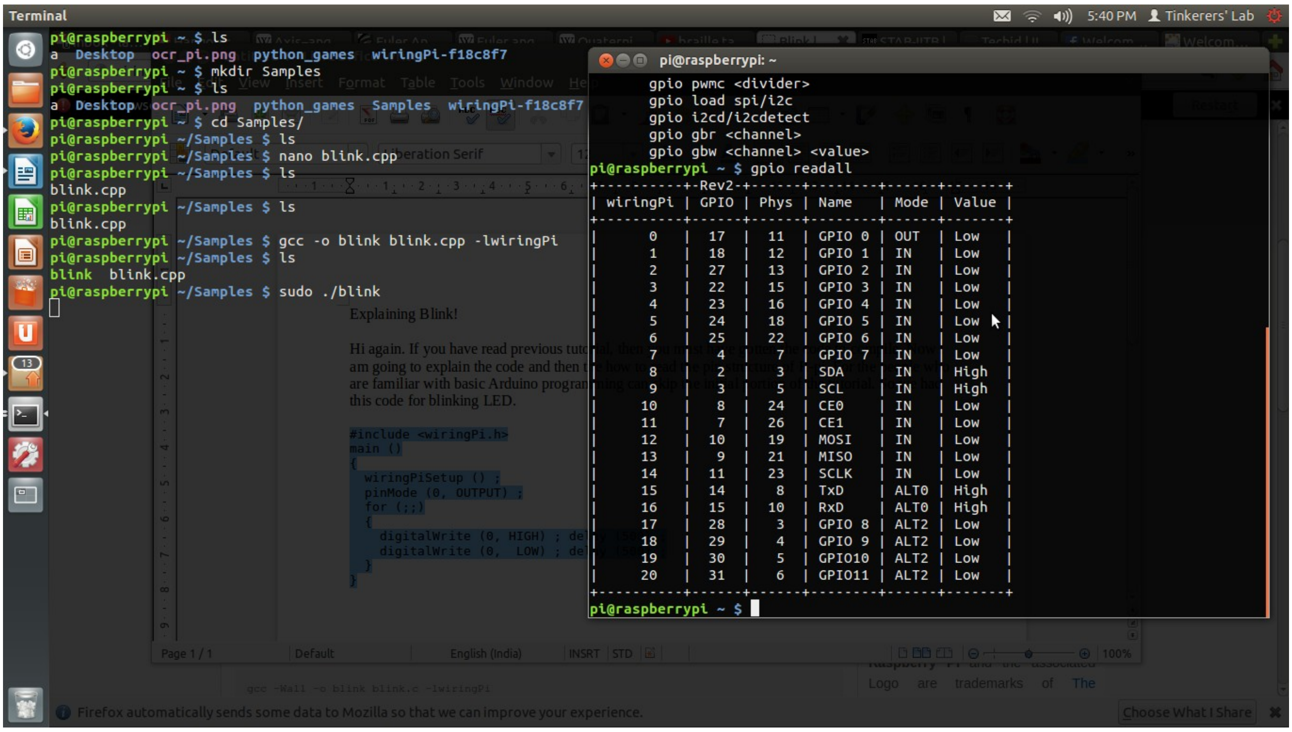
Explaining Blink!

Hi again. If you have read previous tutorial, then you must have gotten the code to compile. Now I am going to explain the code and then the how to read the pin structure of R-pi. For the people who are familiar with basic Arduino programming can skip the final portion of the tutorial. So we had this code for blinking LED.

```
#include <wiringPi.h>
main ()
{
  wiringPiSetup () ;
  pinMode (0, OUTPUT) ;
  for (;;)
  {
    digitalWrite (0, HIGH) ; delay (500) ;
    digitalWrite (0,  LOW) ; delay (500) ;
  }
}
```

