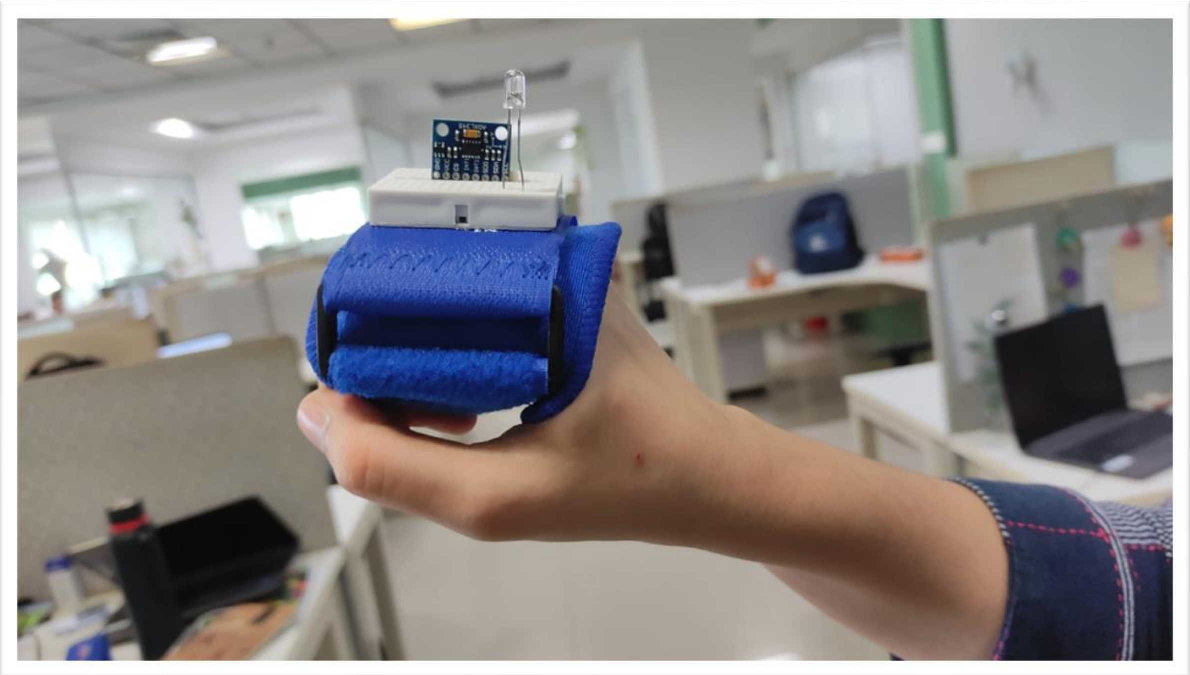
Build Gesture controlled Devices

Introduction to the course:	Build hand gesture-based devices that can be used as alternate control devices for PC mouse or to navigate a mobility device.
What does this course aim to achieve?	• Learn about alternate PC control • Learn about inertia measurement sensors , HID devices and how to convert a normal microcontroller to a PC HID device. • Learn more about the people with limited hand movements and their requirements. • Learn about Bluetooth wireless communication • Learn about IMU sensors, Motor controller boards, Bluetooth modules. • Learn about AT commands and configuration • Learn how gestures can be converted to mobility
What is being built in this course:	1) A wearable gesture controlled mouse that helps to access a windows PC without mouse/keyboard. The microcontroller used mimics an HID device and does the mouse movements. 2) A wearable hand gesture device that includes a gyro sensor & microcontroller, which converts gyro values to corresponding motor movements and sends the data to an RC car via Bluetooth
How is it being tested:	1) Different people will have different comfort hand positions, the device can adjust to that position with the click of calibration switch. Activities are given to adjust sensitivity values and also to look for better positions for sensor and right click/left click buttons 2) The Controllability of the RC car can be corrected by changing the sensitivity value as per individual preference. If the movements are beyond the possible correction values, the sensors/Car setup/wiring needs to be reconfigured/tested

Course Prerequisites	• Basic C/C++ programming skill • Basic understanding on sensors and other electronics • Build club project on maze solving robot

Build a Gesture controlled Mouse for PC



Contents

A. Prerequisite

Topic	Resources
Understanding Gyro Sensor & HID	Project videos HID
AT Command	Link

B. Aim

Build a hand gesture-based device for people with limited hand motor movements that can control mouse pointer of a PC.

C. Concept

In this project, we will create a gesture based device that mimics mouse actions and controls a windows PC. This device can be used by persons with hand impairments who cannot use a conventional mouse to control the PC. The wearable gesture device could communicate with the PC which can in turn communicate with apps and avail many services such as a communication software or accessing the internet for healthcare, food delivery etc.

Here we will be using dexter board as an HID device.

HID (Human Interface Device)

A human interface device, often known as a HID, is a kind of computer hardware that is typically used by people and that receives input from people and provides outputs to people in many forms.

The USB-HID specification is most frequently used when the term "HID" is used. Mouse, Keyboards & Joysticks are all part of HID devices.

D. Components

Sl.no	Component	Cost (Rs)
1	Dexter board	
2	MPU6050 sensor	150
3	Small breadboard (170points)	25
4	Button module	40
5	Long wires	50
6	Jumper wires	60
7	USB cables	100

Note: All the components above are reusable for other projects

Dexter board

MPU 6050 Sensor

Jumper wires

Breadboard

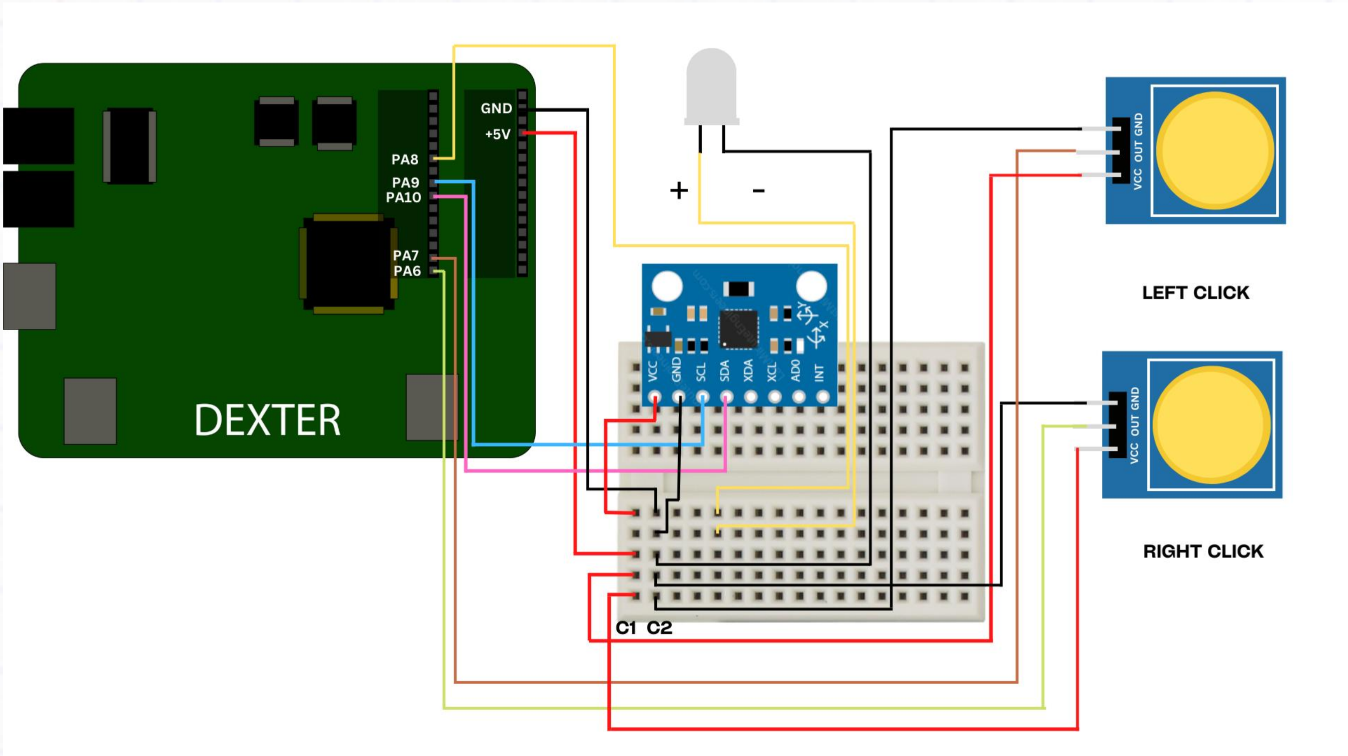
USB Cable

Button module

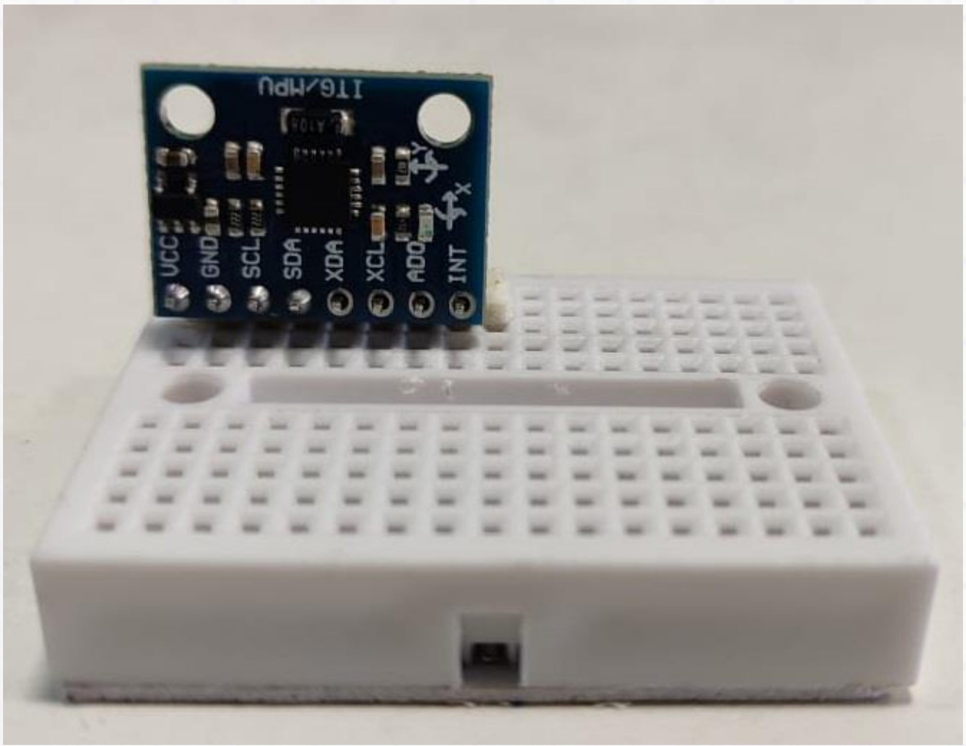
E. Connections

Circuit Diagram

- NOTE: Before starting the connections, verify using a multimeter that all the wires are working. Also ensure that the connections are strong, else the setup may not work.



Orientation of MPU6050 sensor

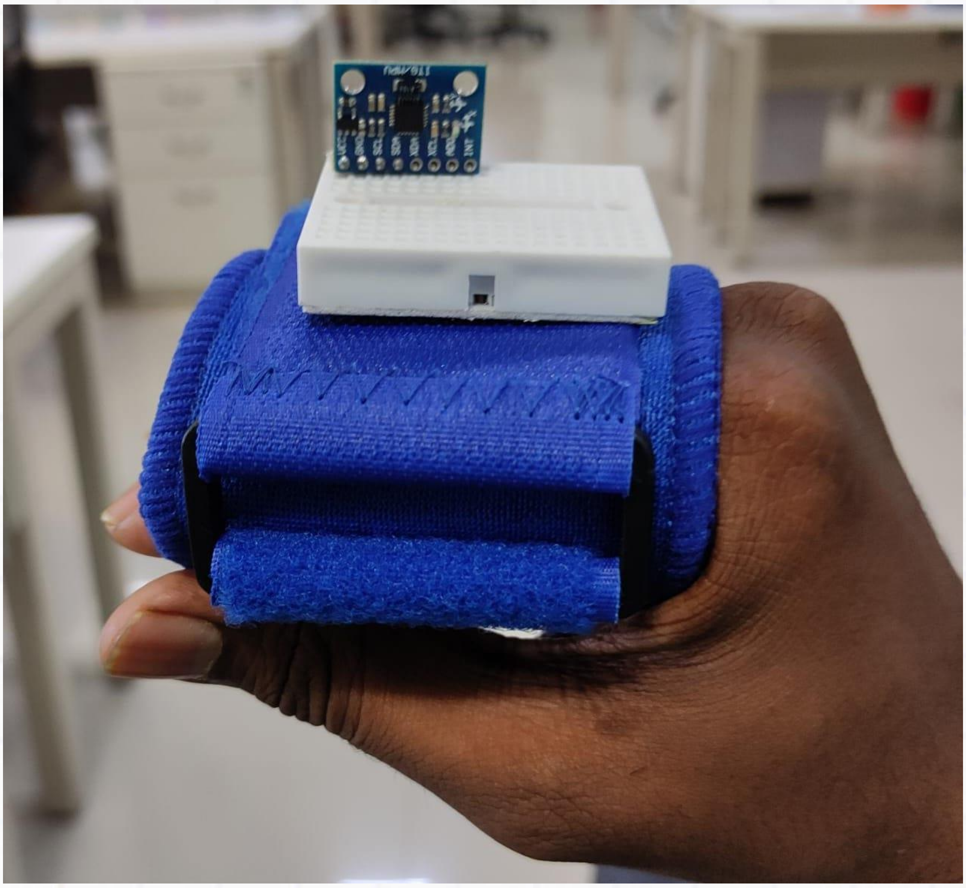


Since we are using only y and z axis values for this project. Orient the sensor in such a way that it is perpendicular to the breadboard as shown in the above picture.

