Interfacing the Camera and Gimbal system with the R-Pi

e-Yantra Team

July 5, 2016

Contents

1 Interfacing the Camera and Gimbal system with the R-Pi

Objective of this tutorial is to interface the desired camera(PiCam/USB Camera)and the Gimbal system(Servos) with the Raspberry-Pi and install the essential packages on the R-Pi board

2 Prerequisites

- A RaspbianOS installed Raspberry-Pi
- Python and Open CV installed on the R-Pi

3 Hardware Requirement

- Raspberry-Pi B+ Development Board
- Raspberry-Pi Camera Module(Pi Camera)
- Ethernet Cable/Wifi Mdoule
- Tower-Pro Micro Servos 9gm(2 No.)
- 3-D printed Camera fixing mount
- Connecting wires
- External power supply(if needed)

4 Software Requirement

- OpenCV 2.4.1
- Python 2.7
- MobaXterm 9.0

5 Setup

5.1 Setting up Raspberry-Pi

We will be using the Raspberry-Pi remotely from our Laptop/PC using SSH connection by MobaXterm using a LAN Cable or by a Wi-fi module that is setup on the Pi.Make sure that the Pi is connected to the Internet as we need to install necessary packages for further setup.

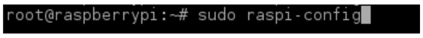
In our case,we connected to the Pi through LAN by assigning it a static IP from the "cmdline.txt" file on the SD card.We are accessing the Internet through the Wi-fi module.

It is also possible to overclock the R-Pi externally through the "config.txt" file from the SD card or internally through the "/boot/config" file.The need for overclocking is to meet the excessive processing needs of our Gimbal system.(Although it is not recommended as it may cause heating problems).

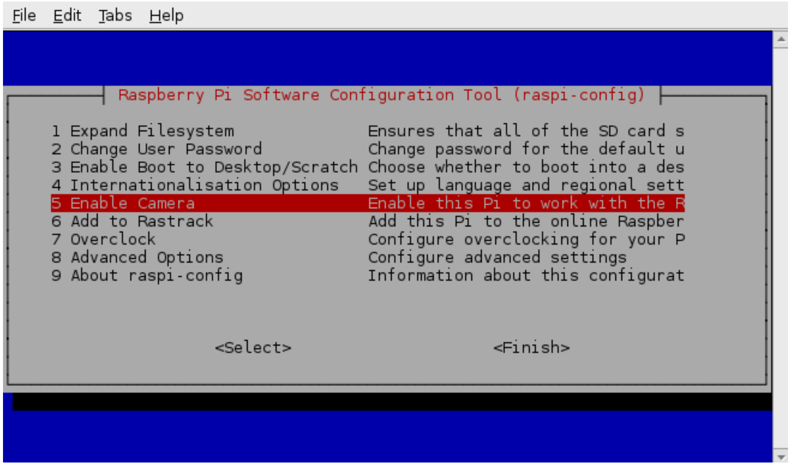
In our case,we overclocked the R-Pi from 700 Mhz(default) to 900 Mhz.

5.2 Setting up Camera(Pi-Cam)

After fixing the Pi-cam in its connector located between the Ethernet port and HDMI port,boot up your Raspberry-Pi and open the terminal and run the following command and enable the camera(if not enabled) and reboot ur R-Pi.



```
root@raspberrypi:~# sudo raspi-config
```



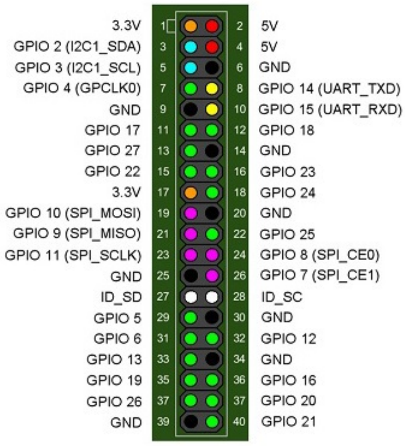
```
Raspberry Pi Software Configuration Tool (raspi-config)

 1 Expand Filesystem                Ensures that all of the SD card s
 2 Change User Password             Change password for the default u
 3 Enable Boot to Desktop/Scratch   Choose whether to boot into a des
 4 Internationalisation Options     Set up language and regional sett
 5 Enable Camera                    Enable this Pi to work with the R
 6 Add to Rastrack                  Add this Pi to the online Raspber
 7 Overclock                        Configure overclocking for your P
 8 Advanced Options                 Configure advanced settings
 9 About raspi-config               Information about this configurat

                <Select>                      <Finish>
```

We can check the working status of the camera by using "raspistill -o image.jpg" which takes a picture with the camera and stores it in image.jpg.

