Self Balancing Robot : Into the Embedded Systems

Hands On Workshop

Indian Institute of Technology, Bombay

February 15, 2024

IIT BOMBAY

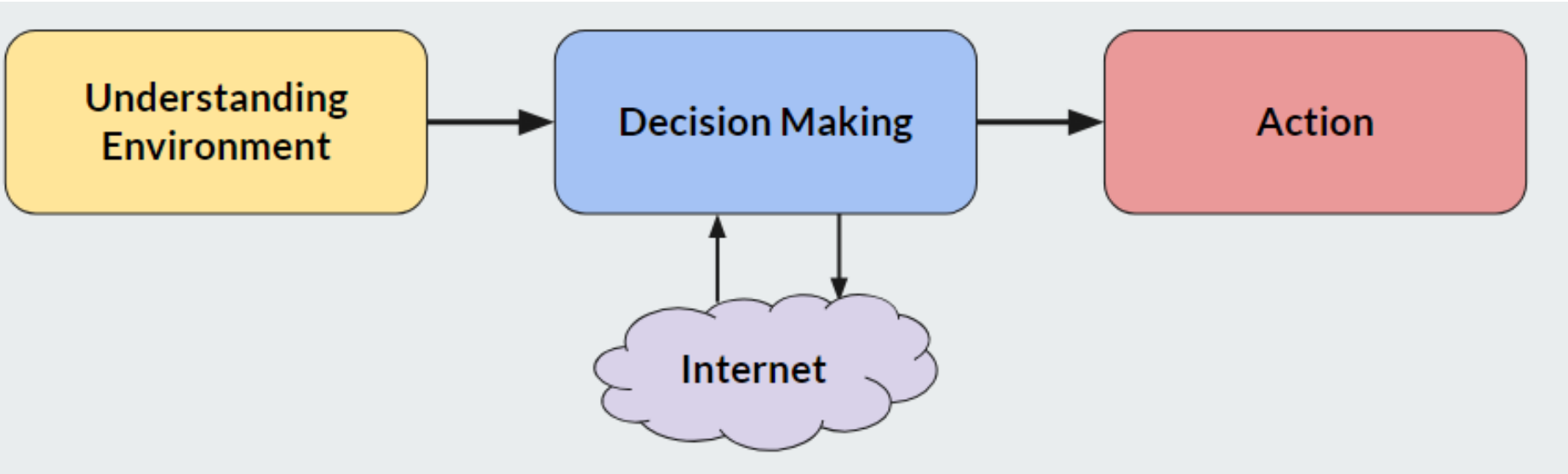
e-Yantra Engineering a better tomorrow

Agenda for Discussion

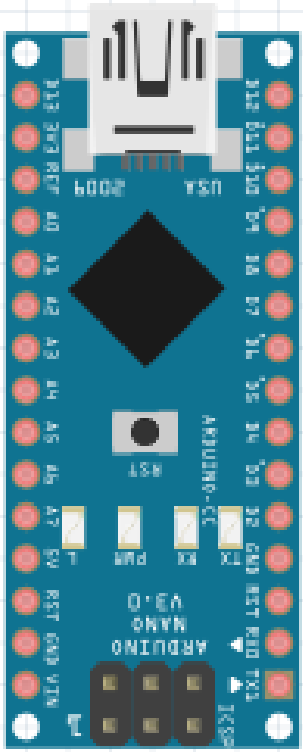
1. Embedded Systems
2. Why Arduino?
3. Arduino Nano
4. Sensors & Actuators
5. Summary

Embedded Systems

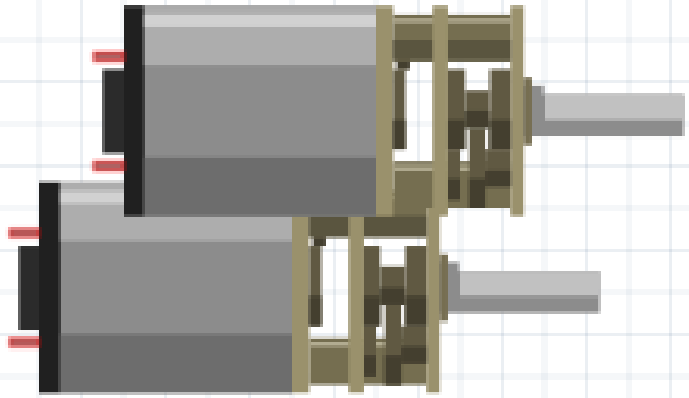
Embedded systems are compact computing systems dedicated to performing specific tasks.