Detailed Connection Steps

Step 1

Stick the small breadboard to the wrist band you have in this orientation



Important

Take 2 male-to-male **long** wires Connect -

- **GND** of the dexter to **C2 column** of the breadboard
- **+5V** of the dexter to **C1 column** of the breadboard

Step 2

Take 2 male-to-male **long** wires and connect the pins of the MPU6050 sensor:

- **VCC** to **C1 Column** of breadboard
- **GND** to **C2 Column** of breadboard

Step 3

Take 4 male-to-female **long** wires and connect the pins **VCC** and **GND** pins of 2 button modules as below :

- **VCC of button** to **C1 Column** of breadboard
- **GND of button** to **C2 Column** of breadboard

Step 4

Take 2 male to male **long** wires and connect MPU6050 sensor to dexter board as below:

- **SCL** of sensor to **PA9** of dexter board
- **SDA** of sensor to **PA10** of dexter board
-

Take 2 more male to male **long** wires and connect the two button modules to dexter board as below:

- **OUT** pin of RIGHT CLICK button to **PA6** of dexter board
- **OUT** pin of LEFT CLICK button to **PA7** of dexter board

Step 5

Take 1 male to male **long** wire, take a LED. and connect as below:

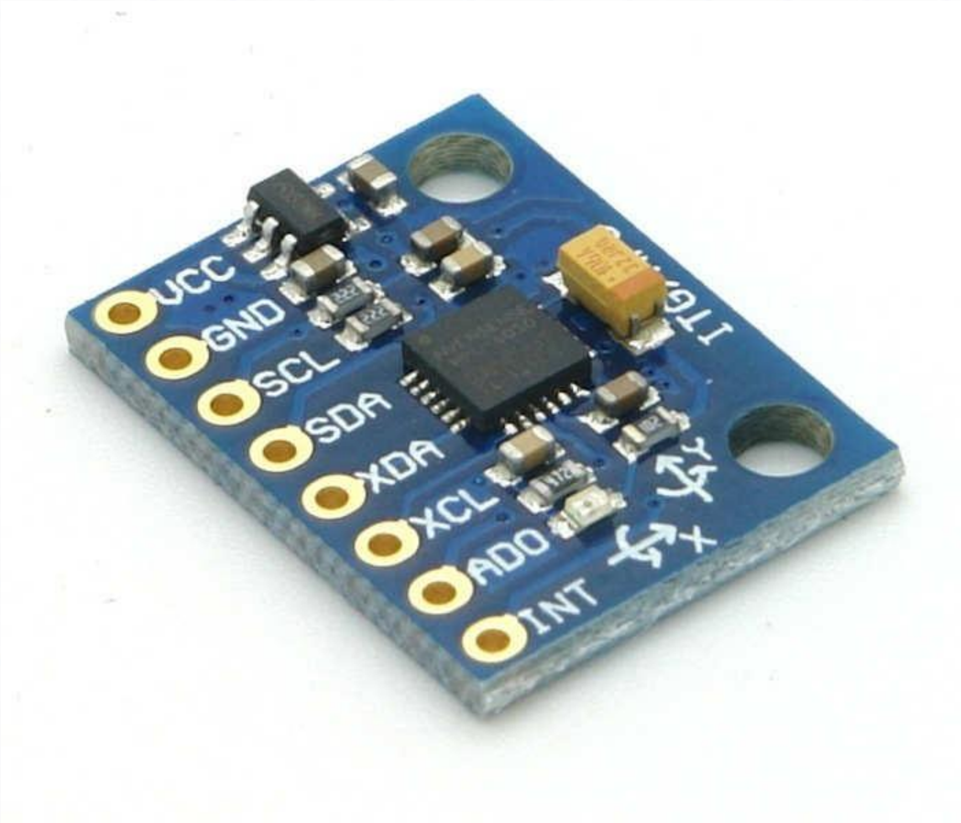
- Negative side of LED to **C2** column of the breadboard
- Positive side of LED to **breadboard.** From that point on breadboard connect the long wire to **PA8** of dexterboard

Step 6

Place the whole setup in such a way that breadboard is held on hand, the buttons and dexter board are fixed to the table.

F. Components in detail

1)MPU6050 Sensor Module



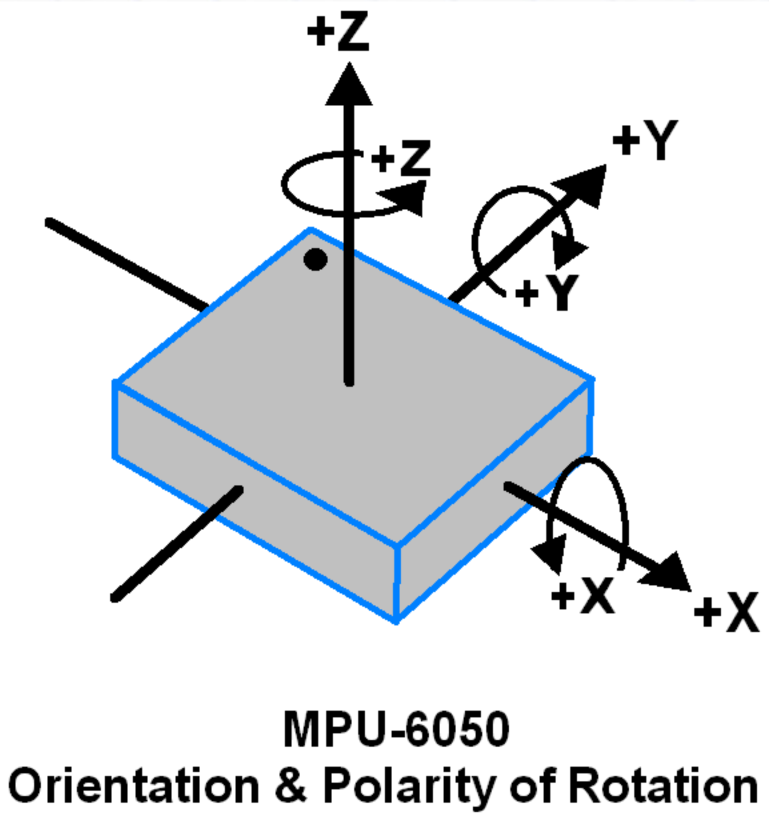
MPU6050 sensor module is complete 6-axis Motion Tracking Device. It combines 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in small package.

NOTE: Also the sensor has an additional feature of on-chip Temperature sensor. It has I2C bus interface to communicate with the microcontrollers.

It has Auxiliary I2C bus to communicate with other sensor devices like 3-axis Magnetometer, Pressure sensor etc.
If 3-axis Magnetometer is connected to auxiliary I2C bus, then MPU6050 can provide complete 9-axis Motion Fusion output.

3-Axis Gyroscope

The MPU6050 consist of 3-axis Gyroscope with Micro Electro Mechanical System(MEMS) technology. It is used to detect rotational velocity along the X, Y, Z axes as shown in below figure.

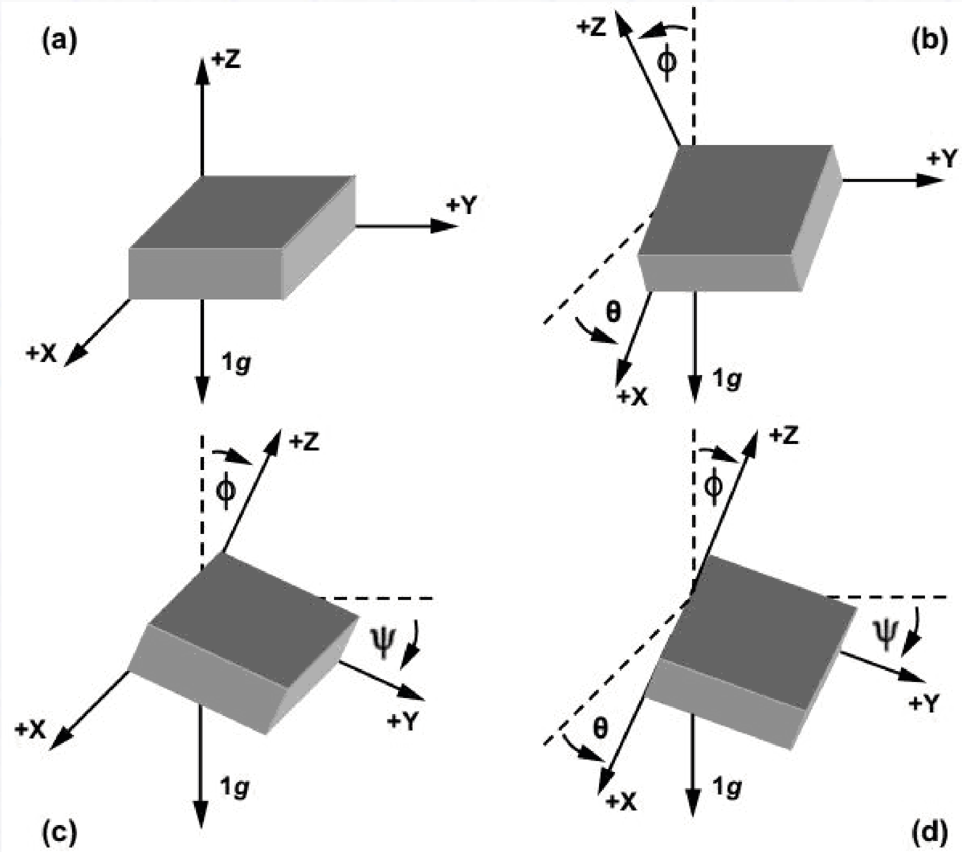


- When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a MEM inside MPU6050.

- The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate.
- This voltage is digitized using 16-bit ADC to sample each axis.
- The full-scale range of output are +/- 250, +/- 500, +/- 1000, +/- 2000.
- It measures the angular velocity along each axis in degree per second unit.

3-Axis Accelerometer

The MPU6050 consist 3-axis Accelerometer with Micro Electro Mechanical (MEMs) technology. It used to detect angle of tilt or inclination along the X, Y and Z axes as shown in below figure.



- Acceleration along the axes deflects the movable mass.

- This displacement of moving plate (mass) unbalances the differential capacitor which results in sensor output. Output amplitude is proportional to acceleration.
- 16-bit ADC is used to get digitized output.
- The full-scale range of acceleration are +/- 2g, +/- 4g, +/- 8g, +/- 16g.
- It measured in g (gravity force) unit.
- When device is placed on flat surface it will measure 0g on X and Y axis and +1g on Z axis.

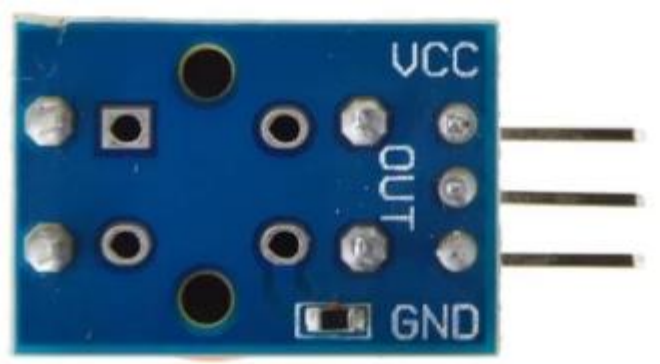
The MPU-6050 module has 8 pins as listed below:

1. **INT:** Interrupt digital output pin.
2. **AD0:** I2C Slave Address LSB pin. This is 0th bit in 7-bit slave address of device. If connected to VCC then it is read as logic one and slave address changes.
3. **XCL:** Auxiliary Serial Clock pin. This pin is used to connect other I2C interface enabled sensors SCL pin to MPU-6050.
4. **XDA:** Auxiliary Serial Data pin. This pin is used to connect other I2C interface enabled sensors SDA pin to MPU-6050.
5. **SCL:** Serial Clock pin. This pin to connected to microcontrollers SCL pin.
6. **SDA:** Serial Data pin. This pin is connected to microcontrollers SDA pin.
7. **GND:** Ground pin. This requires a ground connection.
8. **VCC:** Power supply pin. This requires a +5V DC supply.

For detailed theory refer this link

> **https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/**

2) Button module



Features

Interface: Standard electronic block (GND, VCC, SIG)
Working voltage: 2 - 5.5 Volt DC
Working Current: 0.55mA(MAX)
Output: Digital level (press high, release low)

G. Implementing the gesture project

<u>Variables & Functions</u>

1) **Sensitivity**

Inside main.c file, this variable can be found. Increasing/Decreasing the existing value will change the sensitivity of the mouse movements.

2) **MPU6050_Init()**

This function initializes the IMU sensor by checking for power issues, setting the data rate, and also setting up the gyro and accelerometer configurations.

3) **MPU6050_CALIB()**

This function calibrates the minimum and maximum values of x, y & z to the sensor's current position. It reads the current x, y & z values and helps in adapting to the current sensor position.

4) **MPU6050_Read_Gyro()**

Reads the gyro values ie, the angles of x ,y, z with respect to the calibrated position

5) MPU6050_SenttoMouse()

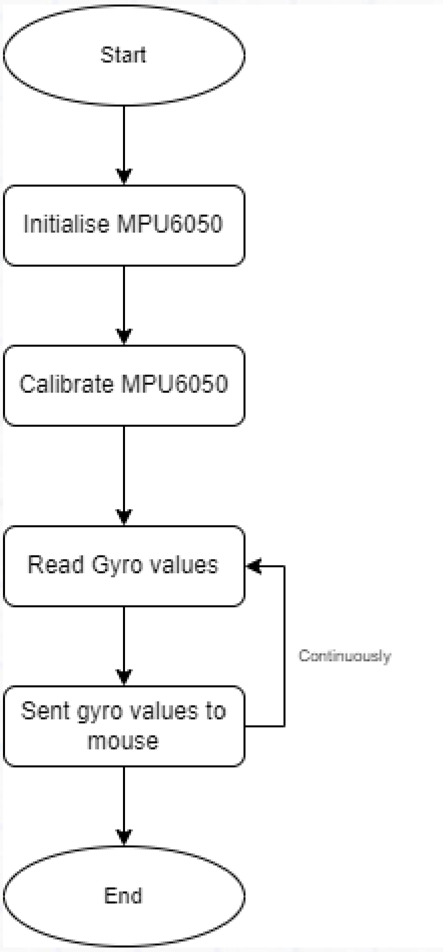
Sends the x,y,z position values of sensor to PC to move the mouse pointer between different positions.

Downloads & Installation

1) Download the Project Workspace file '**Gesture_mouse_windows.zip**' given in the project page on the Build Club website.
2) In the **Workspace** folder in your C: drive, create a new folder named **'Gesture_mouse_windows**'.
3) Then i) Launch the STM IDE, ii) Select the **Gesture_mouse_windows**' folder as workspace, iii) Import the ZIP file **Gesture_mouse_windows**', iv) Navigate to **main.c**

Flowchart of the Code

The flowchart diagram represents the flow of the code for the gesture device. Along with the flowchart the program part can be easily implemented.



Implementing the Code

Refer the flowchart and functions above to implement the code in **App** function in main.c file. Follow the steps below:

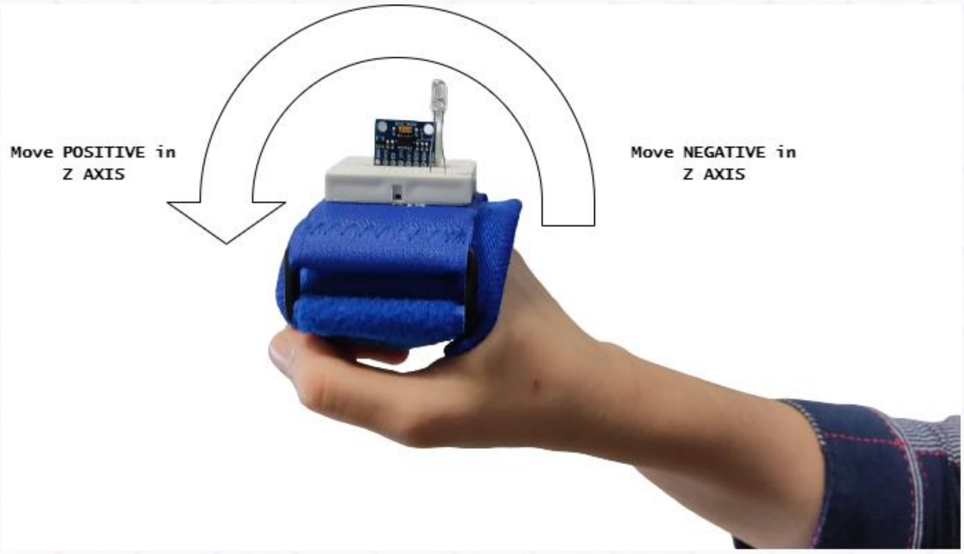
1) Initialize the MPU6050 sensor using the **MPU6050_Init()** function and calibrate the sensor with **MPU6050_CALIB()** function
2) After calibration function, the values of the Gyro should be read continuously and sent to mouse pointer. You can use **MPU6050_Read_Gyro()** and **MPU6050_SenttoMouse()**

   inside a while loop

After writing these codes in STM32 IDE. The code can now be uploaded to the Dexter board by hitting 'Run'.