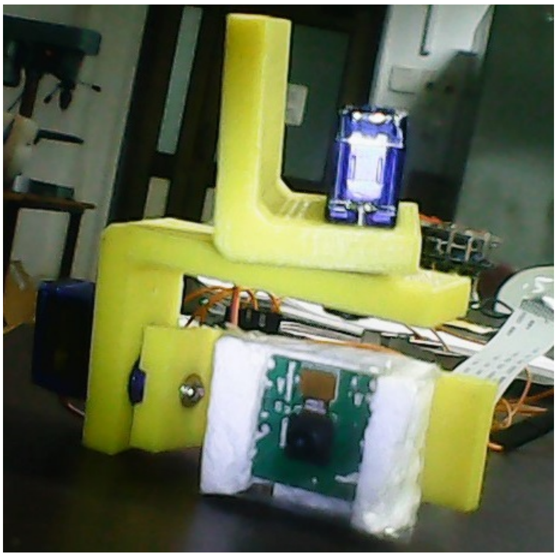
5.3 Setting up the Pan-Tilt System

For making the Pan-Tilt system we will be using Two Tower Pro Servo Motors each weighing 9g (we used micro servos as they would draw a little amount of current which can be produced by the R-Pi itself)

We connected the first servo to the Pin 16(GPIO 23) and second servo to Pin 18(GPIO 24) on the board.We use this pins as signal channels when we use the servos from R-Pi.All the available GPIO ports that can be used are shown below.



We calibrated each of the motor's PWM values(explained in the later section) and assembled to the frame mount which is shown below.



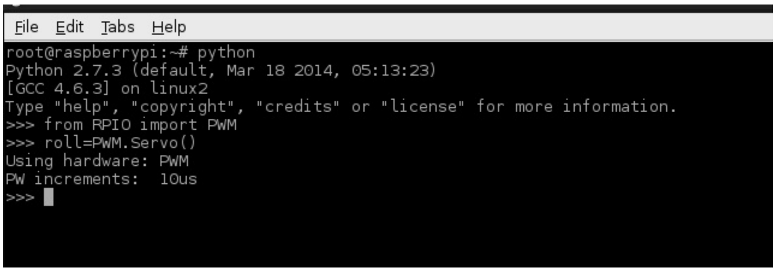
6 Necessary Packages

These are the packages that need to be installed in python for using the Pan-Tilt system.

- OpenCV
- picamera
- picamera-array
- RPIO

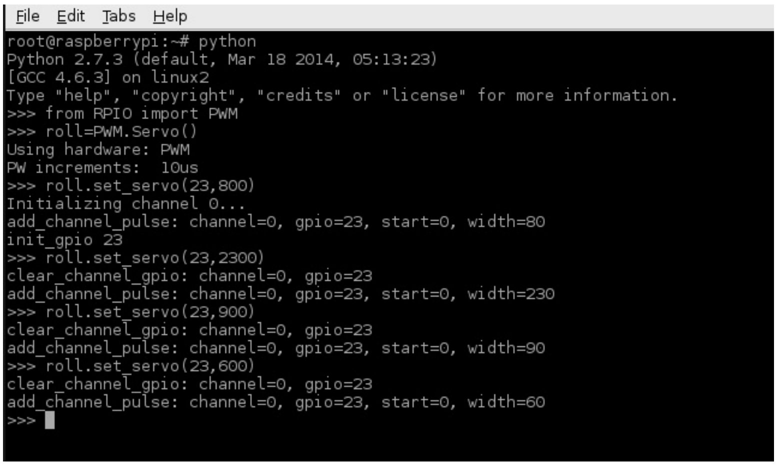
7 Calibrating the Servos

After installing the RPIO package we can calibrate the servos accordingly.Open the Terminal and run Python and import RPIO package and run the following.



