Hand Gesture Controlled Robotics Car (Doc Version 1.0)

Vasu Gupta (@ GizmoTinkers)

February 18, 2020

Contents

1 Document Version History

S.No	Version	Notes
1	1.0	First Version

Table 1: Document Version History

2 Project Description

Objective of this project is to design a robotics car controlled using hand gestures.

Building this project provides exposure to multiple design aspects including making a robo car and controlling it using a microcontroller (Arduino is used in this project), adding wireless connectivity to robots, wearable electronics, working with gyroscope and accelerometer sensor, communication between two microcontrollers over a bluetooth connection and much more. This project is also super fun to build and play. The concepts used here can easily be extended to design robots for helping with work in homes and offices, etc.

3 Bill of Materials

Below is a list of components, materials and tools required for building the robotics car and electronics on the car side i.e receiver side.

1. Arduino Uno Rev3 Board
2. L298N Motor Driver Module
3. HC-05 Bluetooth Module
4. Robo Car Materials
   (a) Robo Car Chassis
   (b) 2x DC Motors

(c) 2x Back Wheels, 1x Front Wheel (Castor Wheel)

(d) Nuts and Bolts for assembling chassis and attaching different components to it (usually come with the car chassis)

5. Breadboard, Resistors - 3x 1.5kOhm, Connecting wires - male-to-male, male-to-female

6. 2x Batteries(2x 3.7V 18650 Lithium Ion Battery), battery holder and battery charger

Below is a list of components, materials and tools required for building the hand gesture unit i.e transmitter for sending gesture based commands to the robotics car.

1. Arduino Lilypad

2. Programmer for Arduino Lilypad

3. Gyroscope and Accelerometer Module MPU6050

4. Bluetooth Module HC06

5. 3.7V Rechargable Lithium Ion Battery

6. Mini Breadboard, 1x ON/OFF Switch, Connecting wires - male-to-male, male-to-female, small cardboard base

7. Soldering iron and solder, wire clippers, double sided tape

4 Module Level Functionality

4.1 Arduino Uno Rev3 Board

This is the main board containing the microcontroller responsible for system level logic and control flow and interfacing with all the other modules like HC-05 bluetooth module and L298N Motor driver module.

4.2 L298N Motor Driver Module

This module is useful for controlling the motors to which the wheels are attached. In principle, we could also control the motors without L298N module by instead using transistors and diodes. However, using this module makes it easier to control the motors - we don't have to build and test our own transistor circuit. It is sometimes beneficial to use existing modules wherever appropriate so that we could focus on the overall system level product design. This module interfaces with Arduino which sends it commands for controlling the motors.

4.3 HC-05 Bluetooth Module

This module is used for bluetooth connectivity between Arduino and Smartphone. It is a really nice module and includes all the RF components like antenna, etc required for bluetooth communication and provides a very easy to use interface. Communication between Arduino and this module happens over a 2 wire UART Serial Interface.

This module powered by the 5V supply from Arduino in this project. However, the input-output signaling used in the module treats 3.3V as HIGH versus 5V as HIGH signaling used in Arduino UNO. For transmitting data from HC-05 to Arduino, nothing special needs to be done and a direct connection from TX of HC-05 to RX of Arduino can be used since Arduino also treats 3.3V as HIGH. However, for receving data from Arduino to HC-05, a simple resistive voltage divider circuit can be used so that when Arduino communicates a 5V high signal , the voltage level is reduced to 3.3V from 5V. This ensures HC-05 circuitry is protected from a high voltage of 5V.

4.4 HC-06 Bluetooth Module

HC-06 bluetooth module is similar to HC-05 bluetooth module. The main difference is that HC-05 can be configured to act either as a bluetooth master or slave whereas HC-06 can act as a bluetooth slave only. Therefore, HC-05 could also be used instead of HC-06 but not vice versa. In this project, HC05 is used on the robo car side i.e the receiver side and HC06 is used on the gesture unit i.e transmitter side.

4.5 Arduino Lilypad

Arduino Lilypad is an Arduino variant with a small size and is designed specially for wearable electronics. Due to it's compact size, assembling all the components together on a small handheld unit or on a hand glove is easier. There are multiple variants of Arduino Lilypad as well. Lilypad Atmega328P Main board is used in this project.

4.6 Programmer for Arduino Lilypad

Arduino Lilypad 328P Main board used in this project requires a separate programmer module for uploading programs to it from a computer. FT232RL USB to TTL Serial Adapter Module is used in this project.