3) Place hand in a steady position so that sensor initialization and calibration is carried out. The calibration LED will turn **off** indicating the completion of calibration
4) Now you can move the device. Check whether the mouse pointer is moving along with your hand movement
5) Using left button and right button connected to the device, you can perform the actions of a regular mouse

Wrist movement


Move POSITIVE in
Z AXIS
Move NEGATIVE in
Z AXIS


Move POSITIVE in
Y AXIS
Move NEGATIVE in
Y AXIS

H. Activities

1.Try changing the sensitivity value and find out a value suitable for each of your team mates. Try opening multiple apps in PC

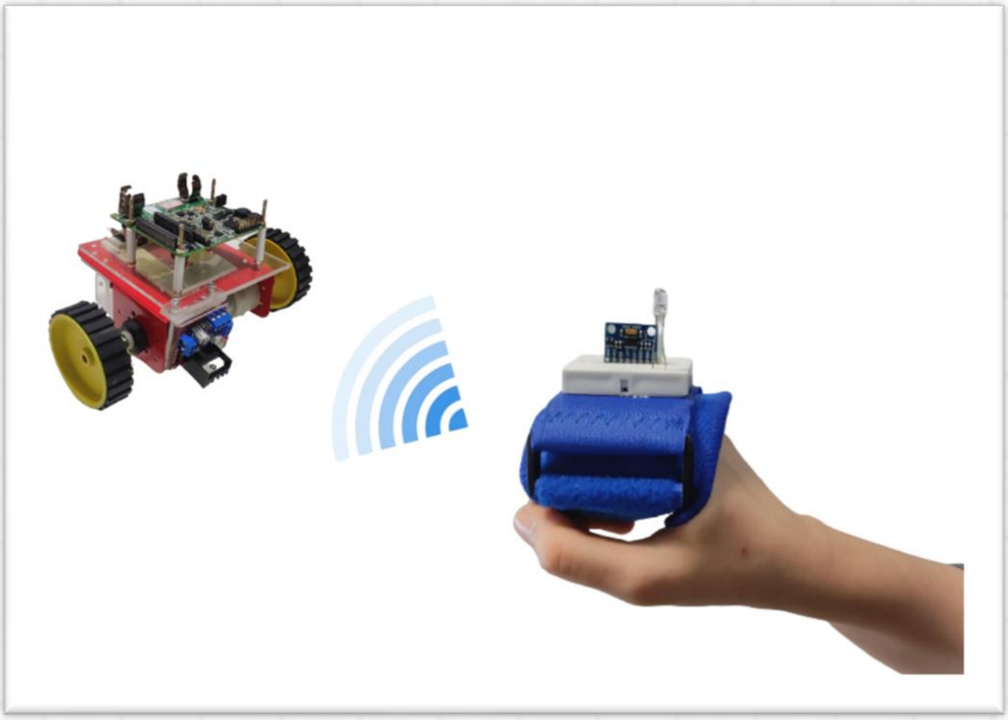
2. Interchange the sensor and buttons positions and check whether the sensitivity is suitable for people with limited hand movements.

3. Try to position the sensors and buttons, so that it can be operated by single hand.

4. Make a video presenting, the build of gesture device, how you completed the activities and challenges. Share it with the Build Club Community on the Discord Server!

This completes part A of the project

Build a Gesture controlled MicroMouse



-

Contents

A. Prerequisite

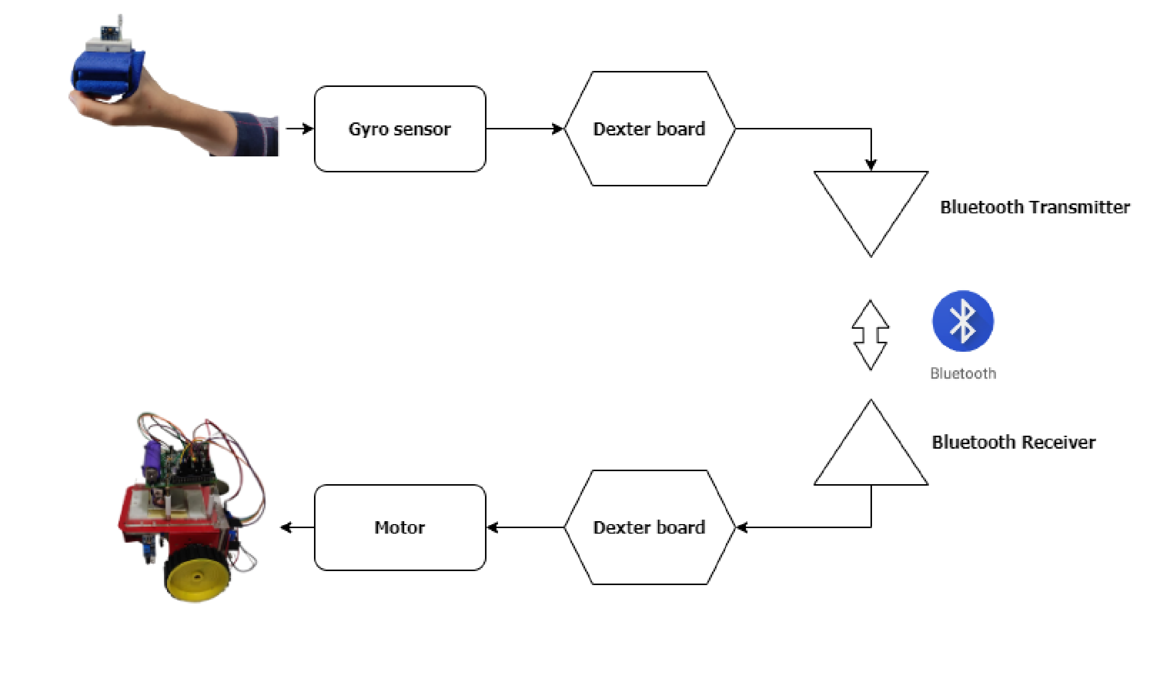
Topic	Resources
Understanding Bluetooth Modules	Bluetooth module
Maze robo Build club project	Link

B. Aim

Build a hand gesture-based device to wirelessly control a robot car (MicroMouse).

C. Concept

This project will enable you to control a robot car (MicroMouse) wirelessly with dexter boards. The hand movements will be captured by a wearable hand gesture device and the data will be sent to the MicroMouse wirelessly. The Micro Mouse will move according to the hand movements.



The project can be later converted to control real vehicles or to assist people in wheelchairs. The conventional joystick controlled powered wheelchairs can be replaced and the hand gesture device can be used as an alternative drive control for powered wheelchairs

Here we are using a gyro sensor to detect the orientation of hand and the data is transmitted to a bluetooth transmitter module. The Bluetooth receiver then receives the hand data which is converted to corresponding motor movements on the MicroMouse side.

D. Components

Sl.no	Component	Cost (Rs)
1	Dexter board	
2	MPU6050 sensor	150
3	Small breadboard (170points)	25
4	Bluetooth module HC-05	500
5	2 Wheel Smart Car Robot Chassis Kit	400
6	Jumper wires	60
7	USB cables	100
8	L298 Dual Motor driver board	110

Note: All the components above are reusable for other projects

Dexter board

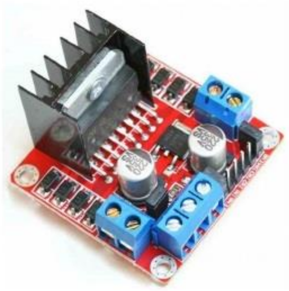
MPU 6050 Sensor

Jumper wires

Breadboard

USB Cable

HC-05 Bluetooth



L298_Motor driver

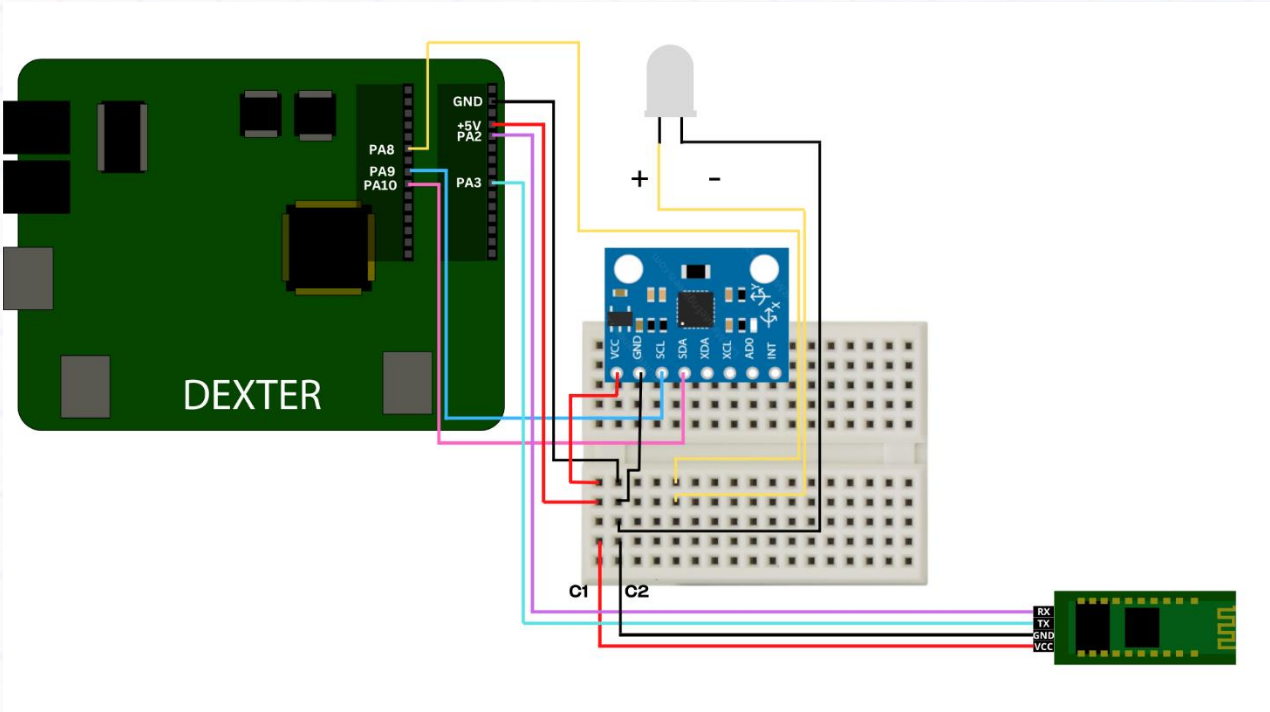
MicroMouse kit

E. Connections

Circuit Diagram

- NOTE: Before starting the connections, verify using a multimeter that all the wires are working. Also ensure that the connections are strong, else the setup may not work.

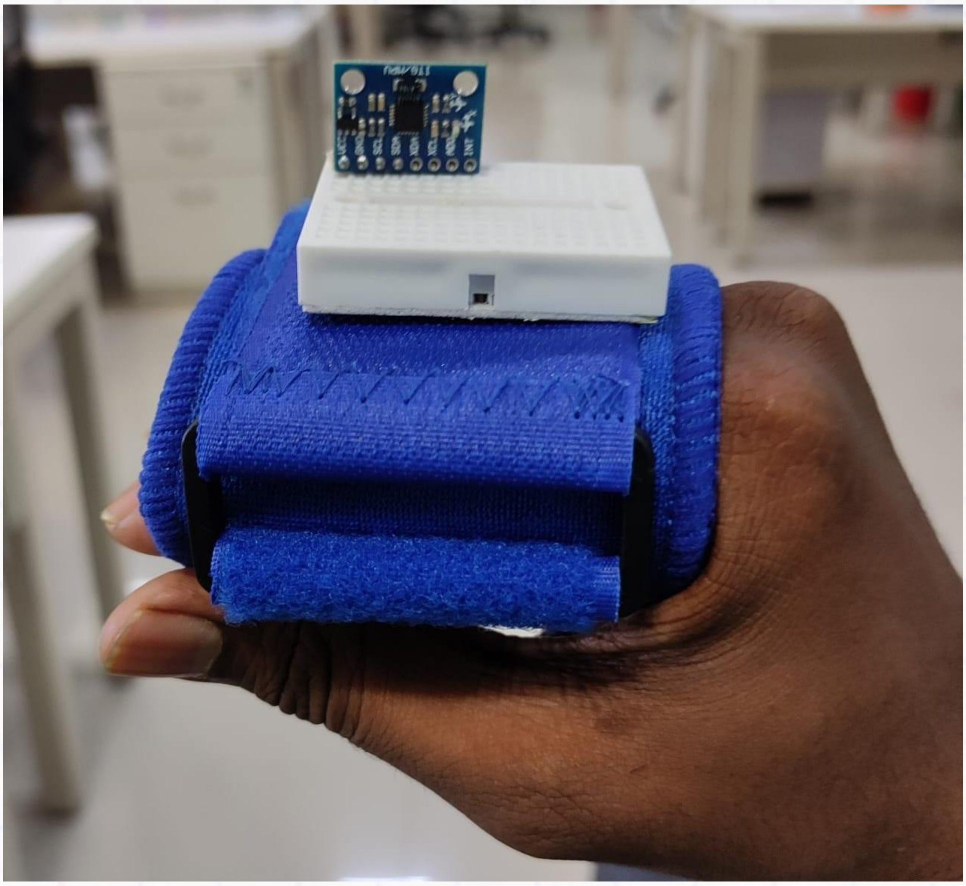
Connection diagram for the Wrist side



Detailed Connection Steps

Step 1

Stick the small breadboard to the wrist band you have, in this orientation



Important

Take 2 male-to-male jumper wires Connect -

- **GND** of the dexter to **C2 column** of the breadboard
- **+5V** of the dexter to **C1 column** of the breadboard

Step 2

Take 2 male-to-male jumper wires and connect the pins of the MPU6050 sensor:

- **VCC** to **C1 Column** of breadboard
- **GND** to **C2 Column** of breadboard

Step 3

Take 2 male-to-female jumper wires and connect the pins **VCC** and **GND** pins of Bluetooth module(master) as below:

- **VCC** of Bluetooth module to **C1 Column** of breadboard
- **GND** of Bluetooth module to **C2 Column** of breadboard

Note: The master/slave configuration of bluetooth module is done in theory session of this manual. You can follow the above steps for now. Later disconnect, configure it as master and reconnect the module

Step 4

Take 2 male to male jumper wires and connect MPU6050 sensor to dexter board as below:

