Final Project Progress Report
PerceptoBot
ECE558-Fall 2018

Submitted by- Shubham Vasaikar, Mrunal Hirve

Description

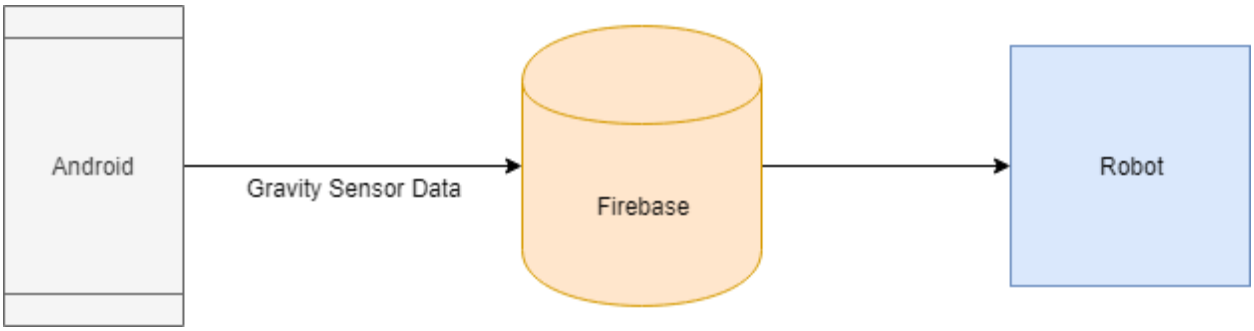
We are making a 2-wheeled robot that will be controlled via an Android app using movement sensors (accelerometer and gyroscope) on the phone.

Current progress

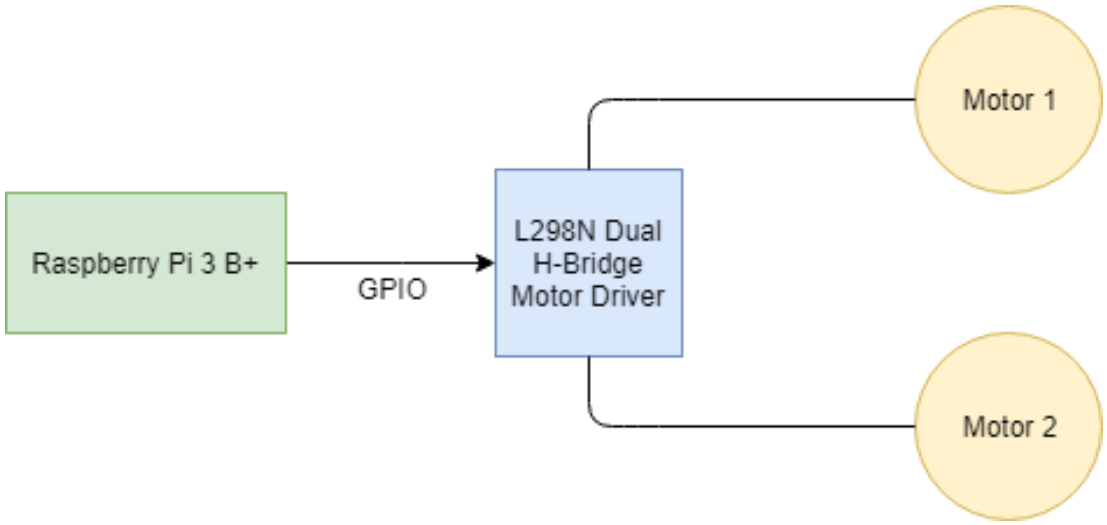
☑ Thursday 15 Nov 2018 – Assembled robot
☑ Thursday 22 Nov 2018 – Robot controlled via app
☐ Wednesday 5 Dec 2018 – Stretch Goals

Design

As a high-level overview of the current project, the app continually monitors and sends sensor values to the Firebase Realtime Database. The robot is listening on the sensor attributes in firebase and makes movements accordingly.