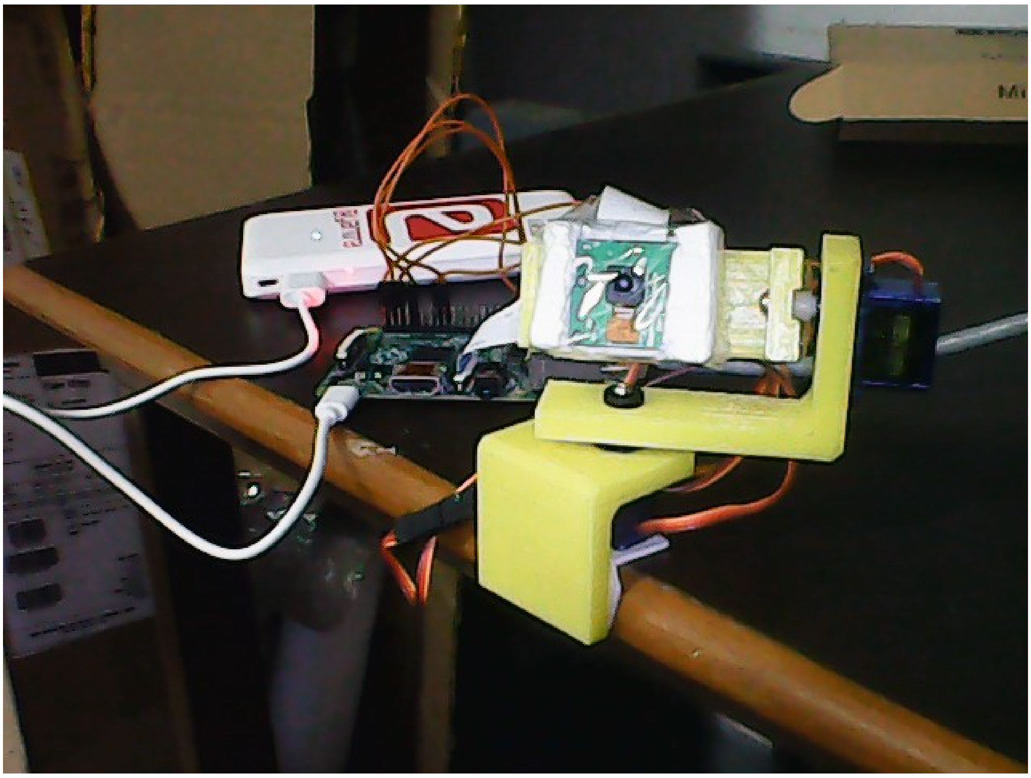
```
root@raspberrypi:~# python
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from RPIO import PWM
>>> roll=PWM.Servo()
Using hardware: PWM
PW increments:  10us
>>> 
```

We check the angle of the servo of at different values of PWM given to it.In our case we got 0 degree between 500 and 600 and 180 degree between 2300 and 2400.After some calibrations we got 90 degree at 1520.



```
root@raspberrypi:~# python
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from RPIO import PWM
>>> roll=PWM.Servo()
Using hardware: PWM
PW increments:  10us
>>> roll.set_servo(23,800)
Initializing channel 0...
add_channel_pulse: channel=0, gpio=23, start=0, width=80
init_gpio 23
>>> roll.set_servo(23,2300)
clear_channel_gpio: channel=0, gpio=23
add_channel_pulse: channel=0, gpio=23, start=0, width=230
>>> roll.set_servo(23,900)
clear_channel_gpio: channel=0, gpio=23
add_channel_pulse: channel=0, gpio=23, start=0, width=90
>>> roll.set_servo(23,600)
clear_channel_gpio: channel=0, gpio=23
add_channel_pulse: channel=0, gpio=23, start=0, width=60
>>> 
```

After the calibration,we assembled the Servos at 90 degree to the frame and making it a Pan-Tilt System.Here we used an external Powerbank(2 Amp) as a power supply to the R-pi.(Current requirement of the servos is more)



8 References

- https://www.raspberrypi.org/help/camera-module-setup/
- https://thepihut.com/blogs/raspberry-pi-tutorials/16021420-how-to-install_use-the-raspberry-pi-camera
- http://cmucam.org/projects/cmucam5/wiki/Assembling_pantilt-Mechanism
- https://www.youtube.com/watch?v=2LltXMHVuDs
- https://www.raspberrypi.org/documentation/usage/gpio/
- https://www.raspberrypi.org/blog/using-the-gpio/