- **SCL** of sensor to **PA9** of dexter board
- **SDA** of sensor to **PA10** of dexter board
-

Take 2 more female to female jumper wires and connect RX and TX of bluetooth modules to dexter board as below :

- **RX** pin of Bluetooth module to **PA2** of dexter board
- **TX** pin of Bluetooth module to **PA3** of dexter board

Step 5

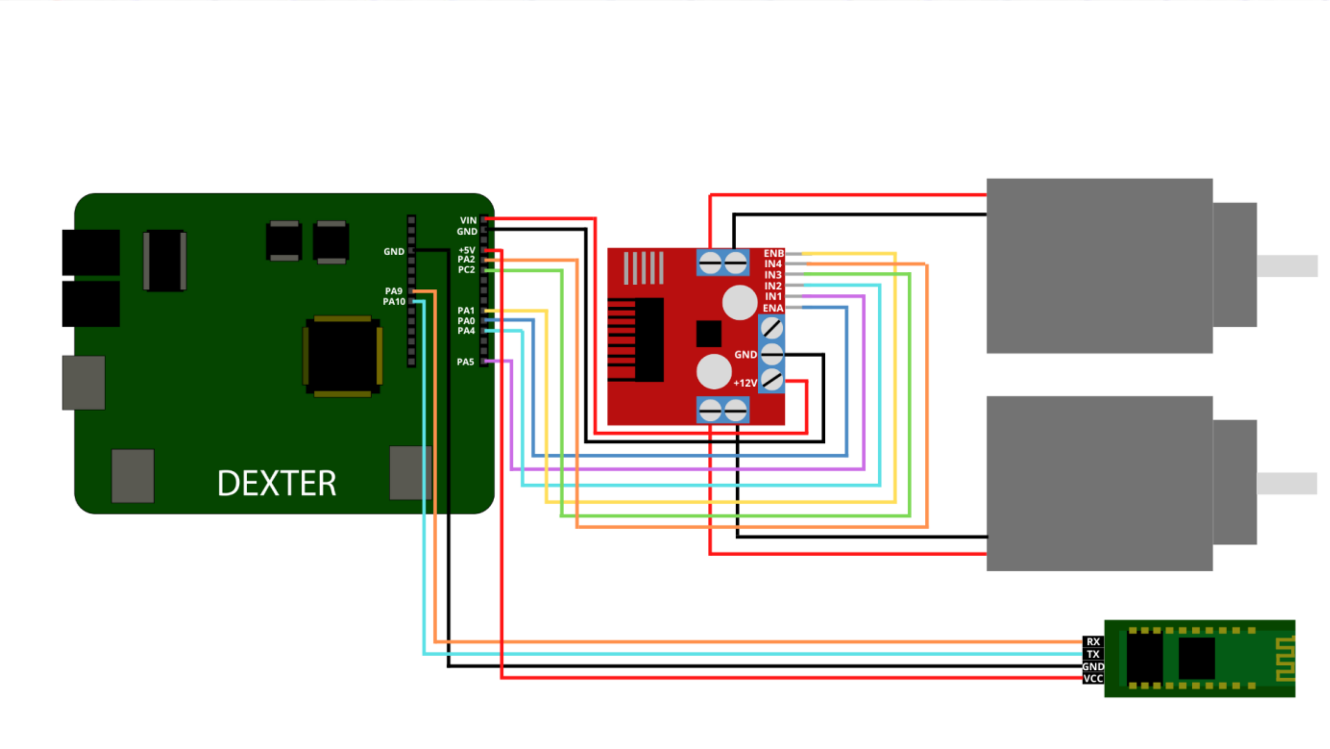
Take 1 male to male jumper wire, take a LED. and connect as below :

- Negative side of LED to **C2** column of the breadboard
- Positive side of LED to **breadboard.** From that point on breadboard connect the long wire to **PA8** of dexter board

Step 6

While operating, take dexter board on one hand with gesture device on other hand

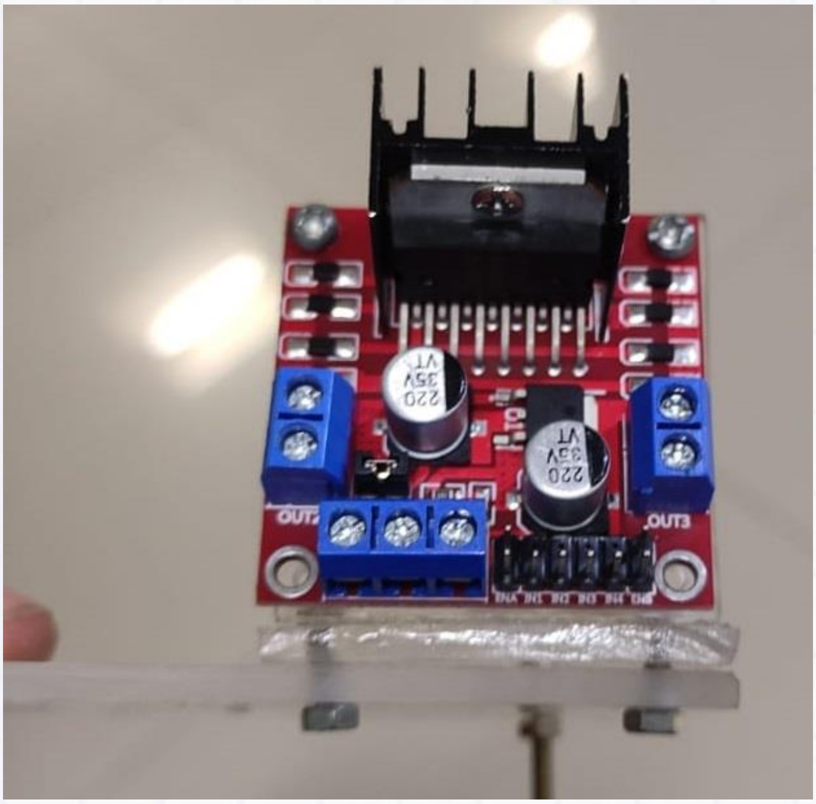
Connection diagram for the Micromouse/Car side

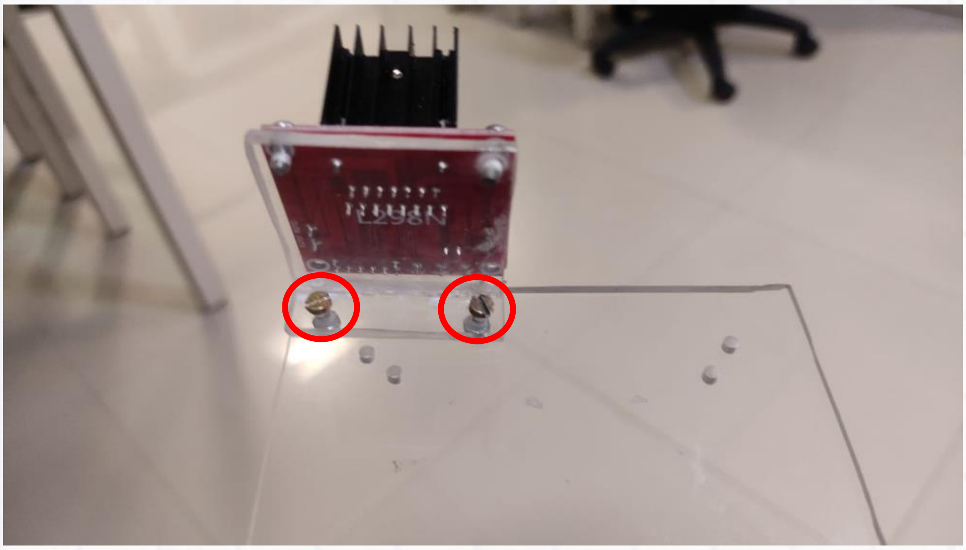


Detailed Connection Steps

Step 1

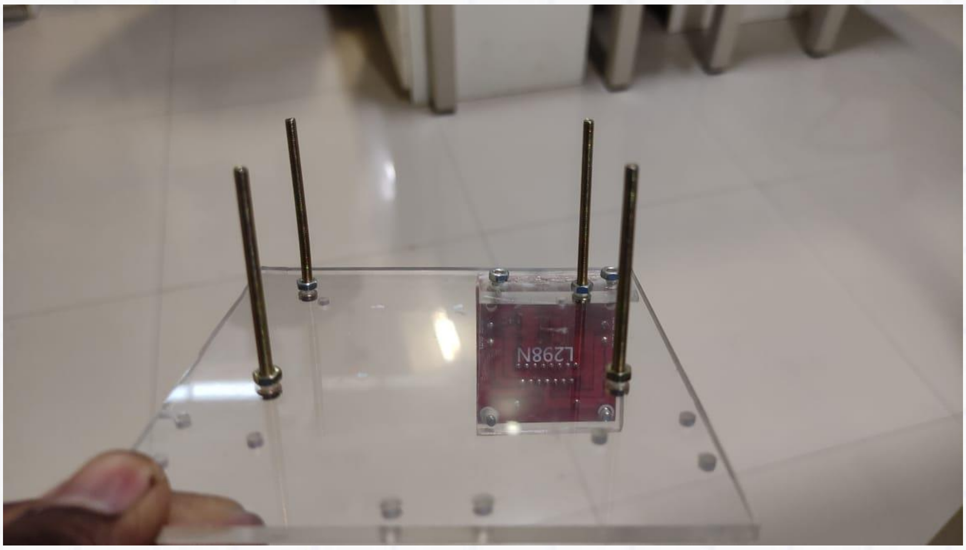
Take the L298 Motor driver board and screw it on the main acrylic sheet as shown below:





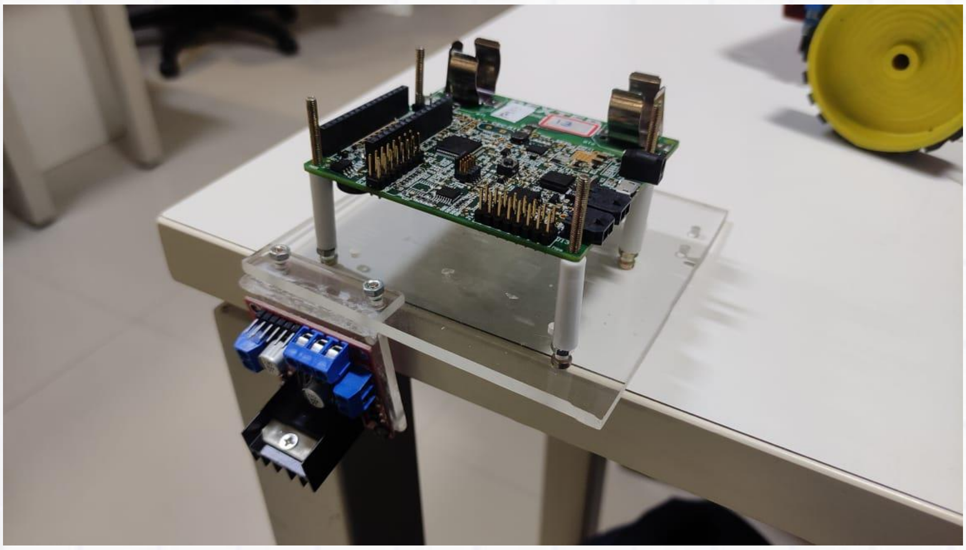
Step 2

Insert screws on main acrylic sheet in order to mount dexter board. The holes should be chosen in such a way that it matches the holes on dexter board



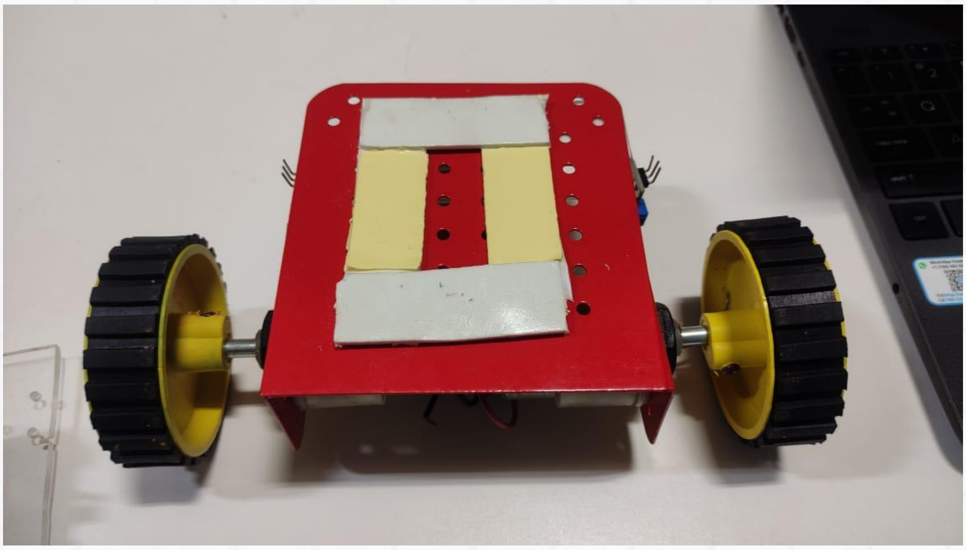
Step 3

Insert spacer, mount dexter board and screw it as below:



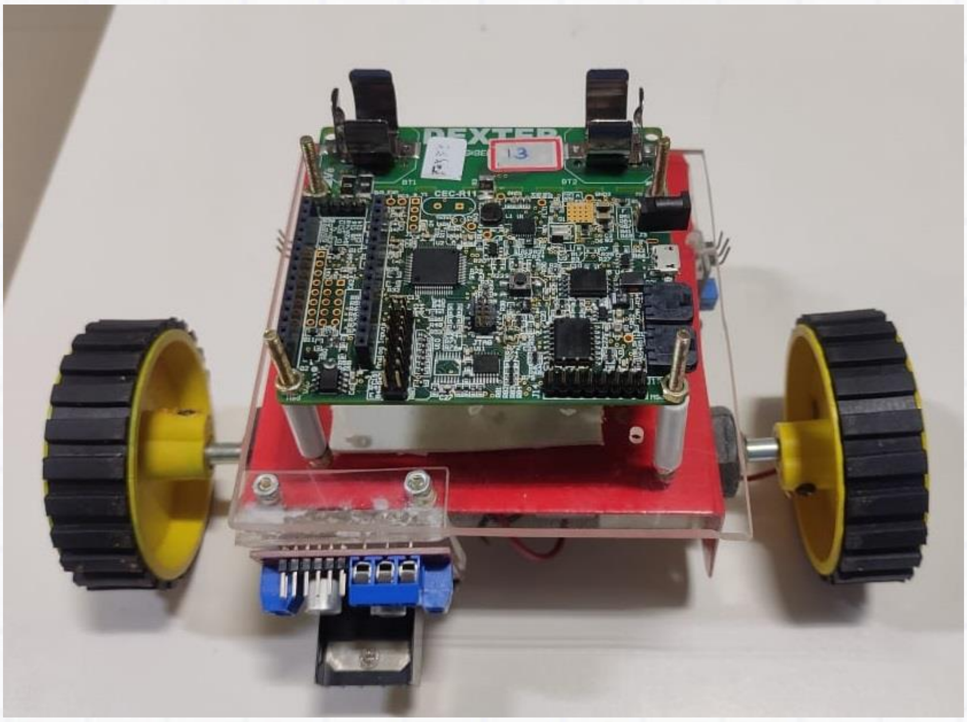
Step 4

Place a double side tape on Micromouse frame as below



Step 5

Mount the dexter board assembly on this double side tape as shown below:



Step 6

Take 6 male-to-female jumper wires and connect the pins of the L298 Driver to dexter board:

Important

- **ENB** of L298 to **PA1** of dexter board
- **IN4** of L298 to **PA2** of dexter board
- **IN3** of L298 to **PC2** of dexter board
- **IN2** of L298 to **PA4** of dexter board
- **IN1** of L298 to **PA5** of dexter board
- **ENA** of L298 to **PA0** of dexter board

Step 7

Take 4 male-to-female jumper wires and connect the pins of the bluetooth module to dexter board:

Important

- **VCC** of bluetooth module to **+5V** of dexter board
- **GND** of bluetooth module to **GND** of dexter board
- **RX** of bluetooth module to **PA9** of dexter board
- **TX** of bluetooth module to **PA10** of dexter board

Step 8

Connect LEFT & RIGHT DC motors to **OUT2** and **OUT3** ports of L298 board.

