GESTURE BASED UI

Robot Car Movement with Myo

Abstract

For the Above mentioned module we were required to produce the project by using any gesture/voice based technology of hardware along the software. This was to demonstrate our understanding of embedded programming and as well the 4 years of understanding of software development

RAJA NASEER AHMED KHAN

Samson_knight@yahoo.com

Table of Contents

Introduction:

I was required to develop a project by using gesture or voice to control hardware devices or software objects. Hence, I have chosen the project of moving a robot car using the hand gestures. Following are the devices/software needed.

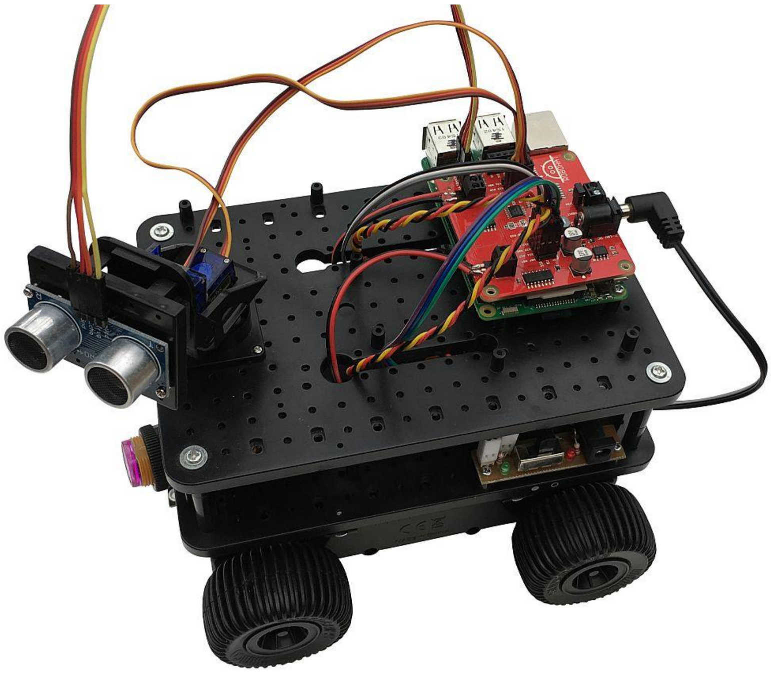
(Important) Make sure python version (2.7)is the version used, otherwise might get conflict and code won't run. Also let myo armband heat up to work.

Required Hardware

4tronix inition robot car chasis - Website here

Raspberry pi 3 Model B V1.2 - Website here

Robohat V1.7 - Website here



Myo Armband - Website here

Required Software

Rasbian OS for Raspberry pi- Website here

Python to run scripts

Direct Bluetooth connection between Linux and Myo armband using Bluegiga BGAPI/BGLib

Tested in a Raspberry Pi. video: https://www.youtube.com/watch?v=AWxEL4l4Fg8

Pre-requirements for your Myo device

- Firmware version 1.3.1448 or higher
- It is necessary to calibrate your Myo device using the official software.
- Use the Myo dongle Bluetooth

Requirements

- python >=2.6
- pySerial
- enum34

Execute Sample

The Myo dongle Bluetooth must be connected.

- Open the console "terminal"
- Go to project folder "project run" and run following command
- Execute this command:

-
  ```
  python project.py
  ```

Installations and Testing

1. Install Rasbian OS into Raspberry Pi.

   i. Click here and follow instruction. Rasbian

2. Get Myo Armband Setup using PyoConnect Software for Raspberry Pi.

   i. Click here and follow instruction. PyoConnect

3. Using the python version -- run the scripts on command line i.e. python project.py.

4. Wear the Myo armband on any of arm just below elbow.

   i. Click here for learn how to use Myo. Myo

5. Do the Following Gesture to move the robot car.

   i. Fist is to move backward
   ii. Open Hand to move forward
   iii. Wave Left to move left
   iv. Wave Right to move right
   v. Relaxing hand to stop car

Code Explaination

1. First is to connect Myo with Pi, Follow the code at http://www.fernandocosentino.net/pyoconnect/) also you can get here all the code to detect the poses.