Let us execute it. You must have done till here, and in seperate terminal open up gpio readall.



```
pi@raspberrypi ~ $ ls
a  Desktop  ocr_pi.png  python_games  wiringPi-f18c8f7
pi@raspberrypi ~ $ mkdir Samples
pi@raspberrypi ~ $ ls
a  Desktop  ocr_pi.png  python_games  Samples  wiringPi-f18c8f7
pi@raspberrypi ~ $ cd Samples/
pi@raspberrypi ~/Samples $ ls
pi@raspberrypi ~/Samples $ nano blink.cpp
pi@raspberrypi ~/Samples $ ls
blink.cpp
pi@raspberrypi ~/Samples $ ls
blink.cpp
pi@raspberrypi ~/Samples $ gcc -o blink blink.cpp -lwiringPi
pi@raspberrypi ~/Samples $ ls
blink  blink.cpp
pi@raspberrypi ~/Samples $ sudo ./blink
```

```
       gpio pwmc <divider>
       gpio load spi/i2c
       gpio i2cd/i2cdetect
       gpio gbr <channel>
       gpio gbw <channel> <value>
pi@raspberrypi ~ $ gpio readall
+----------+-Rev2-+------+--------+------+-------+
| wiringPi | GPIO | Phys | Name   | Mode | Value |
+----------+------+------+--------+------+-------+
|      0   |  17  |  11  | GPIO 0 | OUT  | Low   |
|      1   |  18  |  12  | GPIO 1 | IN   | Low   |
|      2   |  27  |  13  | GPIO 2 | IN   | Low   |
|      3   |  22  |  15  | GPIO 3 | IN   | Low   |
|      4   |  23  |  16  | GPIO 4 | IN   | Low   |
|      5   |  24  |  18  | GPIO 5 | IN   | Low   |
|      6   |  25  |  22  | GPIO 6 | IN   | Low   |
|      7   |   4  |   7  | GPIO 7 | IN   | Low   |
|      8   |   2  |   3  | SDA    | IN   | High  |
|      9   |   3  |   5  | SCL    | IN   | High  |
|     10   |   8  |  24  | CE0    | IN   | Low   |
|     11   |   7  |  26  | CE1    | IN   | Low   |
|     12   |  10  |  19  | MOSI   | IN   | Low   |
|     13   |   9  |  21  | MISO   | IN   | Low   |
|     14   |  11  |  23  | SCLK   | IN   | Low   |
|     15   |  14  |   8  | TxD    | ALT0 | High  |
|     16   |  15  |  10  | RxD    | ALT0 | High  |
|     17   |  28  |   3  | GPIO 8 | ALT2 | Low   |
|     18   |  29  |   4  | GPIO 9 | ALT2 | Low   |
|     19   |  30  |   5  | GPIO10 | ALT2 | Low   |
|     20   |  31  |   6  | GPIO11 | ALT2 | Low   |
+----------+------+------+--------+------+-------+
pi@raspberrypi ~ $
```

Firefox automatically sends some data to Mozilla so that we can improve your experience.

As you can see on the pin out you need GPIO 0 as in your code, that corrosponds to the pin number 11. Counting is alternate. Attach an LED across pin11 and ground, and it should blink! Voila.

Now let me explain you the code.
1) You have to include the wiringPi.h library in all your codes as it contains all the commands linking your code to an hardware output.

2) wiringPiSetup() This initialises ***wiringPi*** and assumes that the calling program is going to be using the ***wiringPi*** pin numbering scheme. This is a simplified numbering scheme which provides a mapping from virtual pin numbers 0 through 16 to the real underlying Broadcom GPIO pin numbers.

3) All other command are self-explanatory, pinMode sets the pin for OUTPUT or INPUT, digitalWrite write the logic 1 or 0 to the pin. Delay gives the delay in milliseconds.

Thats it, we will cover more topics in the coming tutorials.
Till then something to explore, wiringPi has an excellent GPIO utility, which allows you to do many things just using simple terminal commands. Here is the link
http://wiringpi.com/the-gpio-utility/

Riddhish Bhalodia (sophomore EE at IITB)