Note: The +ve and –ve terminal of a motor connection can be switched accordingly if it's found running on opposite sides at the time of final testing

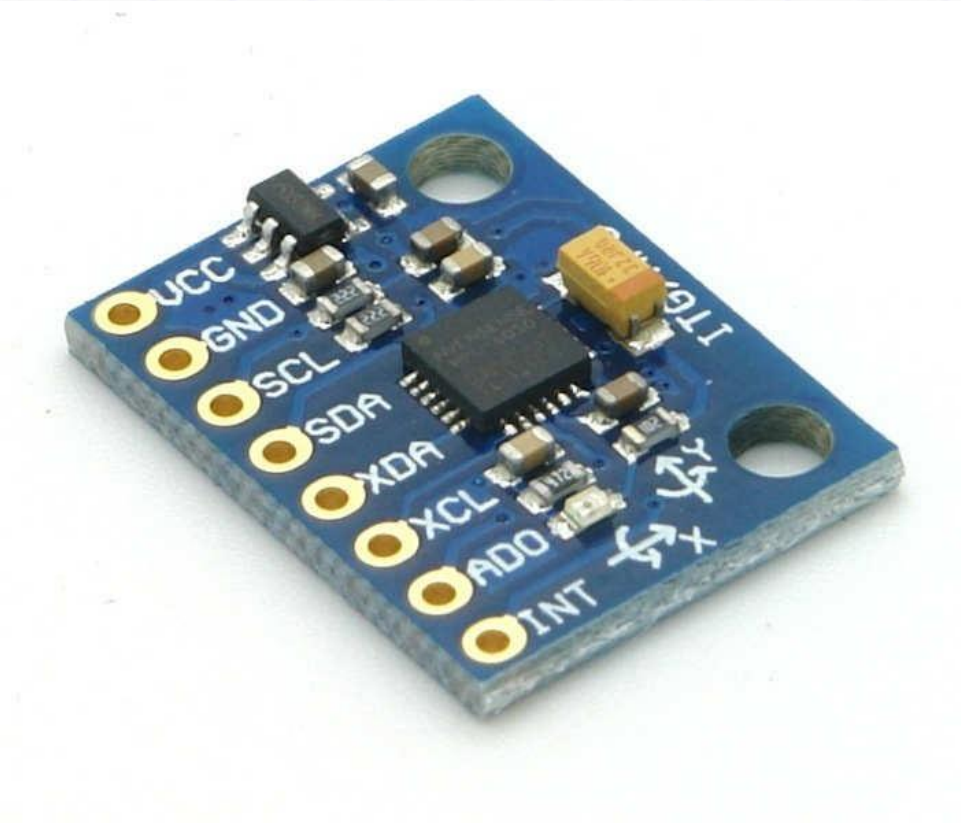
Step 9

Take 2 male to male jumper wires and connect from L298 board and dexter board:

- **12V** of L298 board to **VIN** of dexter board
- **GND** of L298 board to **GND** of dexter board

F. Components in detail

2) MPU6050 Sensor Module



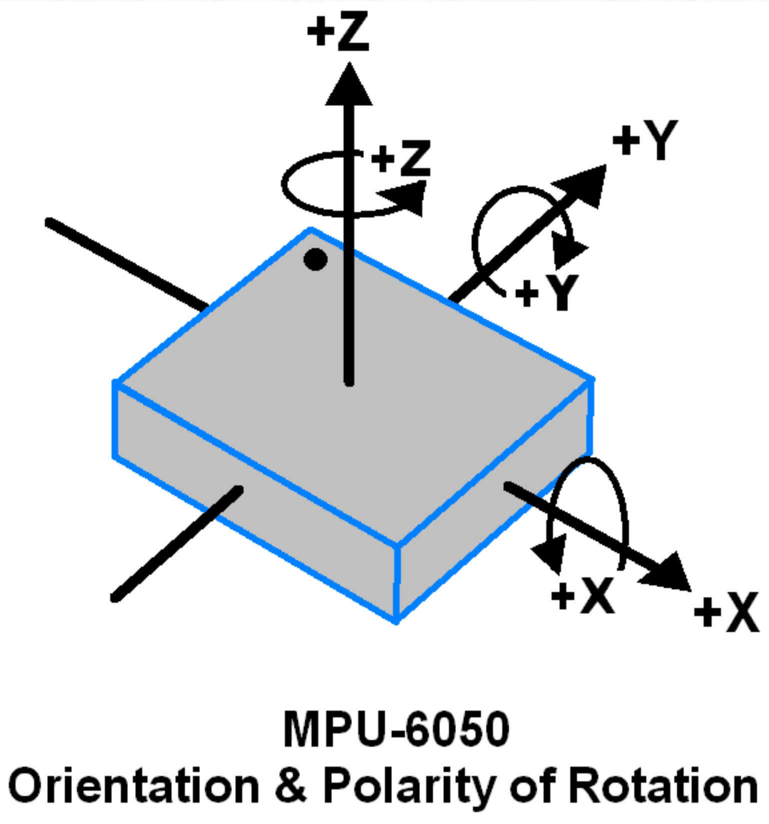
MPU6050 sensor module is complete 6-axis Motion Tracking Device. It combines 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in small package.

NOTE: Also the sensor has an additional feature of on-chip Temperature sensor. It has I2C bus interface to communicate with the microcontrollers.

It has Auxiliary I2C bus to communicate with other sensor devices like 3-axis Magnetometer, Pressure sensor etc.
If 3-axis Magnetometer is connected to auxiliary I2C bus, then MPU6050 can provide complete 9-axis Motion Fusion output.

3-Axis Gyroscope

The MPU6050 consist of 3-axis Gyroscope with Micro Electro Mechanical System(MEMS) technology. It is used to detect rotational velocity along the X, Y, Z axes as shown in below figure.

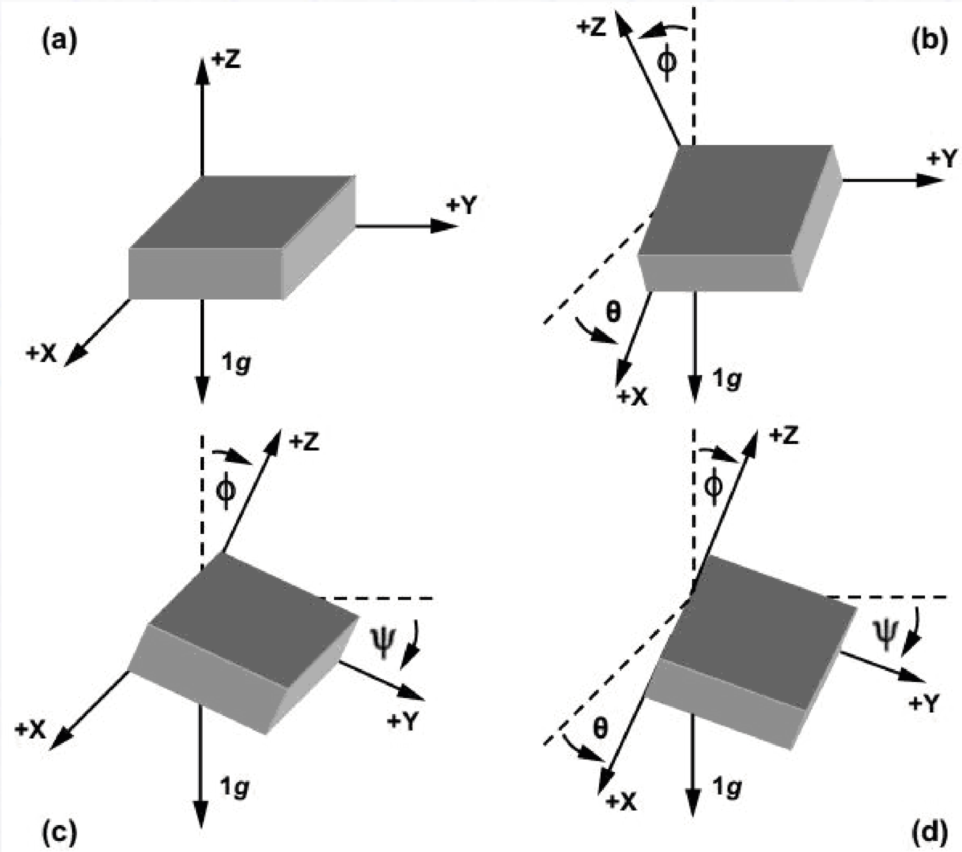


- When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a MEM inside MPU6050.

- The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate.
- This voltage is digitized using 16-bit ADC to sample each axis.
- The full-scale range of output are +/- 250, +/- 500, +/- 1000, +/- 2000.
- It measures the angular velocity along each axis in degree per second unit.

<u>3-Axis Accelerometer</u>

The MPU6050 consist 3-axis Accelerometer with Micro Electro Mechanical (MEMs) technology. It used to detect angle of tilt or inclination along the X, Y and Z axes as shown in below figure.



- Acceleration along the axes deflects the movable mass.

- This displacement of moving plate (mass) unbalances the differential capacitor which results in sensor output. Output amplitude is proportional to acceleration.
- 16-bit ADC is used to get digitized output.
- The full-scale range of acceleration are +/- 2g, +/- 4g, +/- 8g, +/- 16g.
- It measured in g (gravity force) unit.
- When device is placed on flat surface it will measure 0g on X and Y axis and +1g on Z axis.

The MPU-6050 module has 8 pins as listed below:

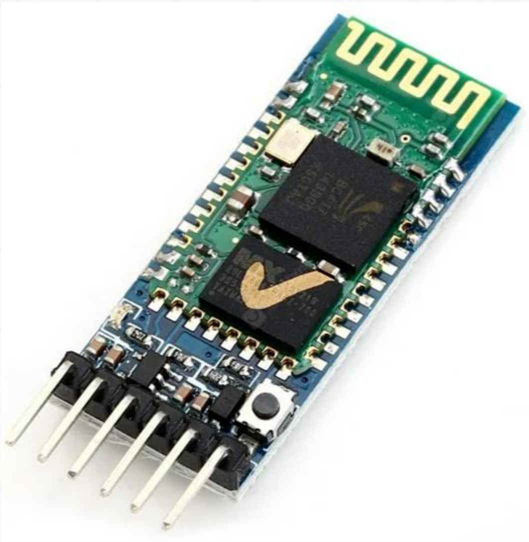
9. **INT:** Interrupt digital output pin.
10. **AD0:** I2C Slave Address LSB pin. This is 0th bit in 7-bit slave address of device. If connected to VCC then it is read as logic one and slave address changes.
11. **XCL:** Auxiliary Serial Clock pin. This pin is used to connect other I2C interface enabled sensors SCL pin to MPU-6050.
12. **XDA:** Auxiliary Serial Data pin. This pin is used to connect other I2C interface enabled sensors SDA pin to MPU-6050.
13. **SCL:** Serial Clock pin. This pin to connected to microcontrollers SCL pin.
14. **SDA:** Serial Data pin. This pin is connected to microcontrollers SDA pin.
15. **GND:** Ground pin. This requires a ground connection.
16. **VCC:** Power supply pin. This requires a +5V DC supply.

For detailed theory refer this link

> **https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/**

2) Bluetooth Module HC05

HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.



- It is utilized in numerous consumer applications, including wireless keyboards, wireless mice, wireless headsets, and game controllers.
- Depending on the transmitter and receiver, atmosphere, geography, and urban settings, the range can go up to about 100m.
- The established protocol used to create wireless Personal Area Networks is IEEE 802.15.1. (PAN). Data is transmitted over the air using frequency-hopping spread spectrum (FHSS) radio technology.

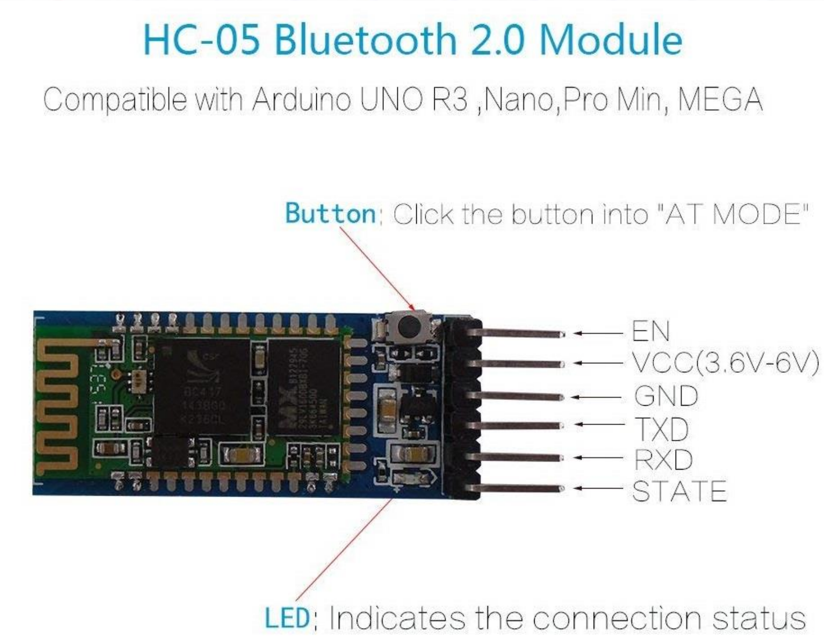
- It communicates with devices using serial communication. It uses a serial port to connect with the microcontroller (USART).

Pin Description

All serial-capable devices can connect with one another with Bluetooth connectivity using Bluetooth serial modules..

It has 6 pins,

1. **Key/EN:** It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.



HC-05 module has two modes,

a. **Data mode:** Exchange of data between devices.

b. **Command mode:** It uses AT commands which are used to change setting of HC-05. To send these commands to module serial (USART) port is used.

2. **VCC:** 5 V or 3.3 V can be supplied to this Pin.

3. **GND:** Ground Pin.

4. **TXD:** This pin transmits Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)

5. **RXD:** Can Receive data serially (received data will be transmitted wirelessly by Bluetooth module).