4.7 Gyroscope and Accelerometer Module MPU6050

MPU6050 sensor for measuring the tilt angle of the hand held gesture unit is used in this project. MPU6050 module provides an I2C interface for reading gyroscope angular velocity and acceleration values. These values can be used to calculate the roll and pitch of the handheld unit. Depending on the roll and pitch values, appropriate code is sent to the robo car.

4.8 Battery Power

Power/current requirements for Arduino are low and it can easily be powered for an extended duration from a regular non-rechargable 9V battery. However, the power/current requirements for DC motors (2 motors in our case) are quite high and the standard 9V battery may get discharged within 30mins or less. A Pair of 18650 Lithium-Ion Batteries connected in series is used in this project on the robo car side.
On the hand gesture unit, a 3.7V rechargable lithium ion battery is used.

5 High Level System Interface Diagram

Figure 1 shows a high level system interface diagram.

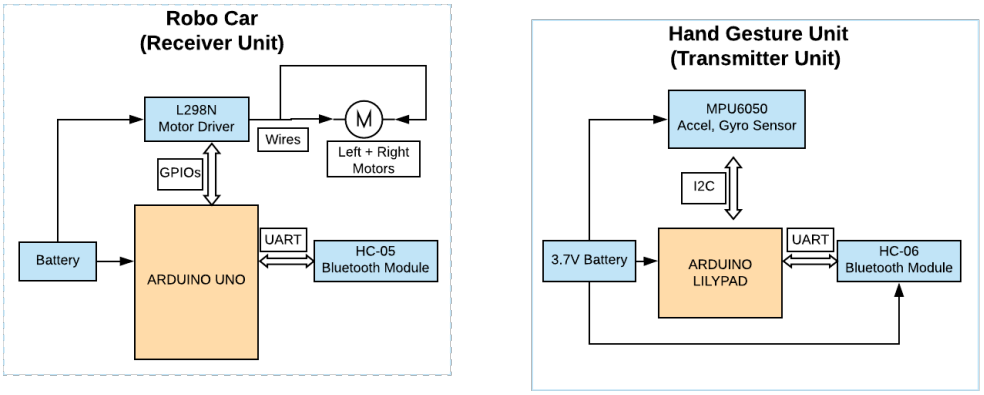

Figure 1: High Level System Interface Diagram

6 Pin Connectivity

Figure 2 shows pins names on Arduino Lilypad. Please refer to this diagram along with lilypad pin connection tables below for understanding connections with Lilypad.

Table 2 documents connections between Lilypad and FT232RL USB to Serial Adapter Module used for programming Lilypad.

S.No	Lilypad	FT232RL
1	GND	GND
2	GND	No Connection
3	VCC	VCC
4	RXD	TX
5	TXD	RX
6	DTR	DTR

Table 2: Pin connections between Lilypad and FT232RL Serial Adapter Module

Table 3 documents connections between Lilypad and the different modules on the handheld gesture control unit. HC-06 refers to the bluetooth module and MPU6050 refers to the accelerometer and gyroscope module.

S.No	Lilypad	HC-06	MPU6050
1	10	TX	
2	11	RX	
3	A4		SDA
4	A5		SCL

Table 3: Pin connections between Lilypad and different modules on handheld gesture control unit

Table 4 documents the pin connectivity of Arduino Uno with the different modules on the RoboCar unit.

S.No	Arduino	L298N	Hc-05
1	9	IN1	
2	10	IN2	
3	11	IN3	
4	12	IN4	
5	A1		RX
6	A0		TX

Table 4: Pin Connectivity of Arduino Uno with different modules on the RoboCar unit

7 Project Demo and Snapshots

Figures 3, 4, 5, 6, 7, 8 show snapshots of the working prototype. Link to Demo Video - (LINK) .

8 Important Notes and Learnings

Important notes and learnings from the project are captured in this section.

8.1 Configuration for HC-05 Bluetooth Module

Some of the important notes regarding HC-05 module and it's configuration are captured below.

1. HC-05 module is used on the RoboCar side i.e receiver side in this project. It is configured to act as bluetooth Master and initiate bluetooth connection with bluetooth Slave on the transmitter side.

