Linux Device Driver for Ambient Light Sensor

Syed Irfan

A device driver does more or less the same job for a piece of computer hardware, what a **taxi driver** does for the taxi. That is, manage, control or monitor etc.They are distinct "black boxes" that make a particular piece of hardware respond to a well-defined internal programming interface; they hide completely the details of how the device works. User activities are performed by means of a set of standardized calls that are independent of the specific driver; mapping those calls to device-specific operations that act on real hardware is then the role of the device driver.

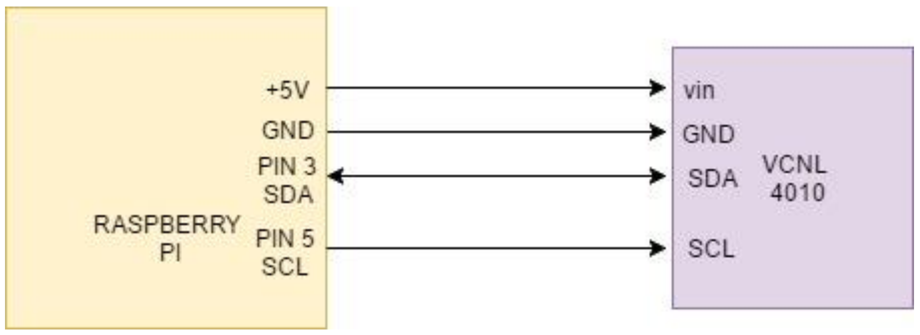
The VCNL4010 is a fully integrated proximity and ambient light sensor. Fully integrated means that the infrared emitter is included in the package. It has 16 bit resolution. It includes a signal processing IC and features standard I2C communication interface. It features an interrupt function.

Purpose Of Code: -

Ambient light sensors are used as backlighting controls in any number of LCD display applications from consumer electronics to automotive, and by automatically adjusting display brightness, they can conserve battery life, which is a key benefit in mobile device applications. In addition, these sensors work well in all kinds of light sources from natural sunlight to fluorescent and incandescent lamps.

As a part of the project, we are designing device drivers for all parts of the mobile. Since Ambient Light Sensor is a part of Mobile phone, I have designed a basic version of this sensor using I2C device driver. To implement it in to a mobile phone and auto adjust the brightness we need to collaborate it with the pwm driver.

Block Diagram: -



How to use:

1. Clone the whole project directory
2. Get into super user mode using $ sudo –s command
3. In the terminal, run the $ make all command, it will build the project
4. Then insert the module using insmod proj.ko
5. Once the module has been inserted to read the sensor values continuously run the python script file
6. To run python file use command $ python script.py

Challenges Faced:

1. Linux headers not installed. We faced this issue while building our module, to overcome this we need to upgrade our kernel to the latest version and download the headers for that kernel version only.

Run all the below commands in root account.
While running all these commands make sure that the date and time of your device are accurate.

To achieve this, we can run a set of commands:

- Usual update routine
  $ apt-get update && apt-get upgrade
- Then we need to update the kernel using
  $ rpi-update

Then we need to reboot the system
Then proceed with the given commands

- To get the rpi-source
  $sudo wget https://raw.githubusercontent.com/notro/rpi-source/master/rpi-source -O /usr/bin/rpi-source

- Then to make it executable
  $ sudo chmod +x /usr/bin/rpi-source
- To tell the update mechanism that it is the latest version of the kernel
  $ /usr/bin/rpi-source –q –tag-update
- To get the kernel file things
  $ rpi-source

Result: -

On running the python script file we will get continuous reading of the sensor values read from the character driver which we have created.