6. **State:** It tells whether module is connected or not.

HC-05 module Information

- The red LED on the HC-05 shows the connection state, whether Bluetooth is active or not. This red LED continuously blinks before being connected to the HC-05 module. Its blinking reduces to two seconds when it connects to any other Bluetooth device..
- 3.3 V is needed for this module. Since the module contains a built-in 5 to 3.3 V regulator, we may also connect a 5V supply voltage.
- There is no need to change the transmit level of the HC-05 Bluetooth module because it has a 3.3 V level for RX/TX and the microcontroller can detect that level. But if the microcontrollers operating voltage level is different ,we must use a level shifter for the operation

Mode of operation

Data Mode:

We can send data from Smartphone/PC terminal to HC-05 Bluetooth module and vice versa.

Smartphone must have a Bluetooth terminal application installed on it, inorder to send and receive data from the HC-05 Bluetooth module. Applications for Bluetooth terminals are available in the corresponding app stores for Android and Windows.

In PC we can use putty/Teraterm to communicate to HC-05 using a USB-serial module.

Command Mode

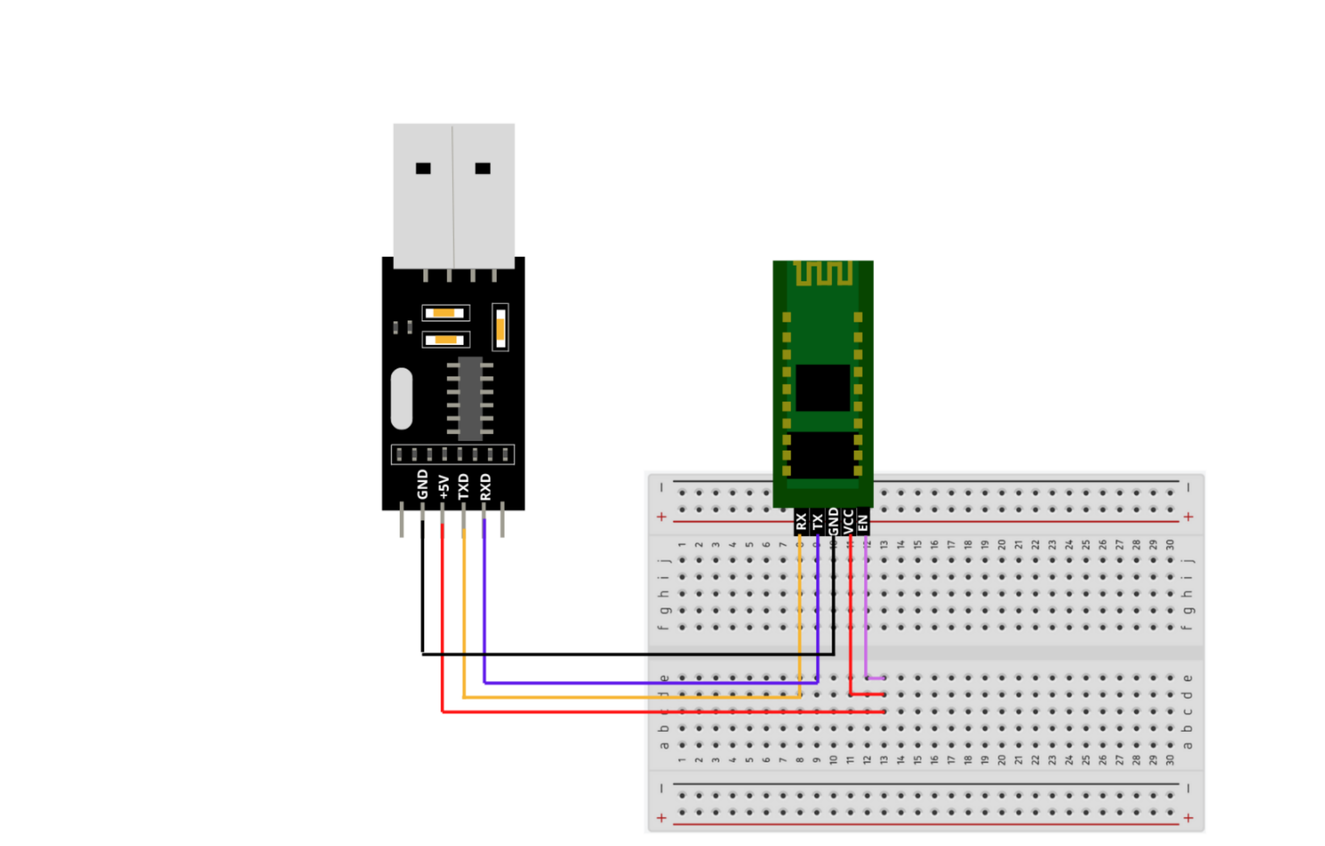
- We need to be in command mode need to modify the HC-05 Bluetooth module's parameters, such as the connection password, baud rate, name of the Bluetooth device, etc.
- The HC-05 contains AT instructions to accomplish this.
- Connect the "Key" pin to High to utilise the HC-05 Bluetooth module in AT command mode (VCC).
- Default The HC-05's command mode baud rate is 38400bps.
- The AT commands listed below are frequently used to modify Bluetooth module settings.
- To transfer these commands, we must connect the HC-05 Bluetooth module to the PC via a serial to USB converter, then use the serial terminal on the PC to do so.

Command	Description	Response
AT	Checking communication	OK
AT+PSWD=XXXX	Set Password e.g. AT+PSWD=4567	OK
AT+NAME=XXXX	Set Bluetooth Device Name e.g. AT+NAME=MyHC-05	OK
AT+UART=Baud rate, stop bit, parity bit	Change Baud rate e.g. AT+UART=9600,1,0	OK
AT+VERSION?	Respond version no. of Bluetooth module	+Version: XX OK e.g. +Version: 2.0 20130107 OK
AT+ORGL	Send detail of setting done by manufacturer	Parameters: device type, module mode, serial parameter, passkey,etc.

Steps to configure the HC05 bluetooth module as Master and Slave

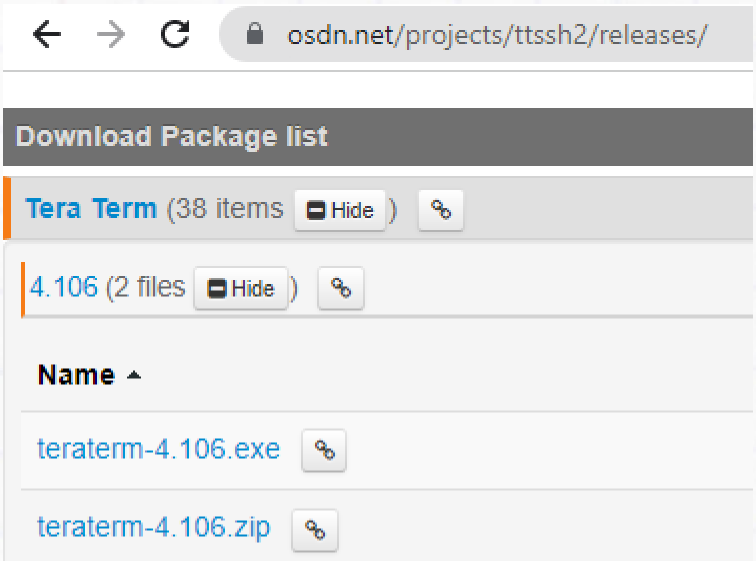
In this project, we need one HC-05 to transmit and other to receive the data. There can be a master device and slave device.
To configure we need to enter AT Command mode.
Follow the below steps for configuring two HC-05 as a master and a slave

Connection Diagram for AT Command mode

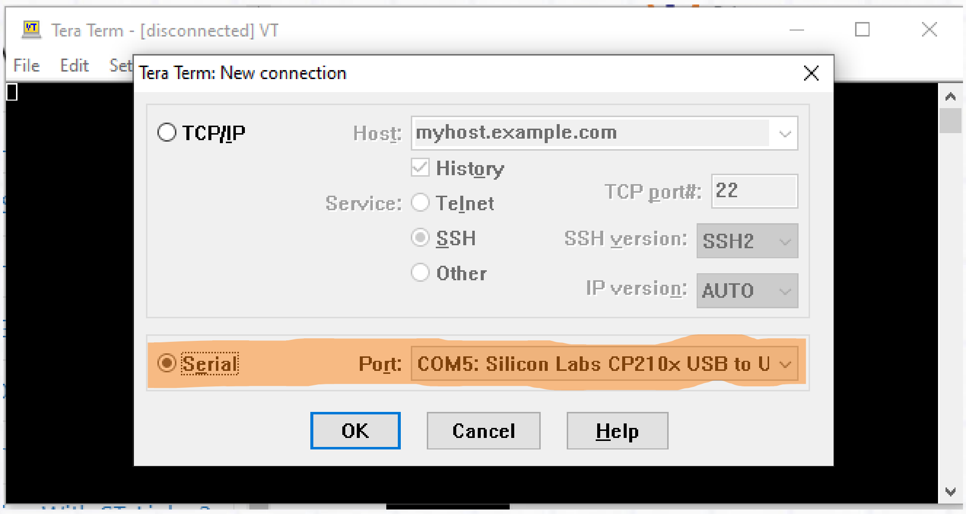


Steps to convert HC-05 as a Slave device

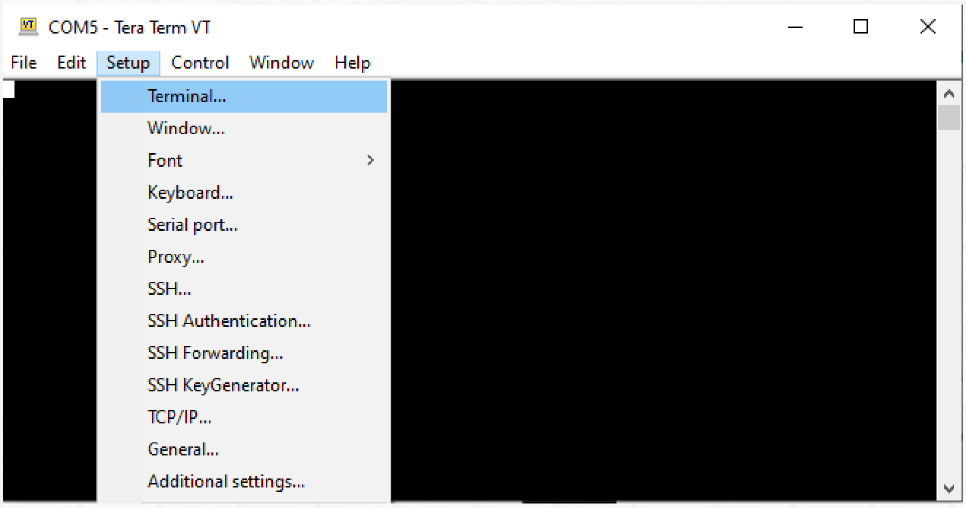
1. Download and install ‘Teraterm’ from given files/web and run the software



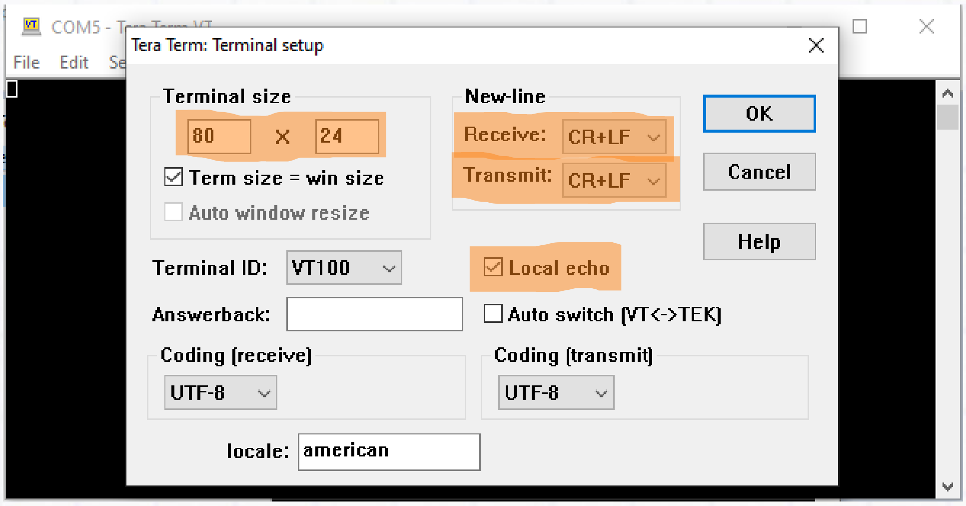
2. Connect the USB-TTL converter to PC, Select the Serial Port