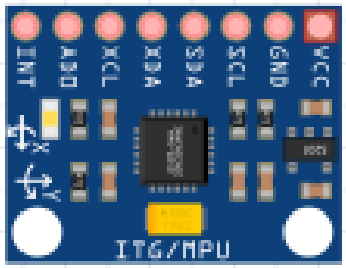
Overview of our System



Arduino Nano



N20 Motors with Encoders



MPU6050

Arduino

Arduino is an open-source hardware development board that can be used. It comes up with a simplified IDE to program their boards.

Embedded System made easy.

- Libraries & APIs
- Wide Community & Forum
- Development Boards

Common Development boards from the Arduino are.

- Arduino UNO
- Arduino Nano
- Arduino Mega

IIT BOMBAY

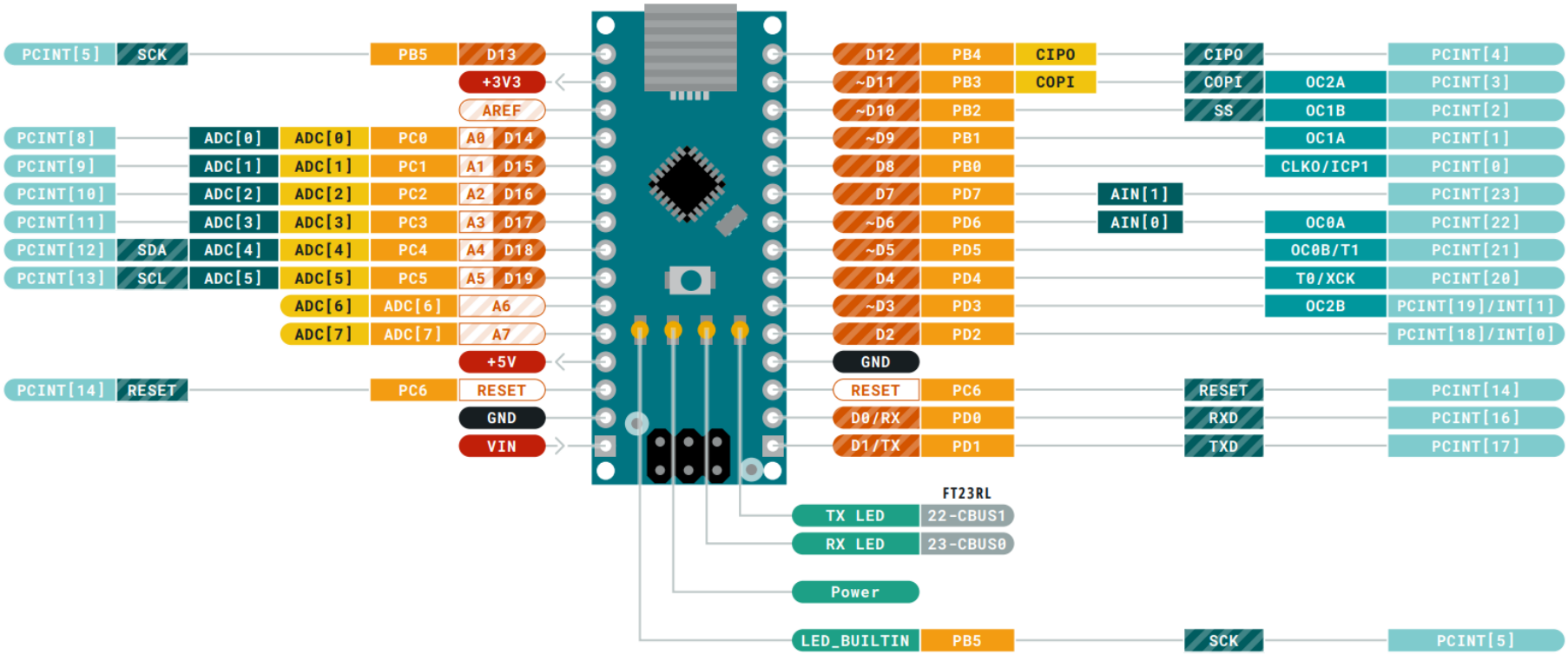
e-yantra Engineering a better tomorrow

Arduino Nano

Features

- Micro-controller - AtMega328p
  - Operating Voltage - 5V
  - Analog Input Pins - 8
  - Digital I/O Pins - 22 (PWM - 6)
  - Flash Memory - 32KB
  - SRAM - 2KB
  - Communication - UART, SPI, I2C
- Clock Speed - 16MHz
- Voltage Regulator
- In-System Programmer (ISP)