2. For configuring HC-05 settings like name, password, baud rate, etc it needs to be in command mode. By default HC-05 is in communication mode. Command mode can be entered by firstly connecting EN pin on HC-05 to 5V or 3.3V and secondly keeping the switch on HC-05 pressed while it is being powered ON. Slow LED blinking indicates HC-05 has entered command mode.

3. When communicating with HC-05 in command mode through Serial Monitor, ensure the following settings are used .

   - Select Both NL and CR in Serial Monitor
   - A simple code using SoftwareSerial library for communicating with HC-05 is uploaded.
   - Correct Baud Rate is used. Default command mode baud rate for HC-05 is either 9600 or 38400.

4. Perform the following configurations in command mode.

   - Configure baud rate to be 38400
   - Configure name to be something indicative like GestureControlledRoboCarBT
   - Configure password. Default password may be 0000 or 1234.
   - Configure HC-05 to act as bluetooth Master
   - Configure HC-05 to pair with specific device only using slave device address

5. Some useful command mode or alternatively called AT mode commands are mentioned below.

   - AT
   - AT+VERSION
   - AT+NAME =
   - AT+ORGL to restore factory defaults
   - AT+UART = 38400,0,0 for baud rate, stop bit, parity setting
   - AT+RESET to reset
   - AT+ADDR for address
   - AT+RNAME?Address of BT device to get name of the BT device with specified address
   - AT+PSWD = "0000" to set password
   - AT+ROLE = 1 to set as Master (0 is for Slave)
   - AT+CMODE=0 to connect to fixed address (1 is for any)
   - AT+BIND = 98D3,34,90BFD7 (for example) to set fixed address to connect to . Use commas instead of colons

8.2 Configuration for HC-06 Bluetooth Module

Some of the important notes regarding HC-06 module and it's configuration are captured below.

1. HC-05 module is used on the Hand Gesture Unit i.e transmitter side in this project. It acts as bluetooth slave.

2. For configuring HC-06 settings like name, password, baud rate, etc it needs to be in command mode.

3. Unlike HC-05, there is no switch on HC06 which needs to be pressed for entering into command mode. By default HC-06 is in command mode until it pairs up with a bluetooth device for communicating.

4. After HC-06 pairs up with another device, LED onboard HC06 remains always ON.

5. When communicating with HC-06 in command mode through Serial Monitor, ensure the following settings are used .
    - Select No Line Ending in Serial Monitor
    - A simple code using SoftwareSerial library for communicating with HC-06 is uploaded.
    - Correct Baud Rate is used. Default command mode baud rate for HC-06 is either 9600 or 38400.
6. Perform the following configurations in command mode.
    - Configure baud rate to be 38400
    - Configure name to be something indicative like lilypadBT
    - Configure password. Default password may be 0000 or 1234.
    - Note the bluetooth address of HC06. This address is used by the bluetooth receiver on the RoboCar side to identify the correct bluetooth device to connect to.
7. Some useful command mode or alternatively called AT mode commands are mentioned below.
    - AT
    - AT+VERSION
    - AT+NAMElilypadBT to set Name to lilypadBT
    - AT+PIN0000 to set PIN to 0000 (for example)
    - AT+BAUD4 for setting 9600 baud rate, AT+BAUD6 for setting 38400 baud rate
    - AT+ADDR? to know bluetooth address of the HC06 module

8.3 Important Notes on MPU6050

Some of the important notes regarding the Accelerometer and Gyroscope module MPU6050 are captured below.

1. MPU6050 module needs callibration for accelerometer and gyroscope values. Factory programmed default values may not give the correct

values. This can be checked by uploading a simple sketch to Arduino which displays raw values for angular velocity and acceleration and keeping the module still.

2. When MPU6050 is kept still, angular velocities in x,y,z directions should be zero. Acceleration in x,y directions should be zero and in z direction should be equal to =/-1g.

3. If raw values are incorrect, there are offset registers present in MPU6050 which can be written through the I2C bus for correcting the raw values.

4. Callibration is the most important step and must always be performed. However, note that once MPU6050 is powered off, offset registers are again set to default factory values. So, callibration needs to be performed every time MPU6050 module is powered ON.

5. Other than writing to registers, dynamic offset calculation can also be performed by adding or subtracting a dynamically calculated offset value from each of the raw readings.

6. The offset values entered into the registers, assume specific sensitivities for angular velocity and acceleration. For our module, offsets entered were found to correspond to gyroscope sensitivity of +/-1000degrees per second and accelerometer sensitivity of +/-16g although register specification document for MPU6050 mentions +/-8g

7. There are some libraries available online in public domain which provide a very good API for communicating with MPU6050.

