Servo camera (Robot Tank)

Conestoga College Institute of Technology And Advanced Learning

Bhushan Gajanan Dandekar, Kaviya Ann Kuncheria, Rohit Gidda

bdandekar6611@conestogac.on, kkuncheria1647@conestogac.on, rgidda4058@conestogac.on

August 07, 2019

Abstract

This project introduces Raspberry Pi Board Technology to implementing Servo Camera for Robot Tank. Sponsor's main aim is to modify the current prototype into a smaller one in a cost effective way. This project is split by sponsor into two parts where we are allocated to work on the movement of the servo camera and its software. In order to eliminate this, the present prototype seems to send more energy to the servos, we will control the input power supply to servo motors. The speed of the Servo motor and also the pan and tilt angle must be reduced during the movements.

I. Introduction

This is a project based on industrial development. This project concept was provided by the sponsor to improve further on it. Their primary goal is to cost-effectively change the present prototype into a smaller one. It can therefore be readily accessible to everyone. We are accountable for creating your servo camera segment in this capstone project. We chose to improvise the motion of the servo pan and tilt.

It consists integration working of our project, in which we include testing for servo motor, Pi-camera and raspberry pi together. Using which we are going to capture images using Pi-camera in various angles. In this Raspberry Pi-camera will be mounted on the servo camera mount through we can make the pan and tilt movement of the camera so that it can able us to capture image in various angles.

II. Proposed solution

We can eliminate the power issue by connecting discharging capacitor across power connection if this doesn't work then we will connect discharging capacitor with register across to the power supply of servo motor, by this may we can eliminate the problem of vibration. To keep the motor at initial position, by eliminating the first problem we can overcome our second problem if not then, we have to do some mathematic calculations and implement into program after solving, first two issues we will get rid of third issue.

To Implementing Pi camera library, we will search for the Pi camera library on google and will include that library into our code and will fix the camera issue. For PWM code, we will try some different duty cycle calculations and implement it in our project then we will see which one suitable for all servo motors. For camera placement, firstly we will use double-sided tape for placing the camera on mount if this doesn't work then we will do some alteration for the camera mount.

III. Key components

i. Raspberry Pi 3 Model B

Figure 1: *Raspberry Pi 3 Model B*

The Raspberry Pi 3 Model B is the third level of Raspberry Pi. The main feature of this level is to measure single board PC can be utilized for some applications and overrides the first Raspberry Pi Model B+ and Raspberry Pi 2 Model B. While keeping up the prominent board position the Raspberry Pi 3 Model B presents to you an all the more dominant processor, 10x quicker than the original Raspberry Pi. Furthermore, it includes remote LAN and Bluetooth availability making it the perfect answer for incredible associated plans.

ii. Servo Motor

Figure 2: *Servo Motor*

A servo motor is a rotary actuator or engine enabling accurate control in terms of angular position, acceleration and speed, capabilities that a regular motor does not have. It uses a regular motor and combines it with a sensor for feedback on the situation.

It is small and lightweight with high power output. This servo can rotate about 180 degrees (90 in each direction) and operates as small as the normal types. To regulate these servos, you can use any servo code, software or library. It comes with 3 horns (arms) and hardware.

iii. Servo Camera Mount

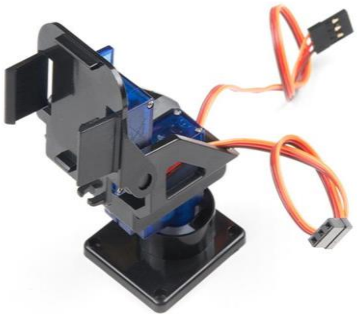


Figure 3: *Servo Camera Mount*

This is an assembler of pan/tilt that includes two servos to move to every sides of the axis and the camera is attached to the top bracket. Its control position and speed very precisely. Now a potentiometer can sense the mechanical position of the shaft. Hence it couples with the motor shaft through gears.