Nano's Pin Diagram



Source:
`https://content.arduino.cc/../assets/Pinout-NANO_latest.pdf`

IIT BOMBAY

e-Yantra Engineering a better tomorrow

Embedded System made easy..

Embedded C

```c
#include <avr/io.h>

int main() {
    // Set Pin D2 as output
    DDRD |= (1 << PD2);

    // Set Pin D3 as input
    DDRD &= ~(1 << PD3);

    while(1) {
        // Your main code loop here
    }

    return 0;
}
```

Arduino

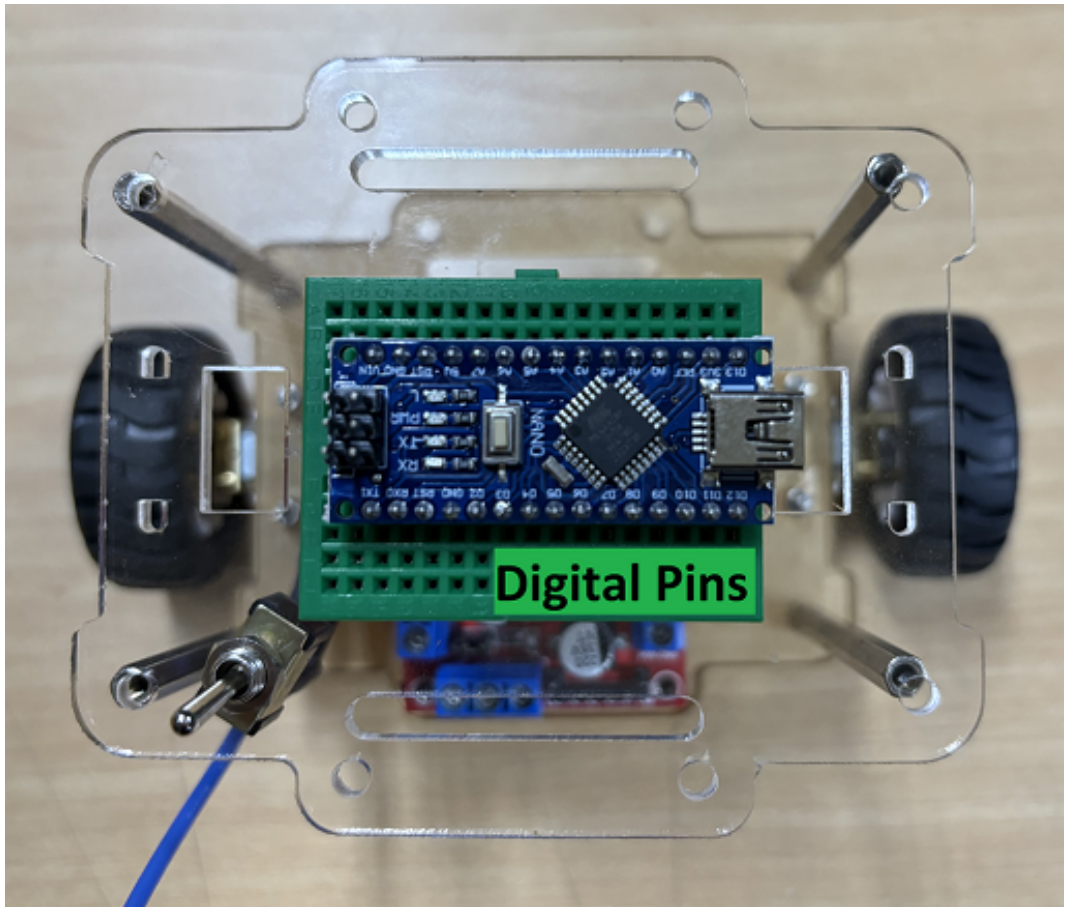
```c
void setup() {
    // Set Pin 2 as output
    pinMode(2, OUTPUT);

    // Set Pin 3 as input
    pinMode(3, INPUT);
}

void loop() {
    // Your main code loop here
}
```