8. Roll and Pitch values for determing the tilt of the hand held gesture unit are calculated using the raw gyroscope and accelerometer values. MPU6050 libraries provide an easy way for getting these values.

9 Contact Information

If you have any questions or need help in building this project, please feel free to reach out to us by messaging us on Facebook (LINK), or through the message option on our website (LINK) or sending me an email at vasu.gupta9@gmail.com

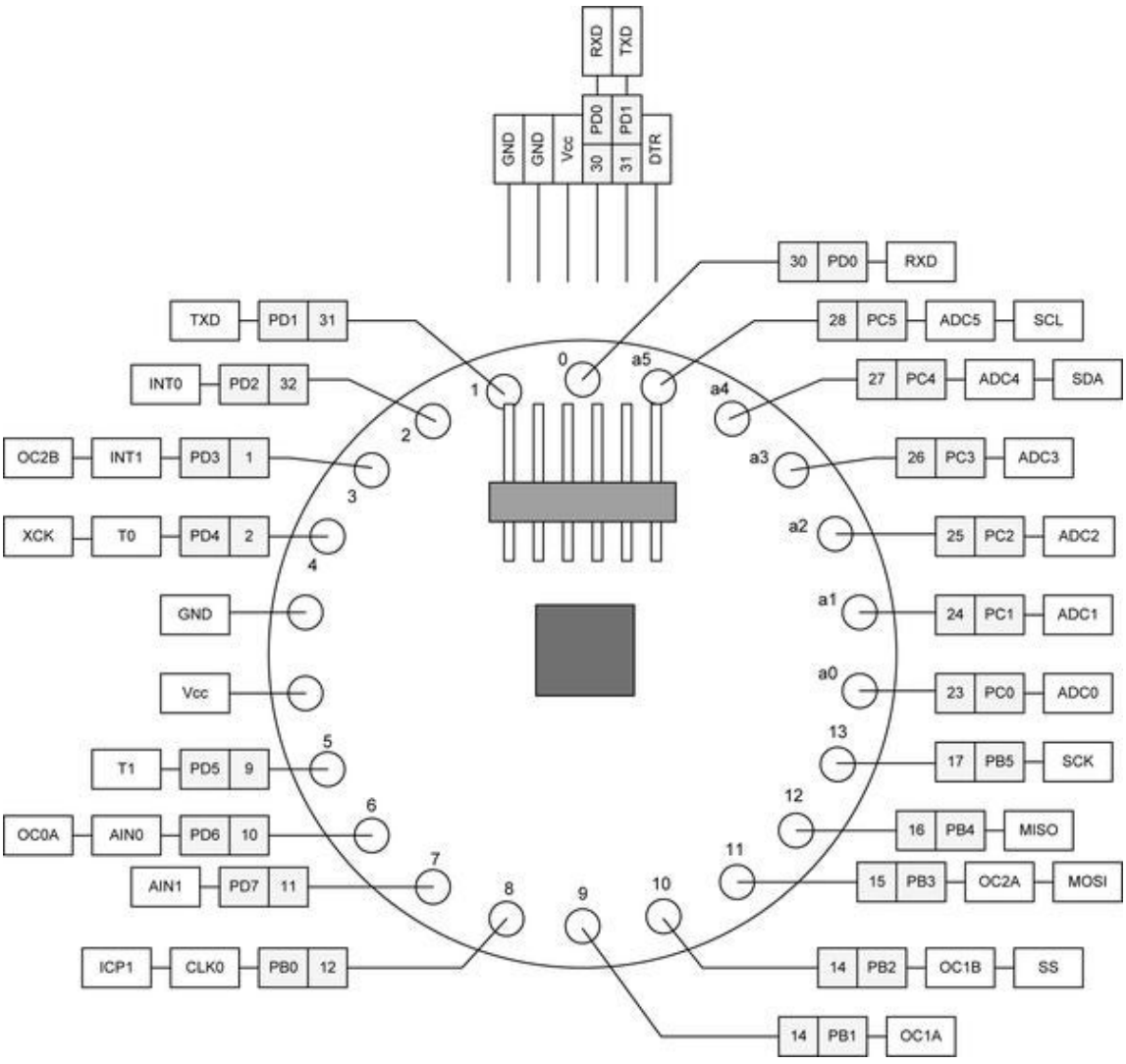

