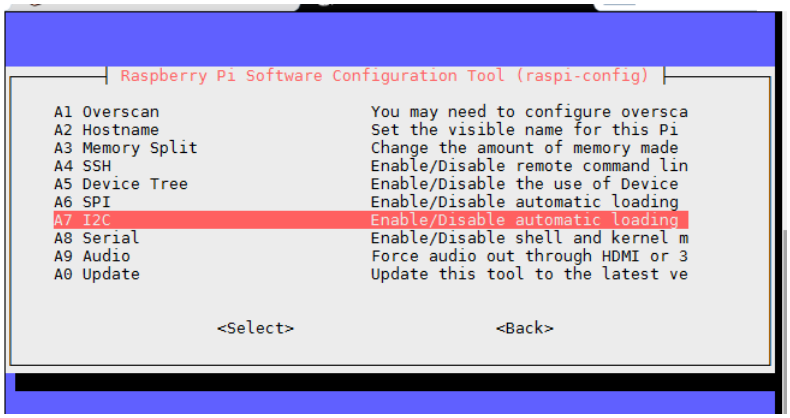

Figure 4: [4]

7. It will ask to enable the ARM I2C interface, click *YES*.

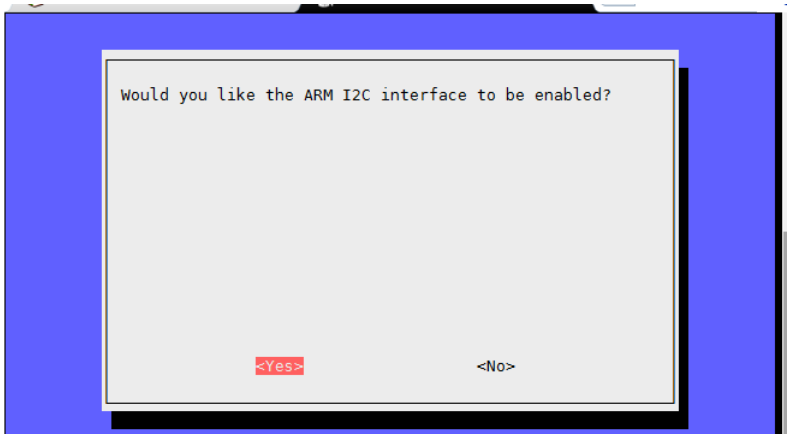

Figure 5: [4]

8. Then it will ask if you would like I2C kernel module to be uploaded by default. Select *YES*.

Figure 6: [4]

9. I2C kernel module will now be loaded by default.Click *OK*

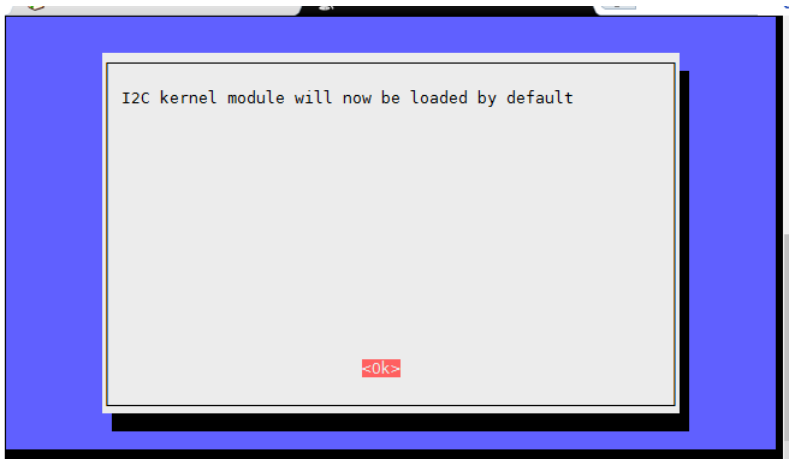

Figure 7: [4]

10. Select Finish to return to command line.

11. Next we need to edit the modules file using :
sudo nano /etc/modules

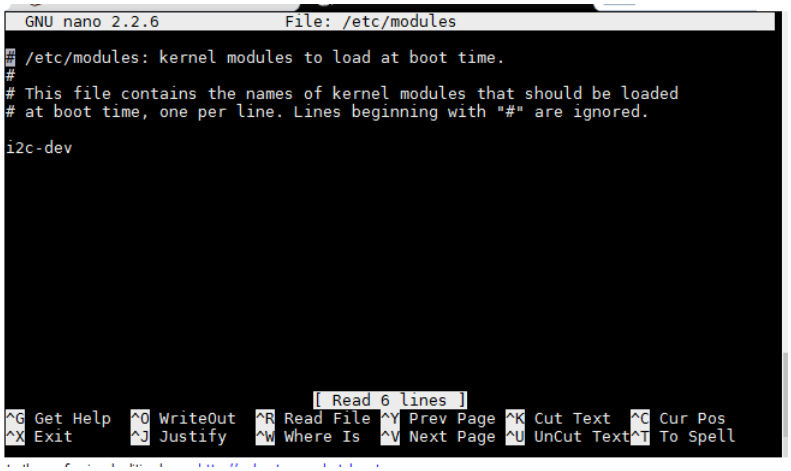

Figure 8: [4]

