**<How to connect Raspberry PI with system>**

**1. ‘wiringPi’ library for using I2C or other buses**

- By using this library, we can directly build c++ program in Debian for Raspberry PI’s GPIO pin

[Step]

**1) Download**

git clone git://git.drogon.net/wiringPi

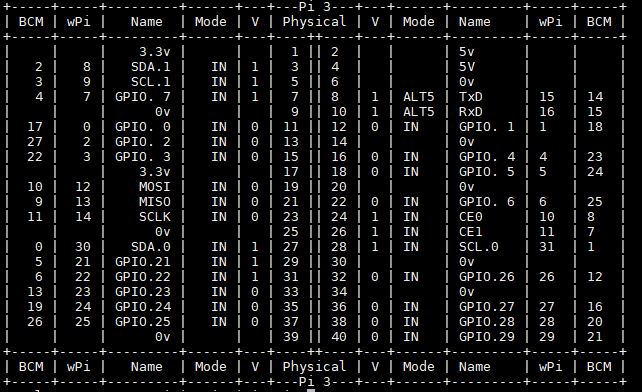
cd wiringPi   
./build

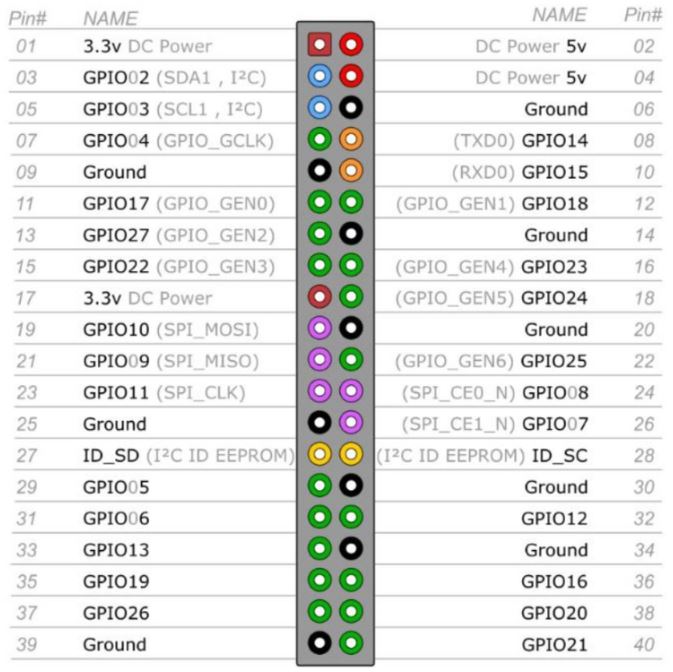
**2) Check**

gpio -v

gpio readall

Then we can see writingPi’s GPIO pin arrangement sheet like below





BCM = Input & Output pin number of Broadcom Processor = Number of GPIO

Physical = Pin number of Raspberry PI’s board

wPi = Pin number of writingPi library

**Example)** If we use ‘pin 36’ in Raspberry PI

#include <iostream>

#include <wiringPi.h>

**#define PRACTICE 27 // BCM\_GPIO 16, PIN 36**

Void PRACTICEexample (void)

{

digitalWrite ( PRACTICE, 0):

}

**2. I2C & SPI**

= <https://learn.sparkfun.com/tutorials/raspberry-pi-spi-and-i2c-tutorial/all>

**<Steps>**

1) Activate Raspberry PI’s buses (I2C or SPI) and install ‘writingPi’ library.

2) we can choose main two methods to compile C++ code

Type A

Write the code in github, and directly download the clone in PI and run = Already demonstrated in lecture

Type B

Using ‘writingPi’ library, directly running the code in Pi which is built from ‘nano’ pad.

If we don’t use ‘writingPi’ library, we have to follow below complicate steps

1) Choose specific GPIO pins, and setup each pin

2) Writing ‘Pre\_GPIOclass’ for controlling single pins

3) Writing main code which is included ‘Pre\_GPIOclass’.

= <http://hertaville.com/introduction-to-accessing-the-raspberry-pis-gpio-in-c.html>