The current position of the shaft is converted into electrical signal by potentiometer, and is compared with the command input signal. In modern servo motors, sensor sense the position of the shaft.

iv. Raspberry Pi Camera

Figure 4: *Raspberry Pi camera*

The camera board attaches to the Raspberry Pi via ribbon cable. One end of the ribbon cable goes to the camera PCB and the other end to Raspberry Pi hardware itself. The Pi Camera module sensor has 5 megapixel native resolution in still capture mode. In video mode it supports capture resolution up to 1080p at 30 frames per second.

v. Portable Battery

Figure 5: Portable Battery

This is 6700 mAh portable charge using for power supply. It has ultra-reliable A+ lithium Ion battery. It prevents short circuit and also over current protection. The battery pack will automatically shut down if a short circuit or overload occurs.

IV. Methodology

High Level Architectural Block Diagram
The figure 6 shows the high level architectural block diagram

The block diagram containing following blocks

1. Raspberry Pi
2. Servo motor
3. Memory Card
4. Wi Fi Router
5. Power Bank
6. Monitor
7. Mobile
8. Raspberry Pi Camera

In this the Raspberry Pi camera is connected to the Raspberry Pi via Camera ribbon. That camera is place on to the Servo camera mount. In that servo camera mount the couple of servo motors are attached in it that will able to move the mount in pan and tilt.
Pan and tilt movement will move into 0 to180 degree and it will in 90 in each direction.

The camera surveillance will be displayed on the camera as well as on the mobile. The mobile and Raspberry Pi is wirelessly connected through the same wi-fi router. Also the recorded video and captured image will be stored in to the memory card which is attach to the Raspberry Pi.

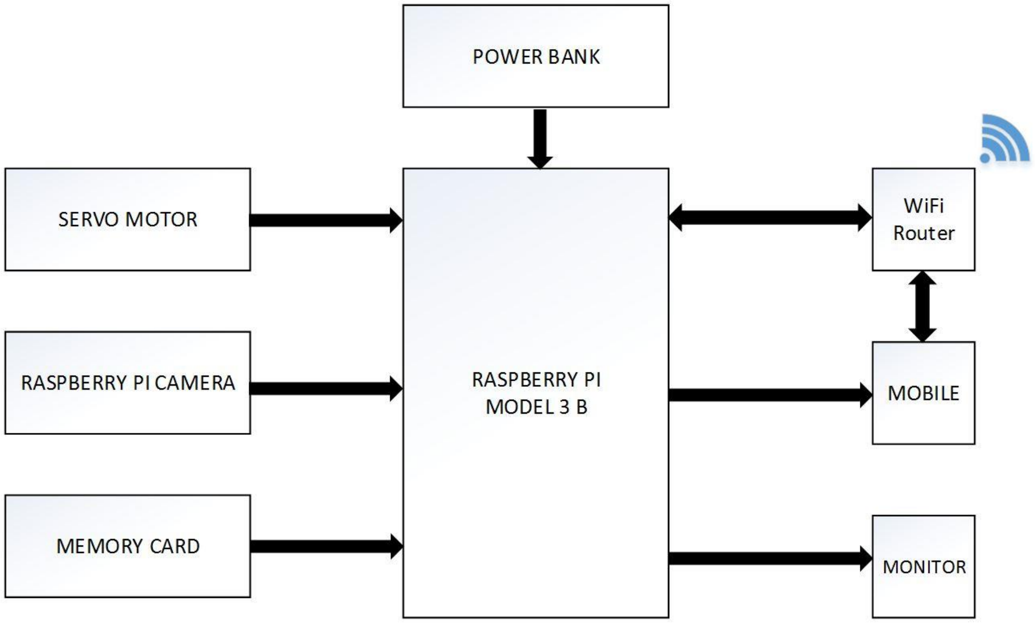


Figure 6: *High level architectural block diagram*

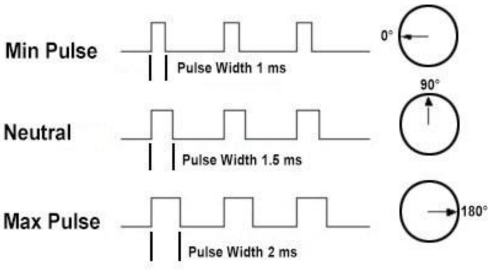
i. Testing of Servo Motor

In this we are using Servo Motor SG90 which rotated 90 degree in each directions. So to test and verify that we have measured the servo motor in various angles and we have got the output angles. To measure the angle we have printed protractor scale and following are output images.