12. Add the following two lines :
    i2c-bcm2708
    i2c-dev

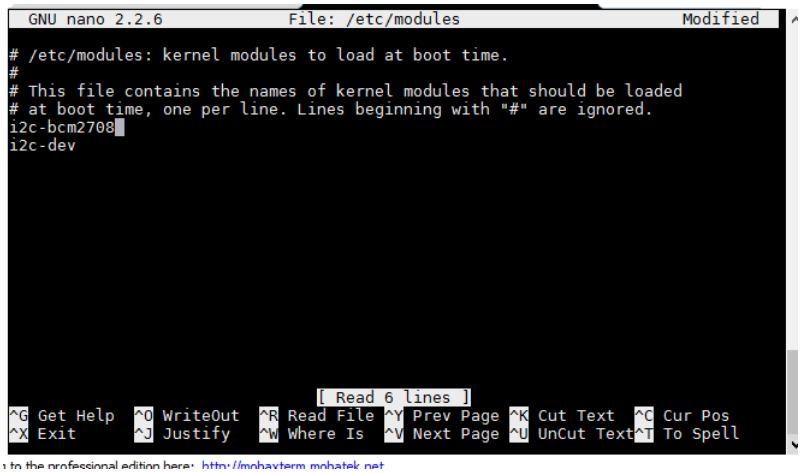

Figure 9: [4]

13. Use CTRL-X, then Y, then RETURN to save the file and exit.

14. To help debugging and allow the i2c interface to be used within Python we can install python-smbus and i2c-tools :
    sudo apt-get update
    sudo apt-get install -y python-smbus i2c-tools

15. Shutdown your Pi using :
    sudo halt

Wait ten seconds, disconnect the power to your Pi and you are now ready to connect your I2C hardware.

16. When you power up or reboot your Pi you can check the i2c module is running by using the following command :
    *lsmod — grep i2c_*
    That will list all the modules starting with i2c_. If it lists i2c_bcm2708 then the module is running correctly.

17. Once you have connected your hardware double check the wiring. Make sure 3.3V is going to the correct pins and you have got not short circuits. Power up the Pi and wait for it to boot.

18. Type the command:
    sudo i2cdetect -y 1

19. You should the output as:

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Figure 10: [4]

6 Appendix

6.1 Raspberry Pi 2 Pin-out Diagram

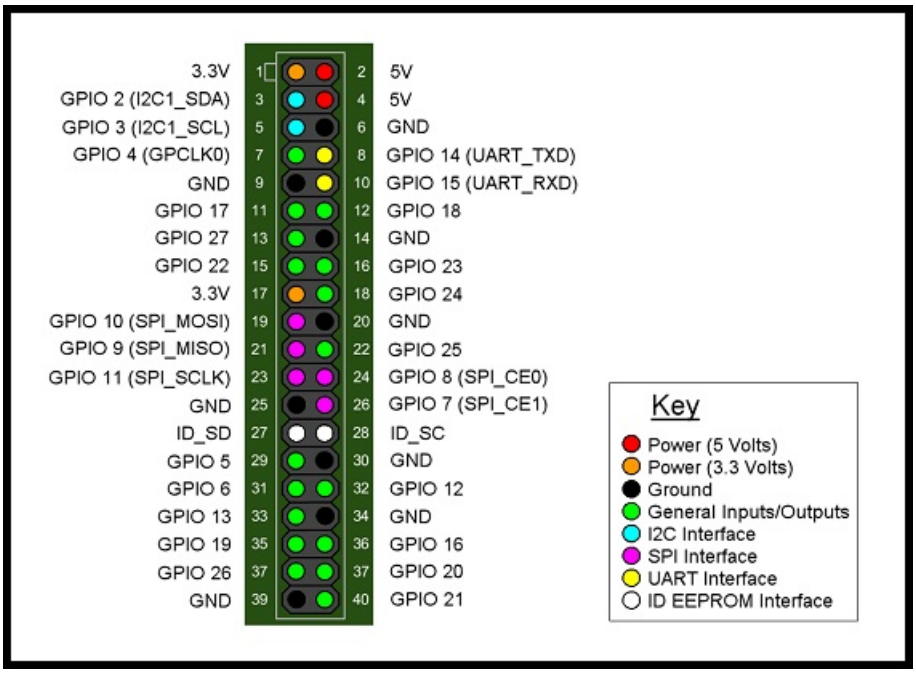

Figure 11: [7]

7 References

1. http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/
2. https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c
3. https://camo.githubusercontent.com/c80be3d0c9e3146cd09e15b7ddf07dbb8b7d40b8/687474703a2f2f692e696d6775722e636f6d2f4575524e722e706e67
4. http://data.designspark.info/uploads/images/53bc258dc6c0425cb44870b50ab30621