IIT BOMBAY

eYantra Engineering a better tomorrow

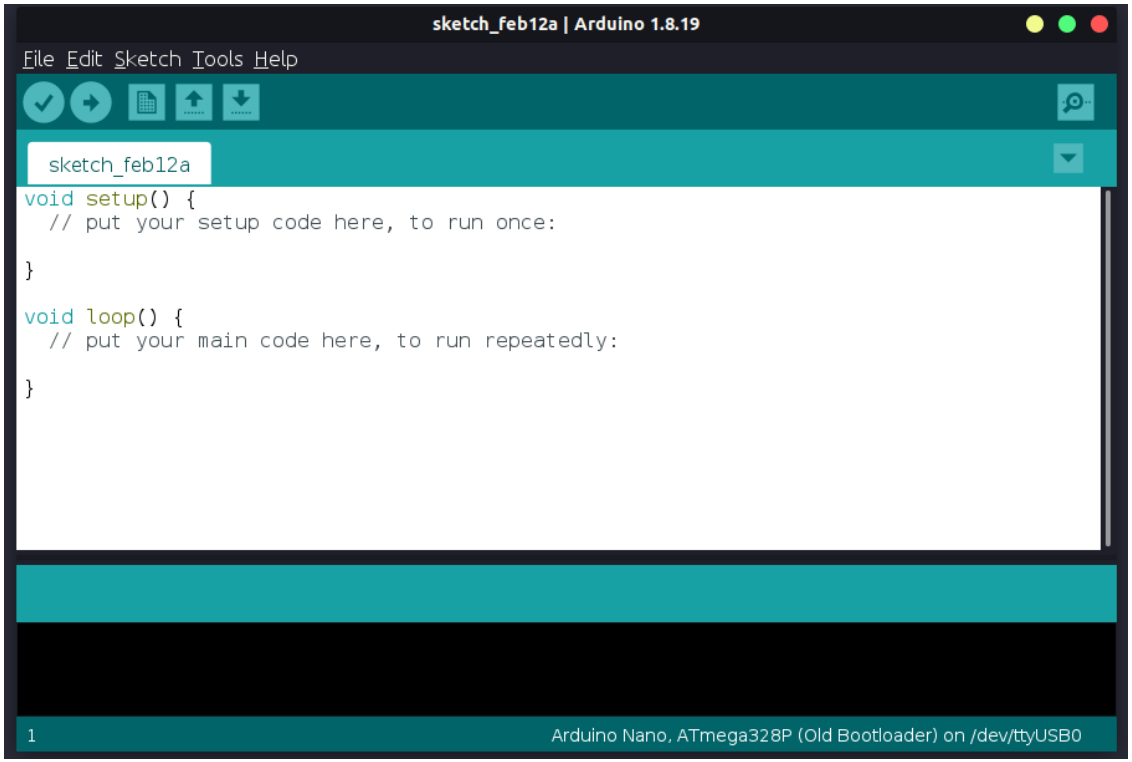
Assembly

Place your Arduino Nano on the breadboard as shown below and Stick the Breadboard on one of the acrylic sheets as shown below.



using Arduino IDE

- Connect your Arduino Nano Board with the Computer using USB Cable
- Open Arduino IDE
- Go to "Tools ⇒ Board ⇒ Arduino Nano"
- Go to "Tools ⇒ Port ⇒ COMx"