Minimum pulse:
Pulse width is 1ms which is 0 degree, i.e., around 5% (1/20*100=5)
Which drives motor to right side

Neutral pulse:
Pulse width is 1.5ms which is 90 degree, i.e., around 7.5% (1.5/20*100=7.5)
Which drives motor to center

Maximum pulse:
Pulse width is 2ms which is 90 degree, i.e., around 10% (2/20*100=10)
Which drives motor left



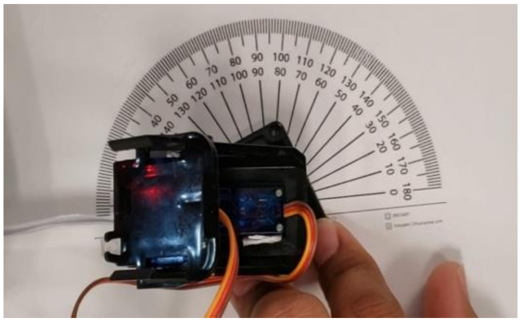


Figure 7: *Motor Directed toward 0 Degree*

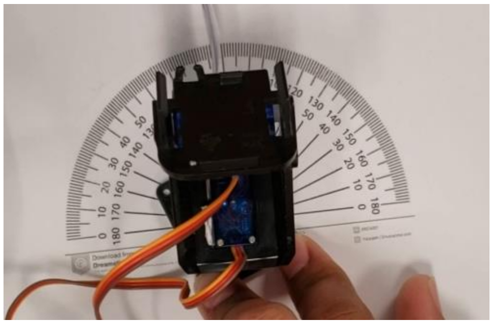


Figure 8: *Motor Directed toward 90 Degree*

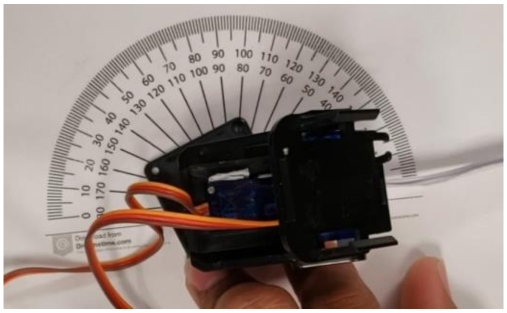
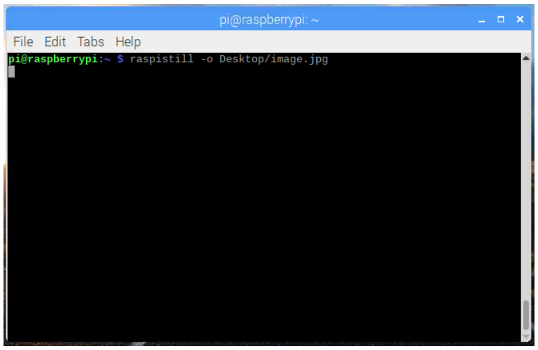


Figure 9: *Motor Directed toward 180 Degree*

ii. Testing of Raspberry Pi Camera

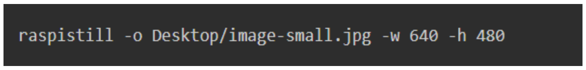
To test the working of camera we have tested it using the simple code to check whether the camera is working or not. For that we have tested the following syntax.



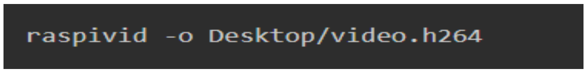
```
pi@raspberrypi:~ $ raspistill -o Desktop/image.jpg
```

Figure 10: *Implementation of Sample Camera syntax*

The above image shows the implementation of simple camera syntax for Raspberry Pi. For that we have tested syntax for capturing the image through camera also record video using Pi camera. The syntax are as follows.