Figure 2: Arduino Lilypad Pins

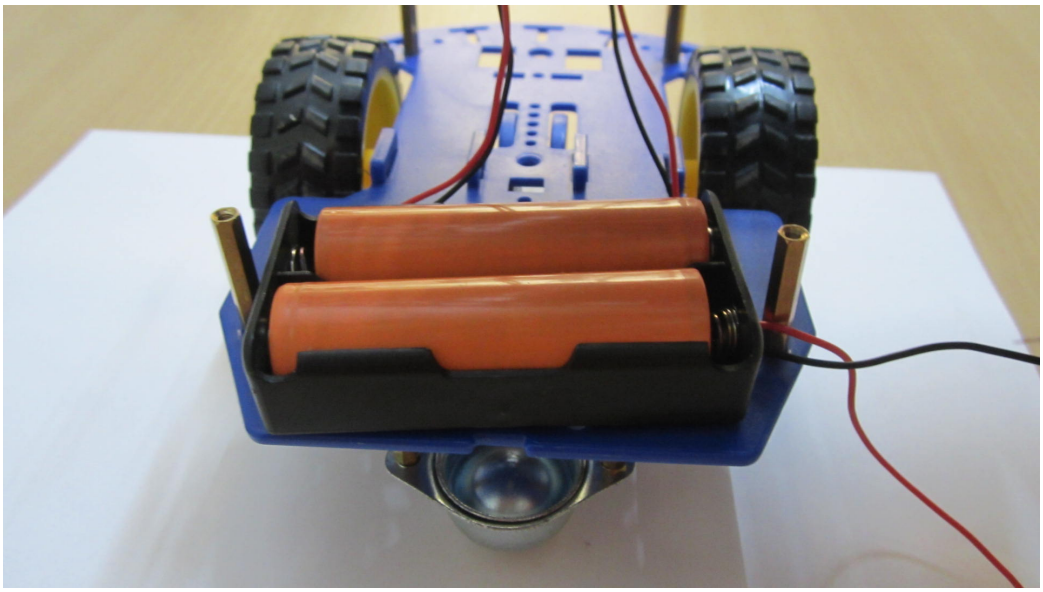

Figure 3: RoboCar Snapshot 1

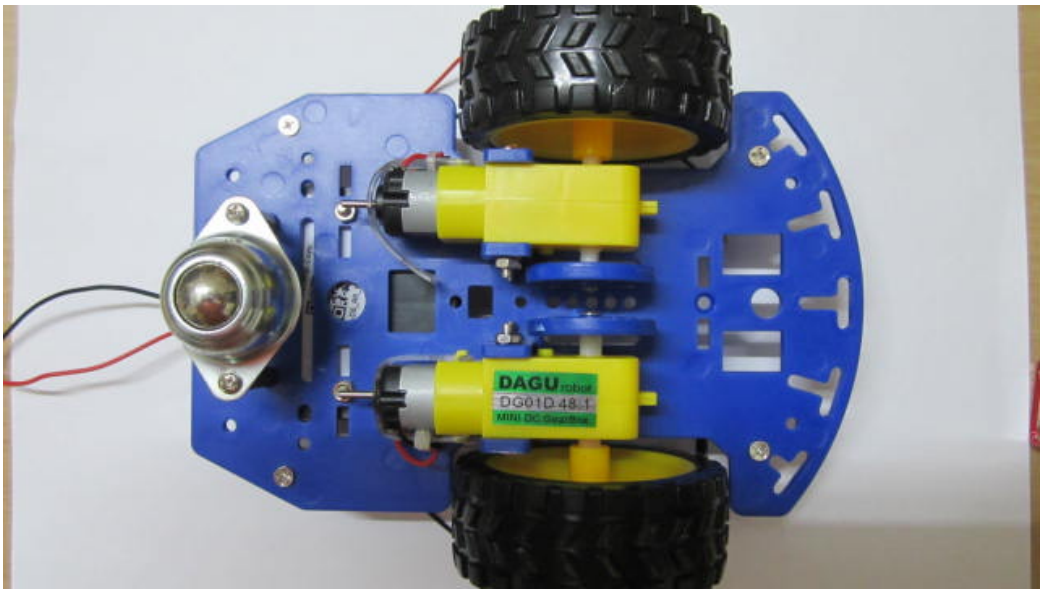

Figure 4: RoboCar Snapshot 2

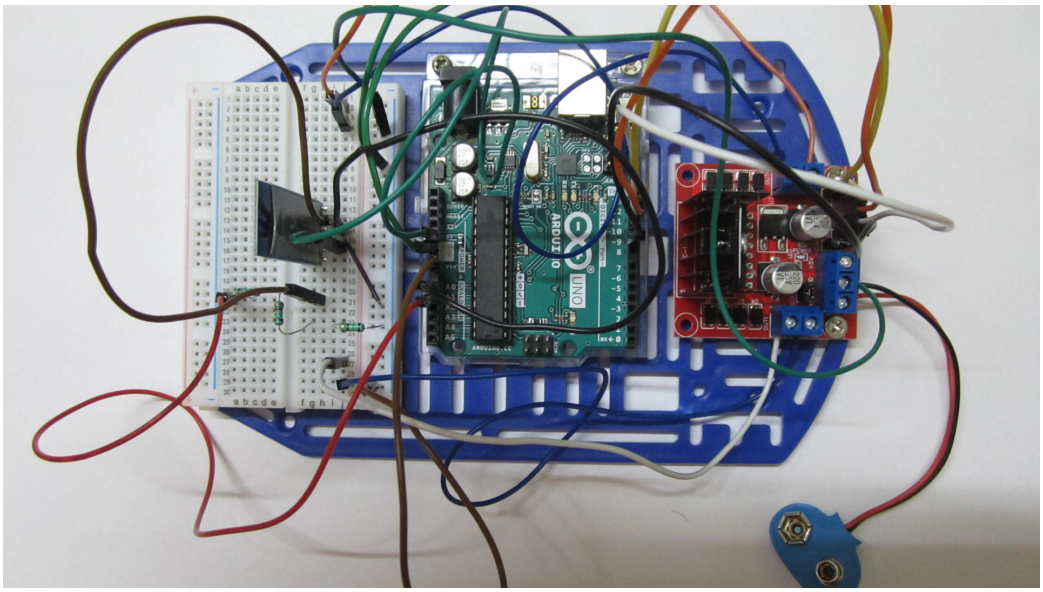