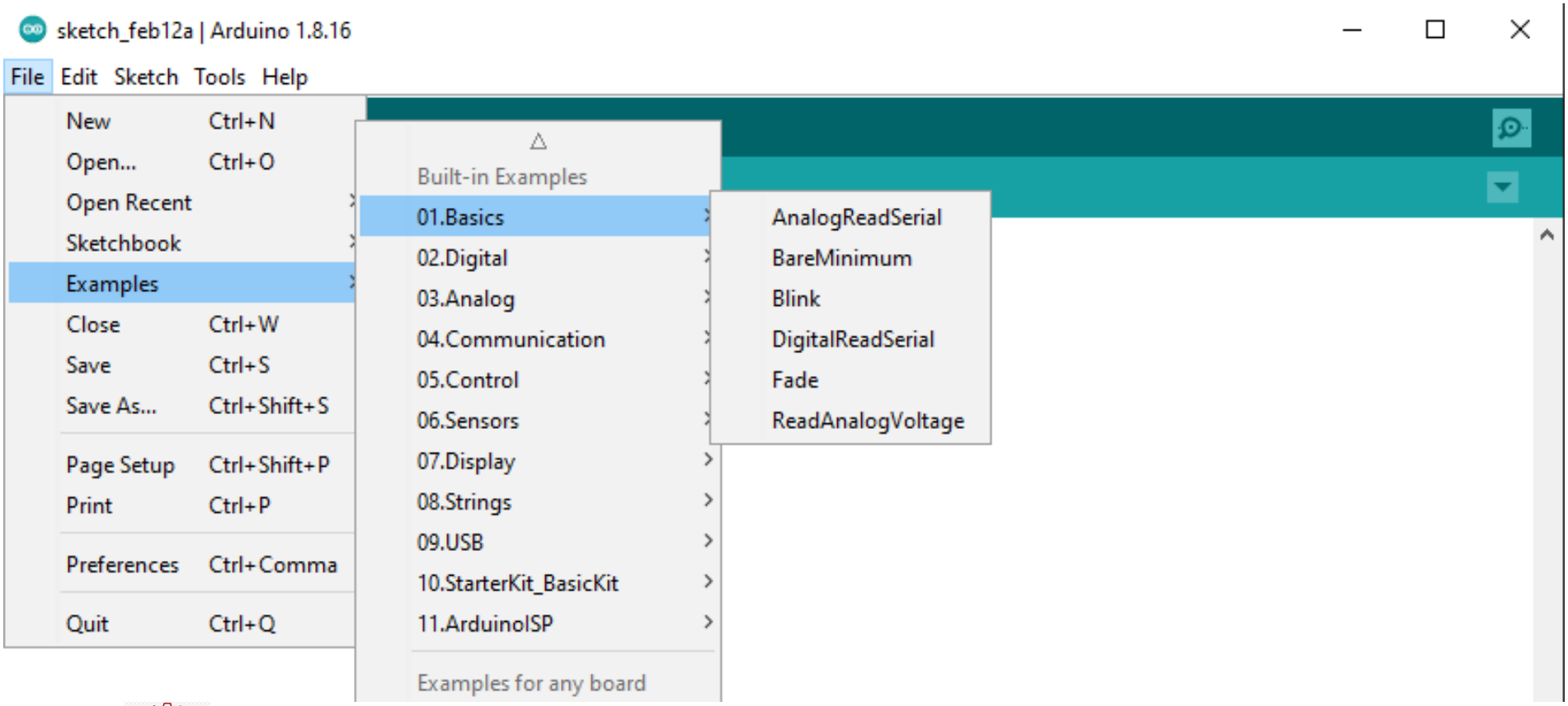


IIT BOMBAY

e-Yantra Engineering a better tomorrow

Our First Program: Blink LED

1. Open Arduino IDE
2. Click on File ⇒ Examples ⇒ 01.Basics ⇒ Blink



MPU6050

For getting data on the orientation of the body, we use **IMUs (Inertial Measurement Units)**.
They are mostly a combination of multiple sensors, like an accelerometer and a gyroscope.
What do they measure???

- **Accelerometer:** It measures the **Linear Acceleration** in a particular direction. MPU-6050 has a 3-axis Accelerometer i.e. it can measure for XYZ axis.
- **Gyroscope:** It measures the **Angular Velocity** on an axis. MPU-6050 has a 3-axis Gyroscope i.e. it can measure for XYZ axis.