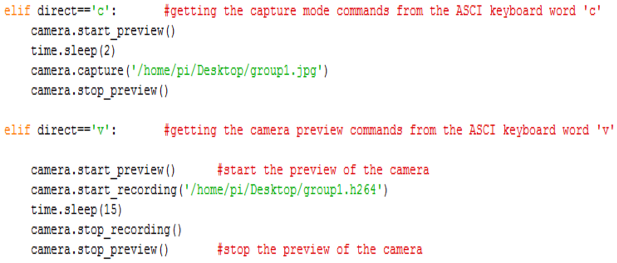


```
raspistill -o Desktop/image-small.jpg -w 640 -h 480
```



```
raspivid -o Desktop/video.h264
```

After testing this syntax we have implemented into the code and tried to compile that. The screen shot of the code is as follows. To capturing image and recording video we assigned key to it. For capturing image we have assigned "C" and to record video we have assigned "V"



```
elif direct=='c':       #getting the capture mode commands from the ASCI keyboard word 'c'
    camera.start_preview()
    time.sleep(2)
    camera.capture('/home/pi/Desktop/group1.jpg')
    camera.stop_preview()

elif direct=='v':       #getting the camera preview commands from the ASCI keyboard word 'v'

    camera.start_preview()      #start the preview of the camera
    camera.start_recording('/home/pi/Desktop/group1.h264')
    time.sleep(15)
    camera.stop_recording()
    camera.stop_preview()       #stop the preview of the camera
```

Figure 11: *Sample Code*

V. Implementation

We mounted the Raspberry pi and servo motor on to the PCB that was given by group C. We have used small standoff (pillar) for raspberry pi because if we use longer it will not allow the movement of servo motor.

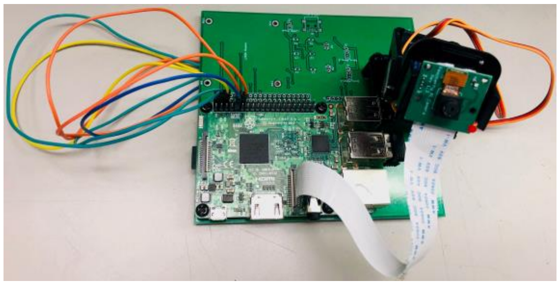
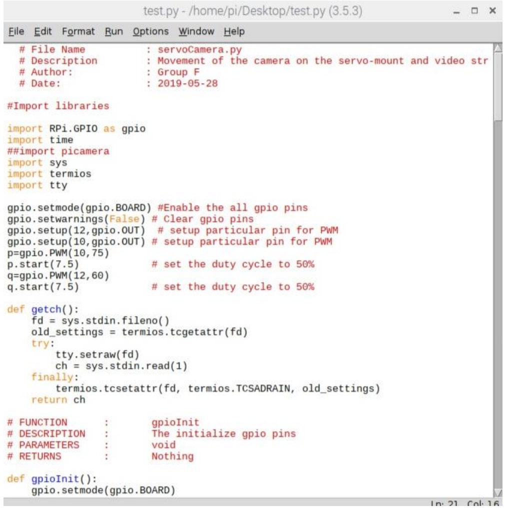


Figure 12: *Implemented of Hardware*

For camera we have use double sided tape so that it can stick properly to the servo motor and it holds the camera tightly. Also while rotating in different angles the camera ribbon doesn't get stuck into any component or with the jumper wires.

The jumper wires are used to do the connection between the servomotor and raspberry pi. By using this we are giving power supply directly through Raspberry Pi. Also we have used long jumper cable so that it could not affect the pan movement of servo.


test.py - /home/pi/Desktop/test.py (3.5.3)
File Edit Format Run Options Window Help
Ln: 21 Col: 16

```
  # File Name          : servoCamera.py
  # Description        : Movement of the camera on the servo-mount and video str
  # Author:            : Group F
  # Date:              : 2019-05-28

#Import libraries

import RPi.GPIO as gpio
import time
##import picamera
import sys
import termios
import tty

gpio.setmode(gpio.BOARD) #Enable the all gpio pins
gpio.setwarnings(False) # Clear gpio pins
gpio.setup(12,gpio.OUT)  # setup particular pin for PWM
gpio.setup(10,gpio.OUT) # setup particular pin for PWM
p=gpio.PWM(10,75)
p.start(7.5)            # set the duty cycle to 50%
q=gpio.PWM(12,60)
q.start(7.5)            # set the duty cycle to 50%

def getch():
    fd = sys.stdin.fileno()
    old_settings = termios.tcgetattr(fd)
    try:
        tty.setraw(fd)
        ch = sys.stdin.read(1)
    finally:
        termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
    return ch

# FUNCTION      :       gpioInit
# DESCRIPTION   :       The initialize gpio pins
# PARAMETERS    :       void
# RETURNS       :       Nothing

def gpioInit():
    gpio.setmode(gpio.BOARD)
```