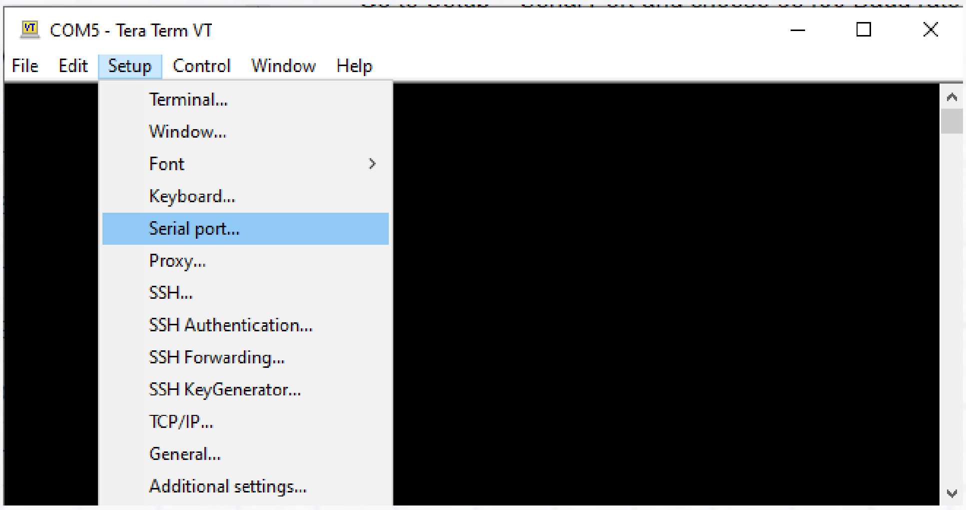
3. Select Setup->Terminal



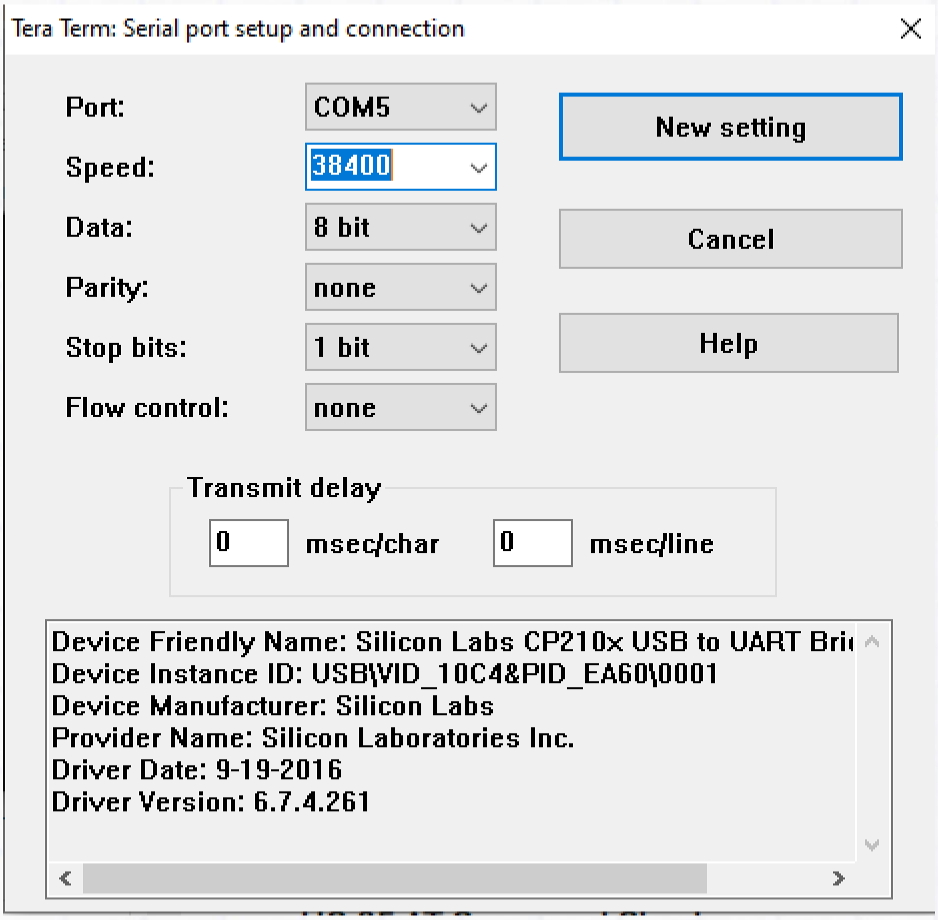
4. Change the settings as below



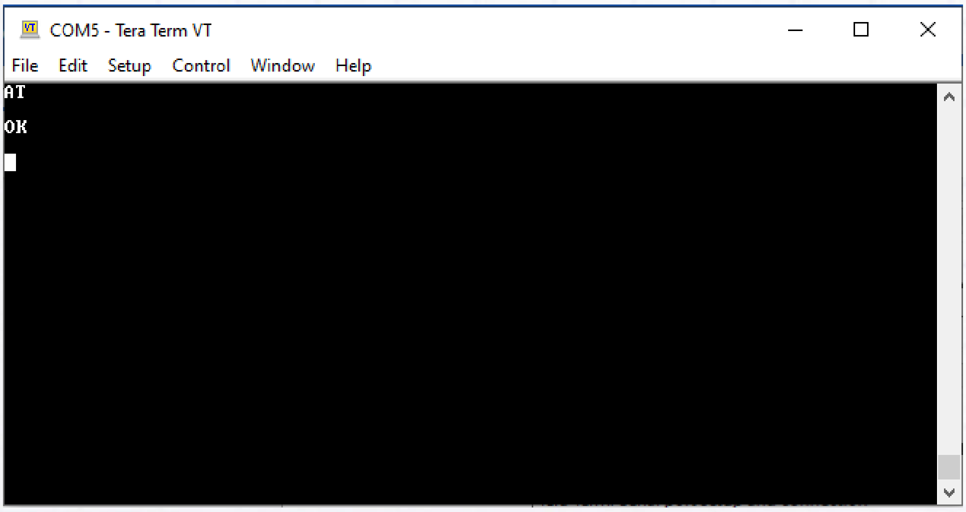
5. Select Setup->Serialport



6. Change the baud rate to 38400



7. Type ‘**AT**’ on the terminal. Check whether the response is ‘**OK**’



8. Type the following commands to set the device as a Slave

- **AT+RMAAD** (To clear any paired devices)
- **AT+ROLE=0** (To set it as slave)
- **AT+ADDR?** (To get the address of this HC-05, write down the address since it is needed while configuring the master device configuration)

Here the address is **0021:11:01385E,** for different devices it can vary

- **AT+UART=38400,0,0** (To fix the baud rate at 38400)
- Make sure all the commands are replied '**OK**'
- Now you have configured the device as a Slave
- Mark it as Slave and now connect the other device to configure it as a master in the same connection configuration

<u>Steps to convert HC-05 as a Master device</u>

1. Type **'AT'** and check the **'OK'** reply
2. Type the following commands in Teraterm

- **AT+RMAAD** (To clear any paired devices)
- **AT+ROLE=1** (To set it as master)
- **AT+CMODE=0** (To connect the module to the specified Bluetooth address and this Bluetooth address can be specified by the binding command)
- **AT+BIND=xxxx,xx,xxxxxx** (To bind the slave)

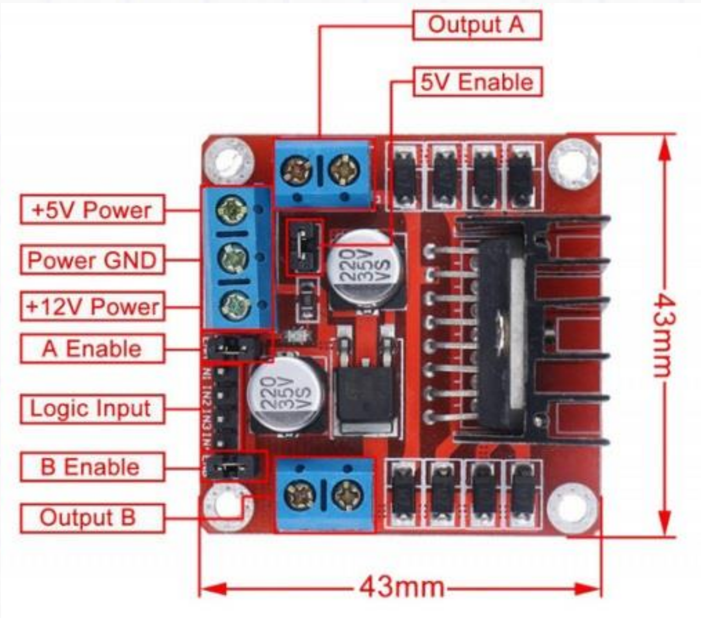
eg) type AT+BIND= 0021,11,01385E with your respective address of the slave. Note the commas instead of colons given by the slave module.

- **AT+UART=38400,0,0** (To fix the baud rate at 38400)

3. Now you have configured master and slave devices. If you power the devices you will see the LEDs blinking with a 2s interval, indicating a successful pair

3) L293D Dual motor driver module

L298 Based Motor Driver Module is a high power motor driver perfect for driving DC Motors and Stepper Motors. It uses the popular L298 motor driver IC and has the onboard 5V regulator which it can supply to an external circuit. It can control up to 4 DC motors, or 2 DC motors with directional and speed control



G. Implementing the gesture Micromouse project

Transmitter part (Master /Wrist side)

Variables & Functions

6) sens_min, sens_max

Inside main.c file, this variable can be found. Default value of sens_max=250 & sens_min=100. Adjusting the values, will change the sensitivity of the sensor and hence micromouse movement.

7) MPU6050_Init()

This function initializes the IMU sensor by checking for power issues, setting the data rate, and also setting up the gyro and accelerometer configurations.

8) MPU6050_CALIB()

This function calibrates the minimum and maximum values of x, y & z to the sensor's current position. It reads the current x, y & z values and helps in adapting to the current sensor position.

9) MPU6050_Read_Gyro()

Reads the gyro values ie, the angles of x ,y, z with respect to the calibrated position

10) Gyro2Mov()

This function is used to convert the gyro values obtained from gyro sensor to movement direction

11) Mov_value

This variable is used to store the movement direction value obtained from gyrosensor.

12) Send2Blu(Mov_value)

This function is used to send the direction value ie, **Mov_value** to the Bluetooth transmitter for transmission.

Receiver part (MicroMouse /Car side)

Variables & Functions

1)Blu_receive()

This function receives data from bluetooth receiver. The data that we receive on the receiver is what was earlier sent by the bluetooth transmitter

3) Blu_val

This variable could store the value received from the function Blu_receive()

4) Motor_init(x)

This function initializes the motor and can be used to initialise the LEFT and right motor

Eg) Motor_init(Motor_L);

Motor_init(Motor_R);

5) Motor_stop(x)

This function stops the motor

Eg) Motor_stop(Motor_R);

Motor_stop(Motor_L)

6) **Move_forward**(x)

This function moves the microMouse in forward direction at **'x'** speed

Eg) **Move_forward(100);** - This moves the MicroMouse at speed 100

7) **Move_backward**(x)

This function moves the microMouse in backward direction at **'x'** speed

Eg) **Move_backward(100)** - This moves the MicroMouse at speed 100

8) **Move(forward,x,forward,x)**

This function can make the left and right motors move at **'x'** speed

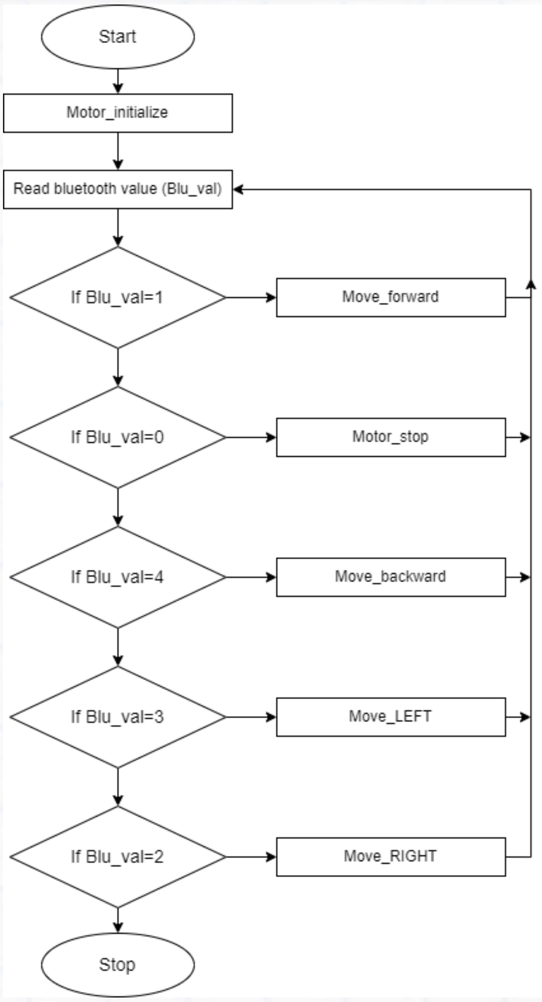
Eg) **Move(forward,5,forward,100)** - This turns the Micromouse left due to relative speeds, since the left motor is run at speed 5 and right motor at speed 100

Implementing the Code

Receiver side (MicroMouse side)

Flowchart of the Code

The flowchart diagram represents the flow of the code for the gesture device. Along with the flowchart the program part can be easily implemented.



Implementation steps

4) Download the Project Workspace file '**Gesture_device_micromouse_receiver.zip**' given in the project page on the Build Club website.

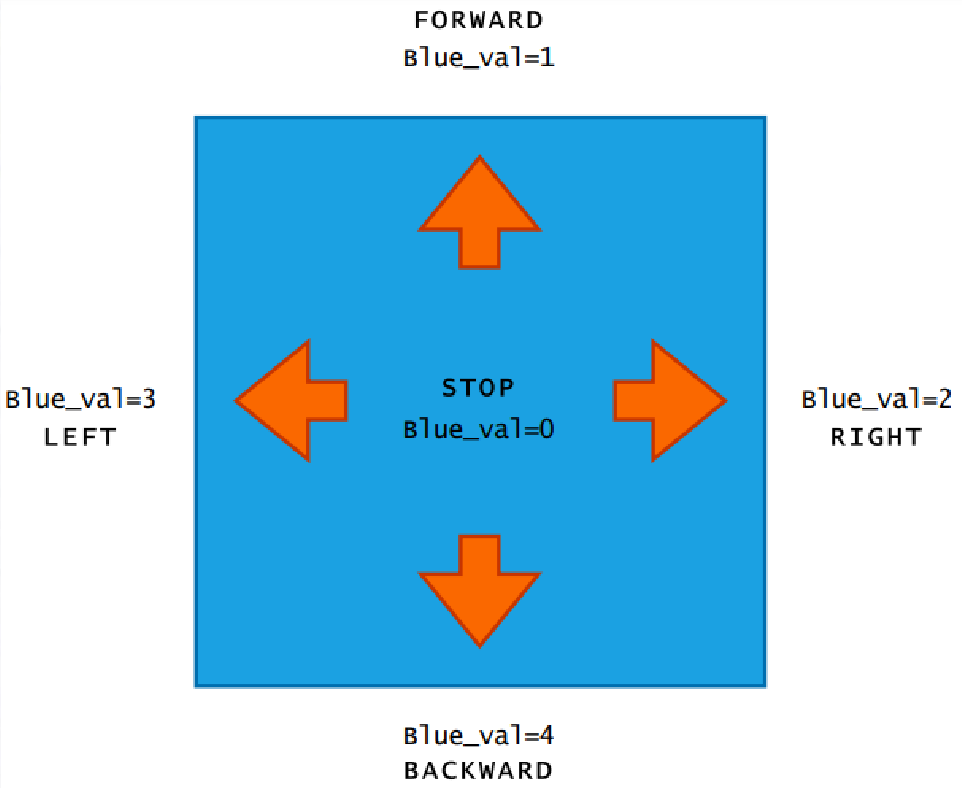
5) In the **Workspace** folder in your C: drive, create a new folder named **'Gesture_device_micromouse_receiver.**.

