

# (EEE G547) Device Drivers Project

## Device Driver to Interface VCNL4010 with Raspberry Pi 3

### Introduction:

Several smart devices these days are equipped with proximity sensors. Through this project we aim to gain clarity regarding how interfacing of proximity sensors is done at kernel level. The project demonstrates the functionality of VCNL 4010 that is interfaced via I2C to Raspberry Pi 3. VCNL4010 is a proximity sensor which works using I2C protocol.

In this code, we have used the standard kernel module `i2c_dev.c` to implement the functionality of client registration to adapter. Read/Write Spinlock is implemented while reading and writing on the Bus, the character device is dynamically allocated and GPIO subsystem is used to indicate the proximity level through LED. Interrupt Status Scan and read calls are also implemented in the code.

### Hardware Requirements:

- Raspberry Pi 3
- Memory card
- Adafruit VCNL 4010 proximity sensor
- LED
- Ethernet cable
- USB cable

### Software Requirements:

- Raspbian OS installed on Raspberry Pi 3
- Compatible Linux headers with your kernel versions

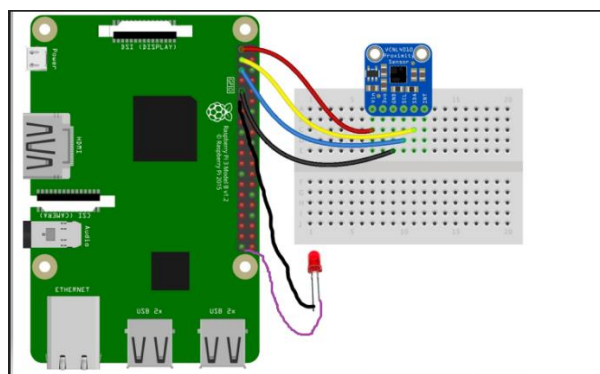
### Documents Required:

- Raspberri Pi datasheet  
([https://www.terraelectronica.ru/pdf/show?pdf\\_file=%252Fds%252Fpdf%252FT%252FTechicRP3.pdf](https://www.terraelectronica.ru/pdf/show?pdf_file=%252Fds%252Fpdf%252FT%252FTechicRP3.pdf))
- VCNL 4010 datasheet  
(<https://www.vishay.com/docs/83462/vcnl4010.pdf>)

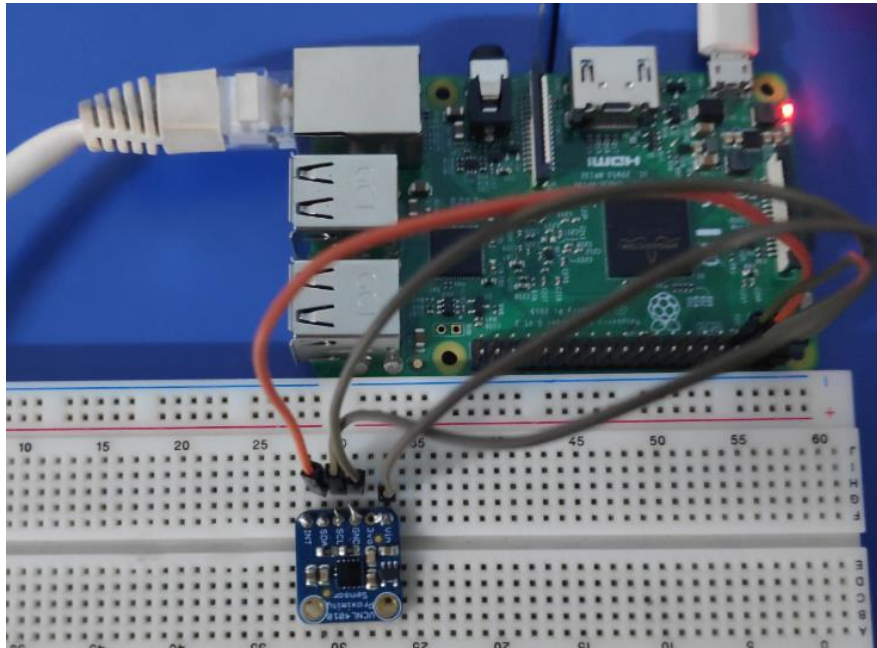
### Connections:

- Connect ground of VCNL 4010 to the ground of the Raspberry pi
- Connect SDA pin of VCNL 4010 to the SDA pin (pin no 3) of the Raspberry pi.
- Connect SCL pin of VCNL 4010 to the SCL pin (pin no 5) of the Raspberry pi.
- Connect Vin pin of VCNL 4010 to the 5v pin (pin no 2) of the Raspberry pi.
- Connect led to GPIO 27 (pin no 13) of Raspberry pi

### Hardware Design (schematic):



## Hardware Design (actual photograph):



### Sensor Test:

Once the connections are complete, check if your Sensor is getting detected in Raspberry pi. Follow the below steps:

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

### Firmware:

`GPIO_Reg_Access` pdf has all the information regarding the GPIO registers.

### Kernel Space driver and its build process along with User Space Application:

Given below are the steps to compile and build the driver and user space programs.

1. Download the folder `vcnl4010_i2c_interface`. The folder contains Kernel Level code "`vcnl_i2c.c`", user space code "`vcnl_user.c`" and Makefile and save them on Desktop or any directory you want to save it in.
2. Go to the Directory where these files are downloaded. E.g. If you have saved in Desktop then use `cd /Desktop/vcnl4010_i2c_interface` to reach the directory.
3. Run `$ make all` on terminal. This should generate `vcnl_i2c.ko` file successfully along with a few other files.
4. Compile the user code using command `$ gcc vcnl_user.c`.
5. This should generate an executable file named `a.out`.
6. Insert the module using `$ sudo insmod vcnl_i2c.ko`.
7. Give permission to the device file using `$ sudo chmod 777 /dev/vcnl4010_sensor`
8. On successful insertion of the module run user code through command `$ ./a.out`. If everything goes right you should see the proximity value on the Terminal.
9. To read the Status of Interrupt and other information run `$ dmesg`.
10. If you bring something close to the sensor the Pin 13 on Raspberry Pi should go high, connect an LED to this Pin. The LED glows if the proximity value is above 6500. The LED remains off if the proximity value is below 6500.
11. To Remove the module run `$ sudo rmmod i2c.ko`

## **Possible Errors and Issues:**

### **1. 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>

### **2. Error while Insmmod, 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/~mhienka/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.

### **3. Error : BC package not installed**

Solution: sudo apt-get update

Sudo apt-get install bc

### **4. 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"

### **5. Error during Insmmod .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/~mhienka/Rpi/linux-headers-rpi/>

Run following command:

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

## **Authors:**

- Kajal Bansala (2018H1400115G), BITS Pilani, KK Birla Goa Campus
- Purab Nandi (2018H1400121G), BITS Pilani, KK Birla Goa Campus
- Surbhi Malani (2018H1400138G), BITS Pilani, KK Birla Goa Campus

## **Licence:**

- This project is licensed under the GNU-GPL License.