Figure 13: *Implemented Code*

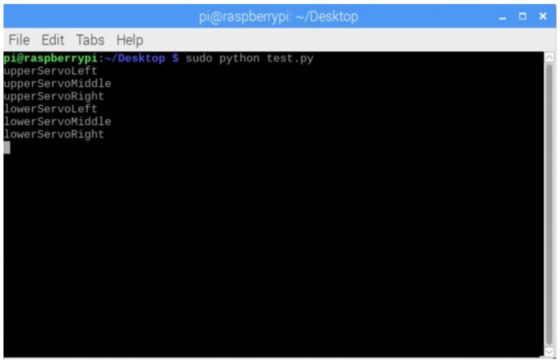
Firstly, it imports the libraries and then initialize the general purpose input /output (GPIO) pins and enable these pins. Set the PWM (Pulse width modulation) with 50% duty cycle. When the code reaches to Exception handling then the servo camera mount on the robot tank and the movement of robot tank are controlled through keyboard. The servo camera movements are controlled through the following keys.

Keyboard Inputs	Function
I	Upper servo left
O	Upper servo middle
P	Upper servo right
J	Lower servo left
K	Lower servo middle
L	Lower servo right
C	Capture image
B	Reset the working
V	Preview image

Figure 14: *Keyboard Input and Functions*

All these ASCII character work through the function by calling it and then compiler starts to search the function to perform their specific task as describe inside the functions. So, for forward function: servomotors stars rotating forward direction also for backward function: it starts rotating towards backward, for left function: turning left and for right function: turning right for the robot tank.

For the camera, its movements is to some particular angle: lower servo (0°-180°) and for upper servo (0°-180°) same. If ‘c’ ASCII key press the cameras capture mode is ON. If ‘v’ ASCII key press the cameras starts the preview of the video it capture recently. Then compiler moves to the except command for the Keyboard Interrupt for besides these key if any key process it interrupts the working of the robot tank and the servo camera.



```
pi@raspberrypi:~/Desktop $ sudo python test.py
upperServoLeft
upperServoMiddle
upperServoRight
lowerServoLeft
lowerServoMiddle
lowerServoRight
```

Figure 15: *Code Output*

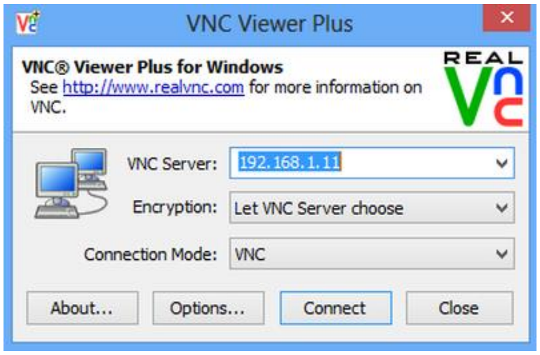
The Servo motor Movement for various angles are shown below. They are for Pan and Tilt. In Pan it is doing Left and right movement using bottom servo motor and in tilt movement it is using upper servo motor which will do the UP and Down movement.

VI. VNC Viewer

In computing, Virtual Network Computing (VNC) is a graphical desktop-sharing system that uses the Remote Frame Buffer protocol (RFB) to remotely control another computer. It transmits the keyboard and mouse events from one computer to another, relaying the graphical-screen updates back in the other direction, over a network.

VNC is platform-independent – there are clients and servers for many GUI-based operating systems and for Java. Multiple clients may connect to a VNC server at the same time. Popular uses for this technology include remote technical support and accessing files on one's work computer from one's home computer, or vice versa.

VNC was originally developed at the Olivetti & Oracle Research Lab in Cambridge, United Kingdom. The original VNC source code and many modern derivatives are open source under the GNU General Public License.



VNC® Viewer Plus for Windows
See http://www.realvnc.com for more information on VNC.

VNC Server: 192.168.1.11
Encryption: Let VNC Server choose
Connection Mode: VNC

Figure 16: *VNC Connection*

VII. Result

The Raspberry pi and servo motor were installed onto the PCB. We have used tiny pillar (standoff) for raspberry pi, as it does not enable moving the servo motor if used taller.