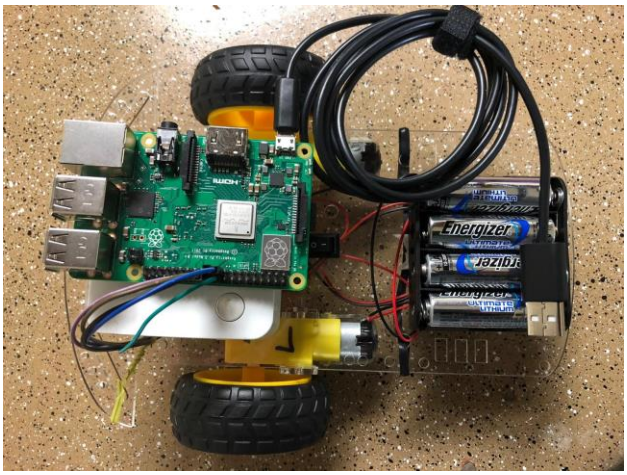


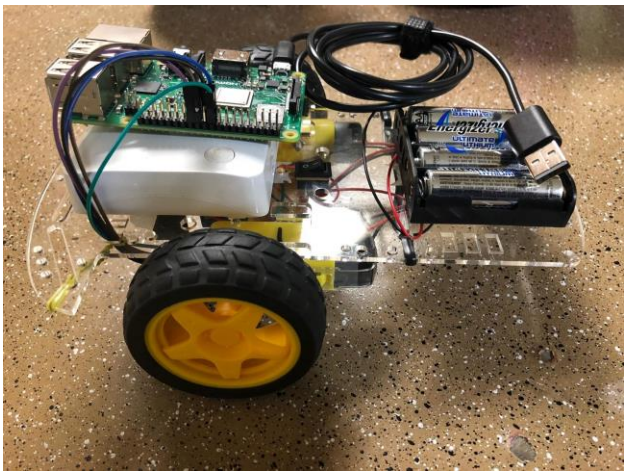
Hardware



Hardware Connections

RPI3B+	L298N		L298N	DC MOTOR
GPIO 7	IN 1		OUT 1	Vcc
GPIO 8	IN 2		OUT2	GND
GPIO 9	IN 3		OUT3	Vcc
GPIO 10	IN 4		OUT4	GND
GND	GND			





We are using a robot chassis to mount a Raspberry Pi 3 B+, 2 wheels, a motor driver, 2 DC Motors, and 2 battery packs (one for the RPi and one to power the motors. The RPi runs Rasbian OS and connects to the motor driver using the GPIO pins. The code that reads the sensor values and runs the motors is written in Python and uses the pyrebase module to listen for changes in firebase data.

```
def linear_handler(message): # Callback
    if (db.child("direction").get().val()):
        robot.forward(message['data'])  # Move the robot.
    else:
        robot.backward(message['data'])

# Listener that activates callback on data change
linear_stream = db.child("linearAcc").stream(linear_handler)
```

Software

To get the sensor data, the app needs to create an instance of sensor manager which allows to interface with the sensors.

```
mSensorManager = (SensorManager)getSystemService(SENSOR_SERVICE);
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY);
```

To listen for changes in sensor values, the activity also needs to implement the sensor event listener interface. This interface provides a callback – onSensorChanged() that we can implement.

```
public class MainActivity extends AppCompatActivity implements
SensorEventListener {

...

@Override
public void onSensorChanged(SensorEvent event) {
    float x, y; // x value is Linear Motion and y value is Lateral Motion
    x = event.values[0];
    if(x > 0 && x < 9) {
        myRef.child("linearAcc").setValue(1-(x/9));  // Send to Firebase
    } else {
        myRef.child("linearAcc").setValue(0);
    }

    y = event.values[1];
    if ((y < -1 && y > -6 ) || (y > 1 && y < 6 )) {
        myRef.child("lateralAcc").setValue(y);    // Send to Firebase
    } else {
        myRef.child("lateralAcc").setValue(0);

    }
}
```

Stretch Goals

We are currently working on:

- Streaming video from Pi Camera mounted on robot to the Android app so that the user can navigate without having a line-of-sight to the robot.
- Using the camera to detect faces and follow the movement of the faces.

Challenges Faced

Weight Imbalance of Robot

Earlier, we had attached both the battery packs over the rear wheel of the robot. This meant that there was too much weight over the rear and the front was very light. Due to this the front wheels could not gain enough traction to propel the robot in a straight line and the robot veered off into whichever direction the rear wheel was facing.

Python-Firebase Interface

Initially, we were planning on using the python-firebase python module to connect to firebase. However, we soon realized that this module did not provide methods to register listeners for firebase data values. Therefore, we decided to move on to pyrebase which provides stream listeners for firebase data values.

New Learning outcomes

Hardware

Motor drivers and H-bridge

Software

Interfacing with sensor values

References

https://github.com/thisbejim/Pyrebase#database
https://developer.android.com/guide/topics/sensors/sensors_motion#sensors-motion-grav
https://developer.android.com/reference/android/hardware/SensorListener
https://gpiozero.readthedocs.io/en/stable/recipes.html#robot