# Device Driver for VCNL 4010 proximity Sensor on Raspberry pi

VCNL 4010 is a proximity sensor which works using I2C protocol. The work here presents an I2C device driver to interface this sensor to Raspberry pi.

I have developed this code with Reference to i2c_dev.c driver.

## Purpose of Code:

This code was written as part of Design Project for the Course - Device Drivers, EEE G 547 (BITS PILANI)

The aim was to demonstrate a small part of Mobile phone using Raspberry pi. Since all the Latest phones are Equipped with Proximity Sensor, this code will give clarity regarding how interfacing of such sensor is done at Kernel Level.

Although there are several Techniques to handle an i2c device (Ex: IIO, bus_register_info, sysfs), in this code I have tried to implement a Blend of all the Methods. However the core functionality of client registration to adapter is in accordance with i2c_dev.c, which is a standard kernel module.

The code uses Read/Write spinlock while reading and writing on the bus, Dynamic registration of character Device, GPIO subsystem usage to indicate Proximity level through Led, interrupt status scan, IOCTL and read System calls.

## Hardware Requirements and Connections:

Please Note that I have used Raspberry pi B+ with Jessie Installed.

Version 4.9.21+

## Requirements:

- Raspberry pi Board
- Memory Card
- Raspbian Jessie (Preferred)
- Linux Header installed and compatible with the kernel version
- Adafruit VCNL 4010 Sensor
- Led
- Connecting Wires
- Raspberry pi Datasheet

### Connections:

- Connect Vin pin of Sensor to 5v on Raspberry pi
- Connect SCL to SCL on board
- Ground to Ground
- SDA pin to SDA
- Led to Pin 11 on Raspberry Pi

**Intial Testing of Sensor :**

Once the Connections are complete Initial Testing involves detecting the address of this Sensor. For that you may have to go through some Steps :

- On the Terminal run $ lsmod
- If i2c_dev appears in the list go to step 4
- If it does not appear run $modprobe i2c_dev and check again using lsmod command
- Assuming that i2c_dev driver is present run $ i2cdetect –y 1
- This command should display the slave address of this sensor which is 0x13.

**Code Compilation Procedure:**

- Download the Kernel Level code "i2c.c", user space code "user.c" and Makefile .
- Go to the Directory where these files are downloaded.
- Run $ make all on terminal
- This should generate i2c.ko file successfully
- Compile the user code using command $ gcc user.c
- This should generate a.out executable file
- Insert the module using $ sudo insmod i2c.ko
- On successful insertion of the module run user code
- On terminal run $ ./a.out
- If everything goes right you should see the proximity value on the Terminal.
- To read the Status of Interrupt and other information run $dmesg
- If you bring something close to the sensor the Pin 11 on Raspberry Pi should go high, connect an Led to this Pin.
- To Remove the module run $rmmod i2c.ko

**Possible Errors:**

- **Error while running i2cdetect –y 1 "could not open /dev/i2c-1" No such file or Directory**

Possible Cause : i2c_Dev not in Kernel Module

Solution :Modprobe should be able to insert the module ,However if that does not happen you may have to check with the blacklisted device .

Follow this link: http://www.runeaudio.com/forum/how-to-enable-i2c-t1287.html

Another reference can be Exploring Raspberry Pi by Derek Molloy chapter -8

- **Error while Insmod ,Build directory not found**

Cause :Linux headers are not installed

Solution : Check the version you are running on by $uname –r

Download the Respective Header file from this link : https://www.niksula.hut.fi/~mhiienka/Rpi/linux-headers-rpi/

Once headers are downloaded install it using command :

sudo dpkg -i linux-headers-"Depending on ur version"armhf.deb

eg sudo dpkg -i linux-headers-4.1.19+_4.1.19+-2_armhf.deb

you may have to check with the errors and possible solutions provided in terminal itself and run those commands .

- **Error : BC package not installed**

Solution : sudo apt-get update

Sudo apt-get install bc

- **Error : Could not Download CA certificates due to date/time error**

Set Date and time on Raspberry using :

Ex : $ sudo date -s "Thu Aug  9 21:31:26 UTC 2012"

- **Error during Insmod .ko file format not valid**

Cause : Kernel Version and Header version differ

Check Kernel Version  using uname –r

Check Header version using :

```
$ dpkg -s linux-headers-$(uname -r)
```

If not same you may have to download linux header compatible with kernel from this link :

https://www.niksula.hut.fi/~mhiienka/Rpi/linux-headers-rpi/

Run following command :

```
sudo dpkg -i linux-headers-“Depending on ur version”armhf.deb
```

Alternatively change the Version of Linux kernel using :

https://tech.enekochan.com/en/2014/03/08/upgradedowngrade-to-a-specific-firmware-kernel-version-with-rpi-update-in-raspbian/

An excellent Link for this Problem is :

https://raspberrypi.stackexchange.com/questions/39845/how-compile-a-loadable-kernel-module-without-recompiling-kernel