For Connecting Raspberry Pi camera to the Raspberry board we have used the ribbon. It is 20 cm in length.

The Servo motor Movement for various angles are shown below. They are for Pan and Tilt. In Pan it is doing Left and right movement using bottom servo motor and in tilt movement it is using upper servo motor which will do the UP and Down movement.

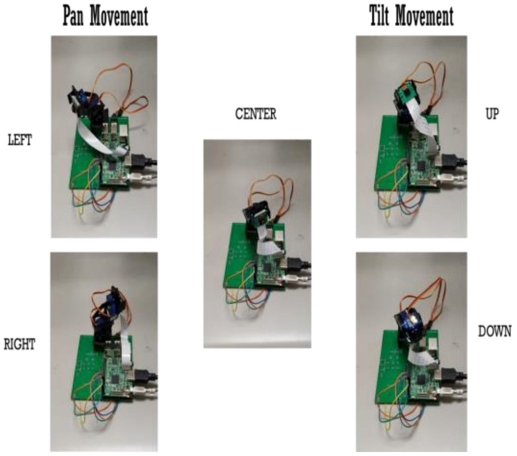


Figure 17: *Servo Movement Outputs*

In this through VNC (Virtual Network Computing) we have connected mobile to Raspberry Pi. We are communicating to Raspberry Pi wirelessly. Also using mobile we can control servo motor movements and also can do Live video surveillance. The final assembly of the project is shown in the following Figure 18. It shows the live streaming on the mobile in that figure.

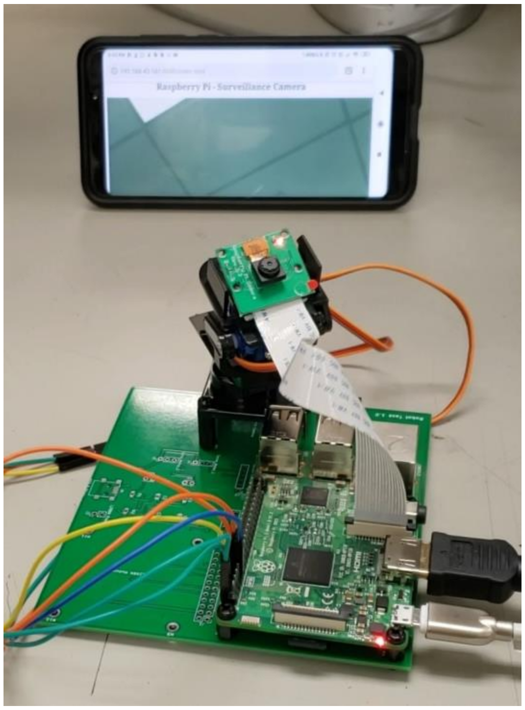


Figure 18: *Assembly of Hardware with Live Streaming*

VIII. Recommendations

In the future development we are going to Replace the current Raspberry Pi 3 with Raspberry Pi 4 which works faster than the Current Pi and have more memory to it. Also we are going to replace Pi camera with Smraza Camera Module for Raspberry Pi 4 Which has night vision mode and have LED flash to it.

IX. Acknowledgment

We would like to express our special thanks of gratitude to Our Supervisor Prof. So-Ra Chung, Ph.D. as well as our Sponsor Robert Elder who gave us the golden opportunity to do this wonderful project on the topic Robot Tank Servo Camera, which also helped us in doing a lot of Research and we came to know about so many new things we are really thankful to them.

References

http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

https://www.banggood.com/Raspberry-Pi-3-Model-B-ARM-Cortex-A53-CPU-1_2GHz-64-Bit-Quad-Core-1GB-RAM-10-Times-B-p-1041862.html?cur_warehouse=USA

https://elmwoodelectronics.ca/products/14391?variant=44983838403&utm_campaign=gs-2019-02-19&utm_source=google&utm_medium=smart_campaign&gclid=Cj0KCQjww47nBRDlARIsAEJ34blDnlAS3v1svdTqj6NBpfrhDpI0DDyFSTCyTPrM3iiz09P6L2edMgEaAii9EALw_wcB

https://www.sparkfun.com/products/14028

https://www.ravpower.com/p/Ravpower-6700mAh-Portable-Charger.html

http://blog.robertelder.org/raspberry-pi-streaming-video-tank/