2. Setup The carFunctions into the python file by following code

   i. import carFunctions

      a. this line gives access to roboHat libraries, which has the options to control 2 motors for running car all around, also i have fitted the ultrasonic sensor to detect the distance from an object.

         a. Following are the functions to move car

            a. carFunctions.stop() for stopping car
            b. carFunctions.forward() moving forward
            c. carFunctions.reverse() to reverse
            d. carFunctions.spinRight() to turn right
            e. carFunctions.spinLeft() to turn Left

         b. Following is the functions made from myo file downloaded from pyoconnect to detect pose

            a. def on_Pose(pose): if pose == pose.REST: elif pose == pose.FIST: elif pose == pose.WAVE_OUT: elif pose == pose.WAVE_IN: elif pose == pose.FINGERS_SPREAD:

         c. After Combining above 2 we can move car with gesture from Myo Armband

            a. def on_pose(pose): if pose == pose.REST: carFunctions.stop() elif pose == pose.FIST: carFunctions.reverse() elif pose == pose.WAVE_OUT: carFunctions.spinLeft() elif pose == pose.WAVE_IN: carFunctions.spinRight() elif pose == pose.FINGERS_SPREAD: carFunctions.forward()

         d. We can get distance from the ultrasonic sensor by using carFunctions.getDistance() and combining this with if statement in pose = Fist then get distance and distance less than 20 cms then turn right and continue until distance is less than 20 cms.

Files and Their Details

1. run this for project.py = has main function to run and starting Myo device to pass on values to car.
2. /project run/poseDetect.py = is used to get the poses performed and return functions of car controls on detection of each pose.
3. /project run/myo_starter.py = is used to start listening to Myo armband.
4. /project run/carFunctions.py = giving access to functions to robohat circuit using low level interace in Raspbbery pi 3.
5. project run/gesture.py = enum giving access to pose types with numbers unique numbers assigned to each pose.
6. \modules\myo_vibrate.py = enum file for assigning numbers to vibration types
7. \modules\myo.py = has all the functions from myo libaries available for working with Myo (its downloaded from pyoconnect)
8. \modules\bluetooth.py = is giving access to bluetooth of Raspberry pi with code downloaded one of the reference.
9. \modules\myo_data.py = getting the structured function from utilities and returning data from my.
10. \modules\dataStruct.py = is defining structure of for some of the functions will be used fetch,pass and map data.

Issues During The Project

1. Shared the Pi and robohat with classmates, who burned down the pi and robohat, bought new robohat first, which was also defected and not supplying the current to dedicated motor connectors on roboHat. So, i utilized my electronics knowledge to change the one of the 2 diodes which were protecting the circuit on adapter connector and wire connector. hence, i used the direct wire connector which is shown by vin on robohat.
2. 2nd problem was the i had no TV, so couldn't connect the Pi to big screen, hence was using my mobile phone to connect to Raspberry desktop as remote. but for this option i had to connect PI to my own phone hotspot as IP address was assigned to PI from phone, and it wasn't changing all the time.
3. Corona Virus outbreak pandemic, colleges closed, no physical access to any material or hardware from college, but still got through the project.
4. Code for both of the above devices was hard to adopt as Myo is discontinued and Robohat is not being used to much. usually people can control low power consumption components with GPIO library and directly from PI board, but power output wasn't enough to run 2 motors of our robot car.

Conclusion

We have learned how to use our coding skills along with the electronic components to get bit glimpse of Embedded Programming, but this wasn't a true embedded programming as we have to go through the libraries to make Myo work, which has been discontinued product, Hence not much support was available. Real Robotics programming is similar where we control the output power of each terminal, which is then input in another function to compute the output of each desired terminal will signal, That will calculate the speed, rotation angle, and stop statements of the robot.

Recommendations

I would sincerely recommend to use the available gesture devices which are still in production, as there will be lots of support available and in all the languages would be supported to.

Authors

Raja Naseer Ahmed Khan Dated 1/03/2020

References

Myo Library File PyoConnect

4tronix inition robot car chasis - Website here

Rasberry pi 3 Model B V1.2 - Website here

Robohat V1.7 - Website here

Some other good tutorials for the project. Click here Tutorial

License