Figure 5: RoboCar Snapshot 3

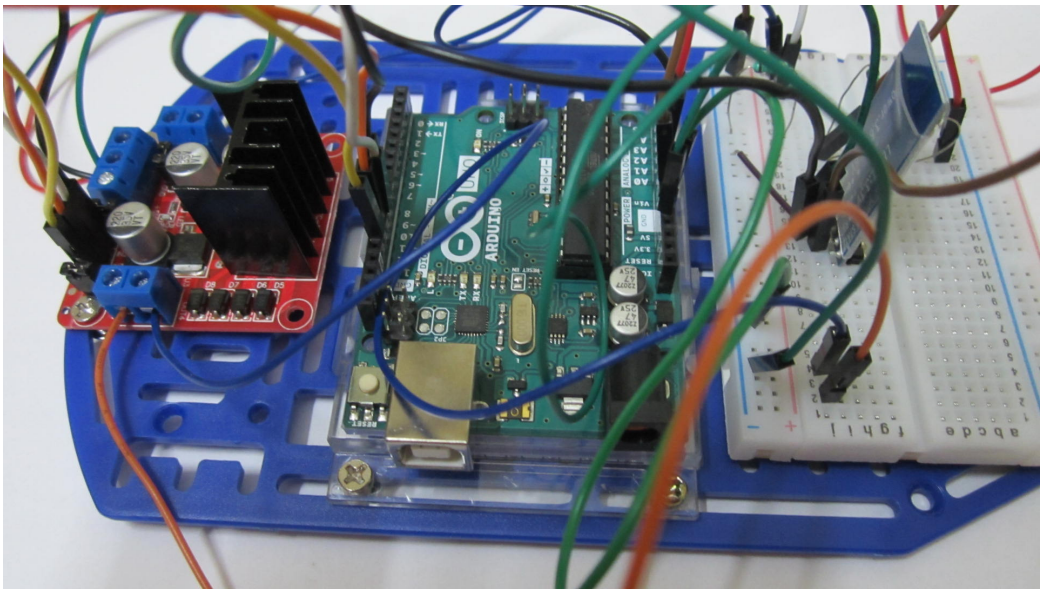

Figure 6: RoboCar Snapshot 4

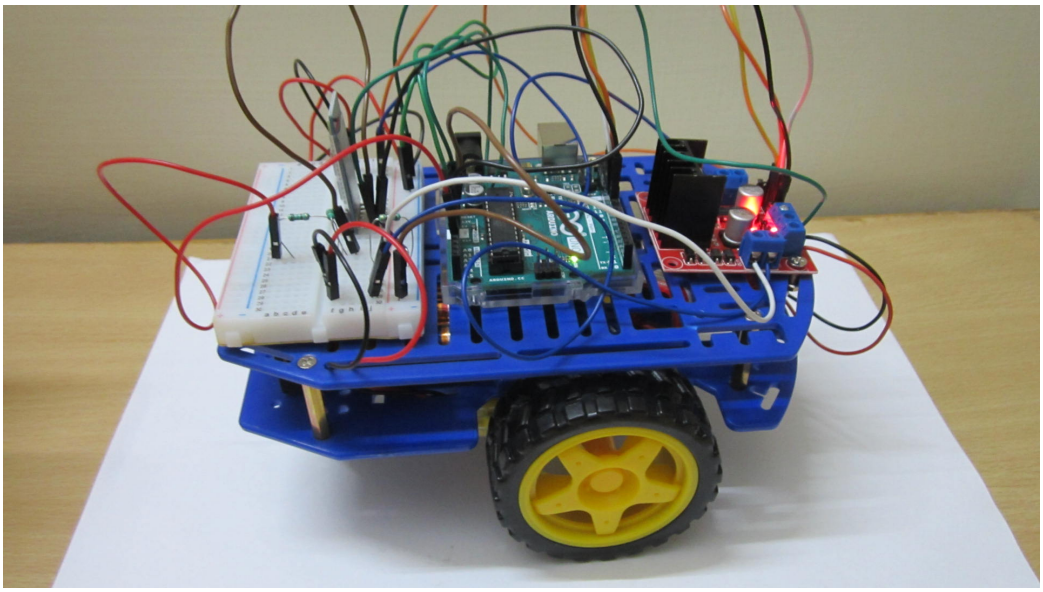

Figure 7: RoboCar Snapshot 5

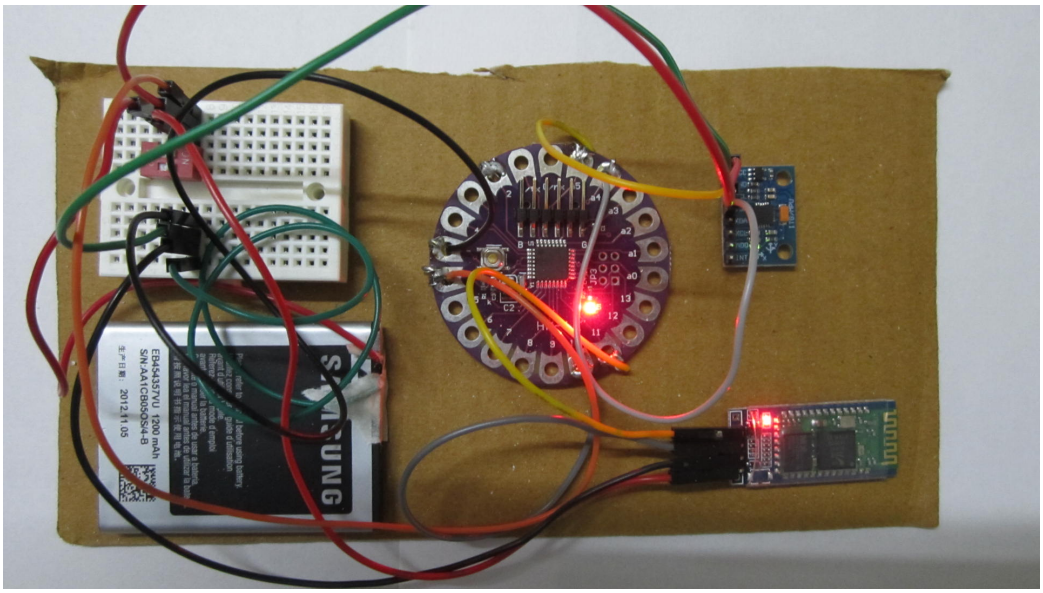

Figure 8: Handheld Gesture Control Unit