6) Then i) Launch the STM IDE, ii) Select the **Gesture_device_micromouse_receiver.**' folder as workspace, iii) Import the ZIP file **Gesture_device_micromouse_receiver.**', iv) Navigate to **main.c**

Refer the flowchart and functions above to implement the code in **App** function in main.c file. Follow the steps below:

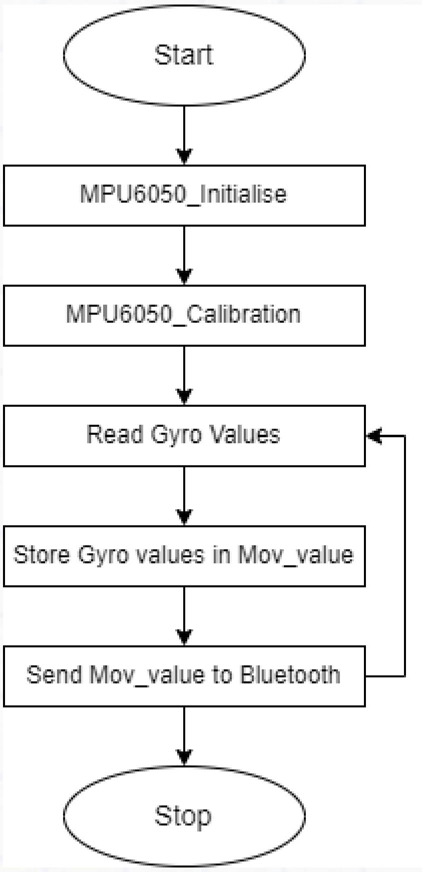
7) Initialize both the left and right side motors using the motor initialize **Motor_init()**
8) Now we have to repeatedly read gyro values and execute action using a **while()** loop
9) Use **Blu_receive()** function and store the value in **Blu_val** variable
10) Using an **if()** condition check the **Blu_val** and execute different motor actions such as **Move_forward(), Move_backward()** , etc as per below flowchart
11) Upload the code to dexter board

Micromouse movements



Transmitter side (Wrist side)

Flowchart of the Code



Implementation steps

1) Download the Project Workspace file **'Gesture_device_micromouse_transmitter.zip'** given in the project page on the Build Club website.
2) In the **Workspace** folder in your C: drive, create a new folder named **'Gesture_device_micromouse_ transmitter..**
3) Then i) Launch the STM IDE, ii) Select the **Gesture_device_micromouse_transmitter.**' folder as workspace, iii) Import the ZIP file **Gesture_device_micromouse_ transmitter.**', iv) Navigate to **main.c**

Refer the flowchart and functions above to implement the code in **App** function in main.c file. Follow the steps below:

4) Initialize the MPU6050 sensor using the **MPU6050_Init()** function and calibrate the sensor with **MPU6050_CALIB()** function
5) After calibration function, the values of the Gyro should be read continuously. You can use **MPU6050_Read_Gyro()** inside a **while()** loop
6) By calling function **Gyro2Mov()** ,the gyro values can be converted to corresponding movement values
7) Store the movement values in **Mov_value**

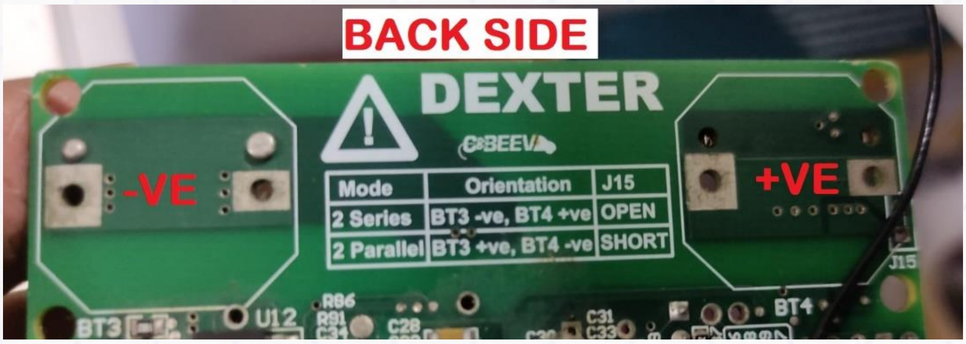
Eg) Mov_value = Gyro2Mov();

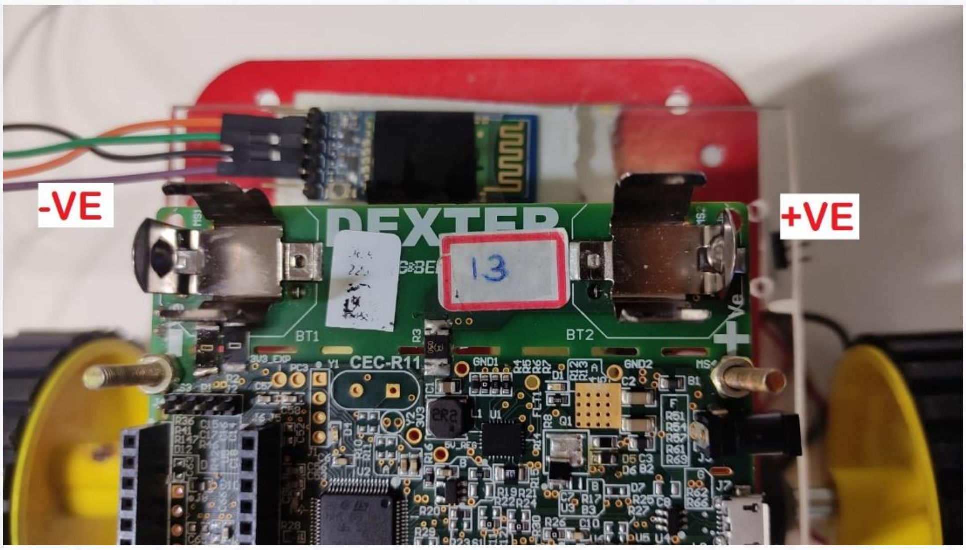
8) Send the Mov_value to bluetooth with **Send2Blu(Mov_value)** function

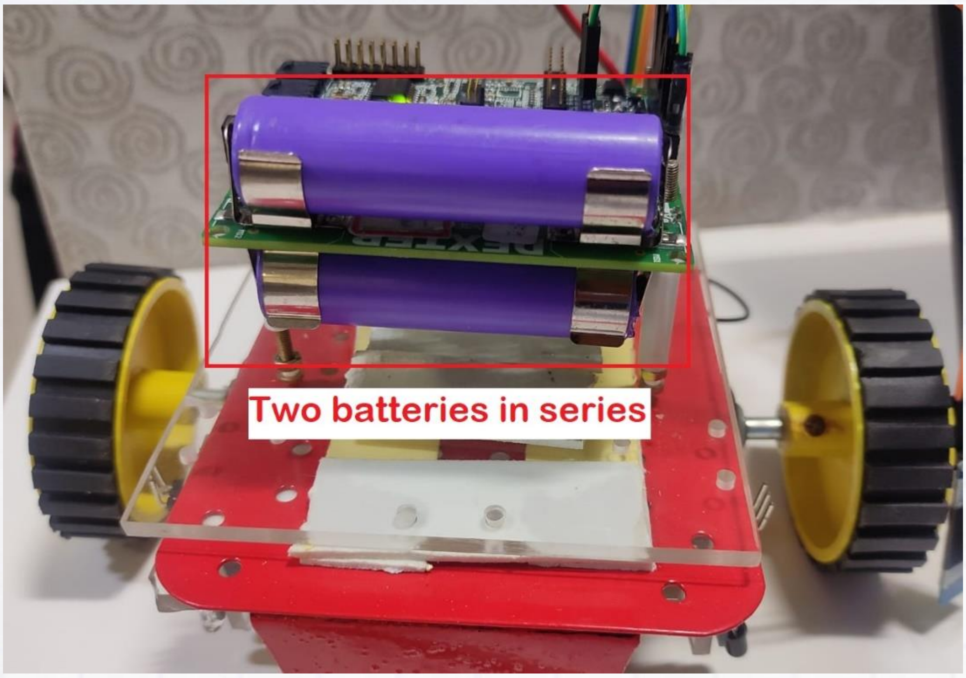
After writing these codes in STM32 IDE. The code can now be uploaded to the Dexter board by hitting 'Run'.

<u>Running the MicroMouse with gesture control</u>

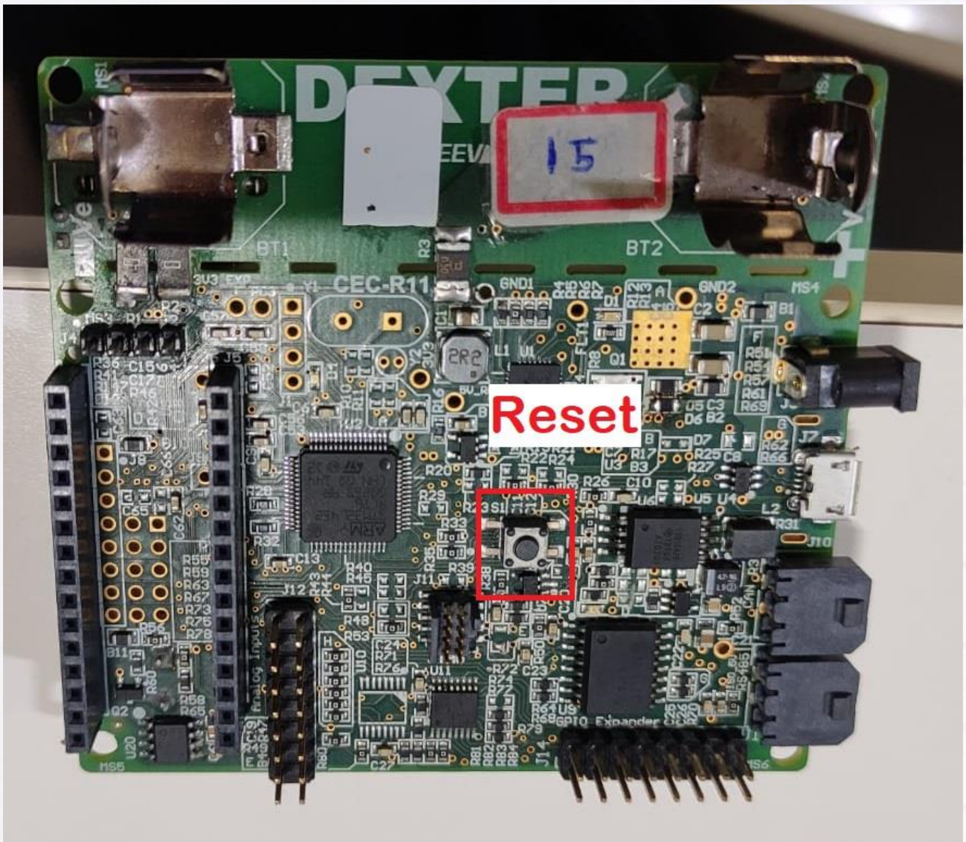
1) Take the Micromouse and power it up using two batteries as per below polarities (ie in series connection)





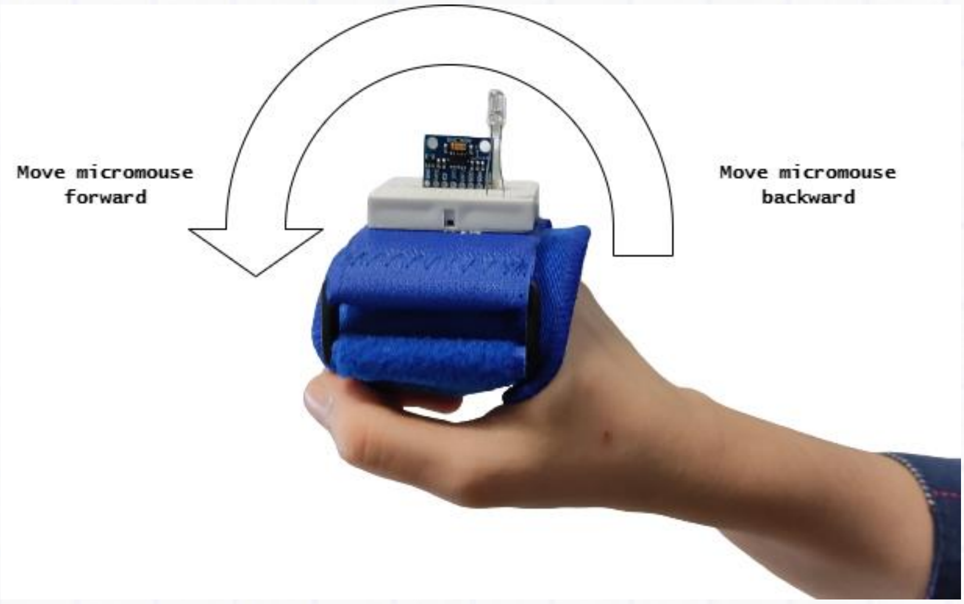


2) Power up the dexter board in wrist side also with batteries.
3) Check whether two bluetooth modules are powered up. If the LEDs in both the bluetooth modules are blinking with a 2 seconds interval ,it means both of them are paired.
4) Place hand in a steady position, press the **reset** button in your wrist side dexter board so that sensor initialization and calibration is carried out in your current hand position. The calibration LED will turn **off** indicating the completion of calibration

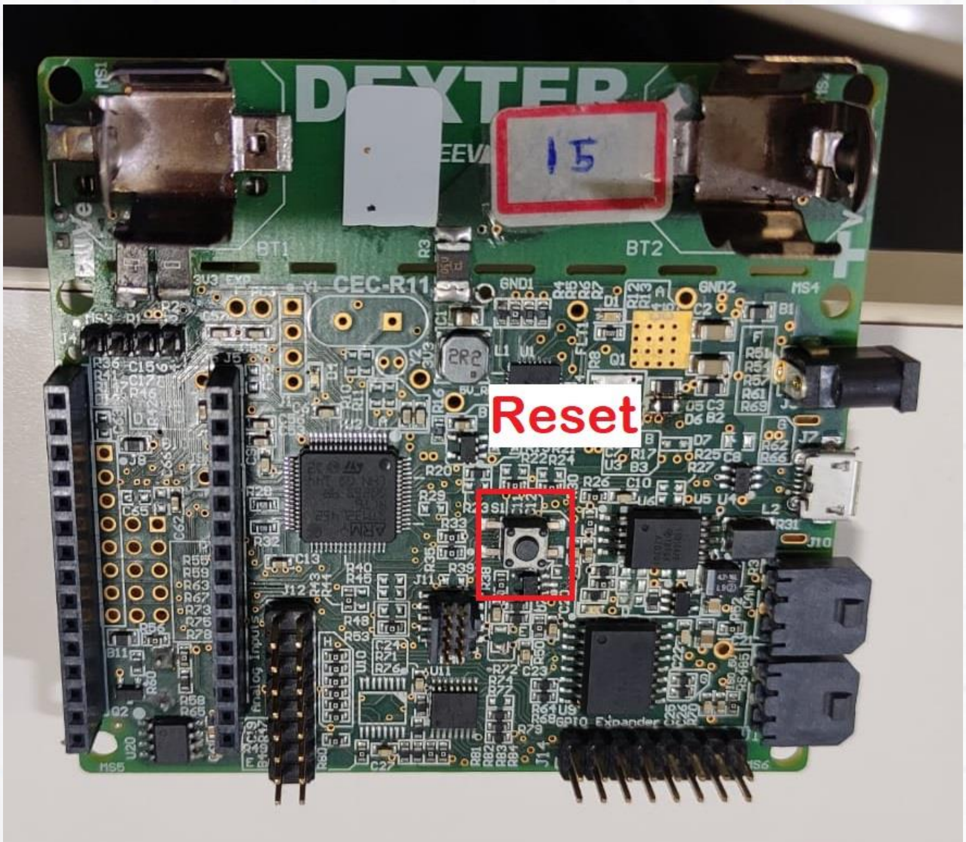


6) Now slowly & gently move the wrist device forward, backward ,left and right.

Hand orientation



7) Check whether the MicroMouse is moving as per the hand direction
8) If you feel that the you need a different hand orientation or the current mean position needs to be changed. Move to new position, again press the **reset** button ,place the hand steady in your required position until the LED turns off.



9) Now you can move the MicroMouse in any direction you like, wirelessly.

H. Activities

1.Try changing the **sens_max** and **sens_min** variable value between **(0 –1000)** ranges to change the sensitivity suitable for every team mate. Check your mate is able to control the MicroMouse with ease with this value.

2. Try to move the Micromouse between tables/paths

3. Make a video presenting, the build of gesture device, how you completed the activities and challenges. Share it with the Build Club Community on the Discord Server!