Reference:
`https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/`

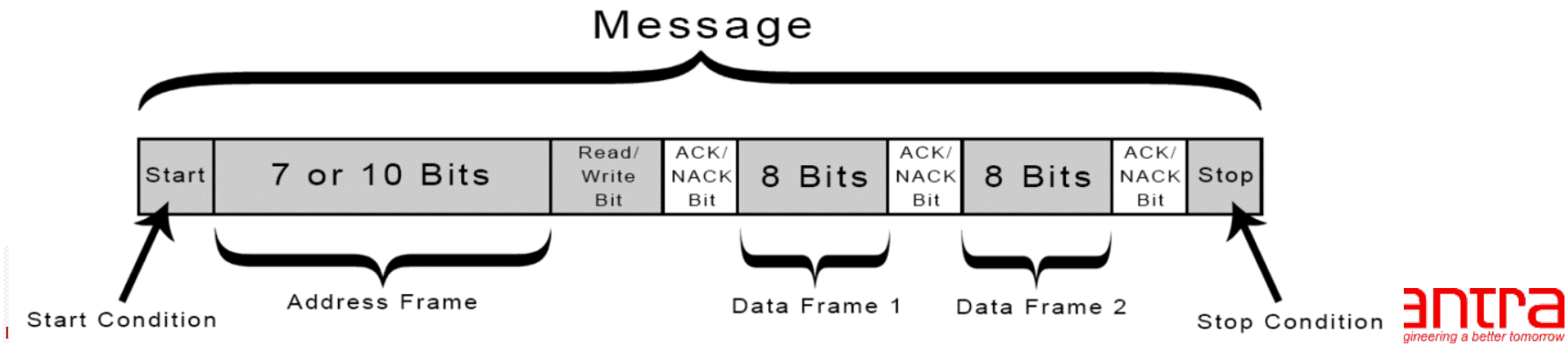
IIT BOMBAY

e-Yantra Engineering a better tomorrow

Communication

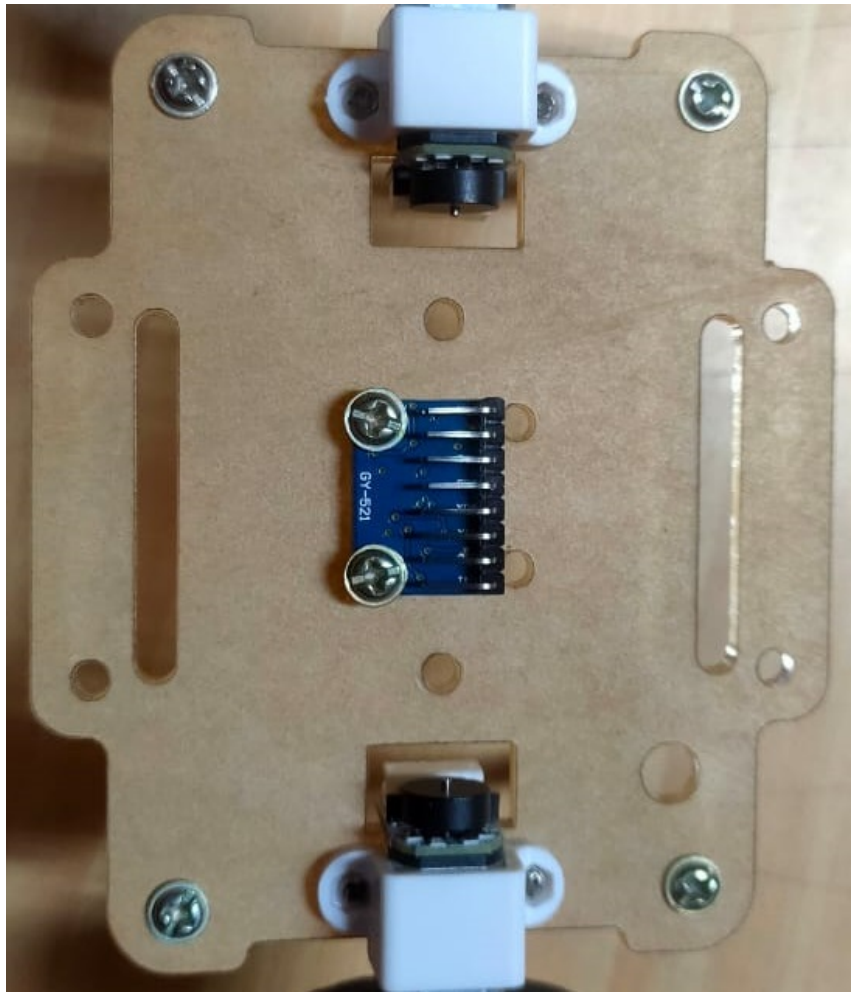
I2C Communication:

- The I2C communication is a two-wire communication method where the two wires are named SCL and SDA.
- SCL is the clock line. It is used to synchronize all data transfers over the I2C bus.
- SDA is the data line.
- In this scenario the Micro-controller is the Master Device and the IMU is the Slave Device.
- Slave will transfer the data to Master only when it is requested. Hence, this action must be done inside the Timer ISR.



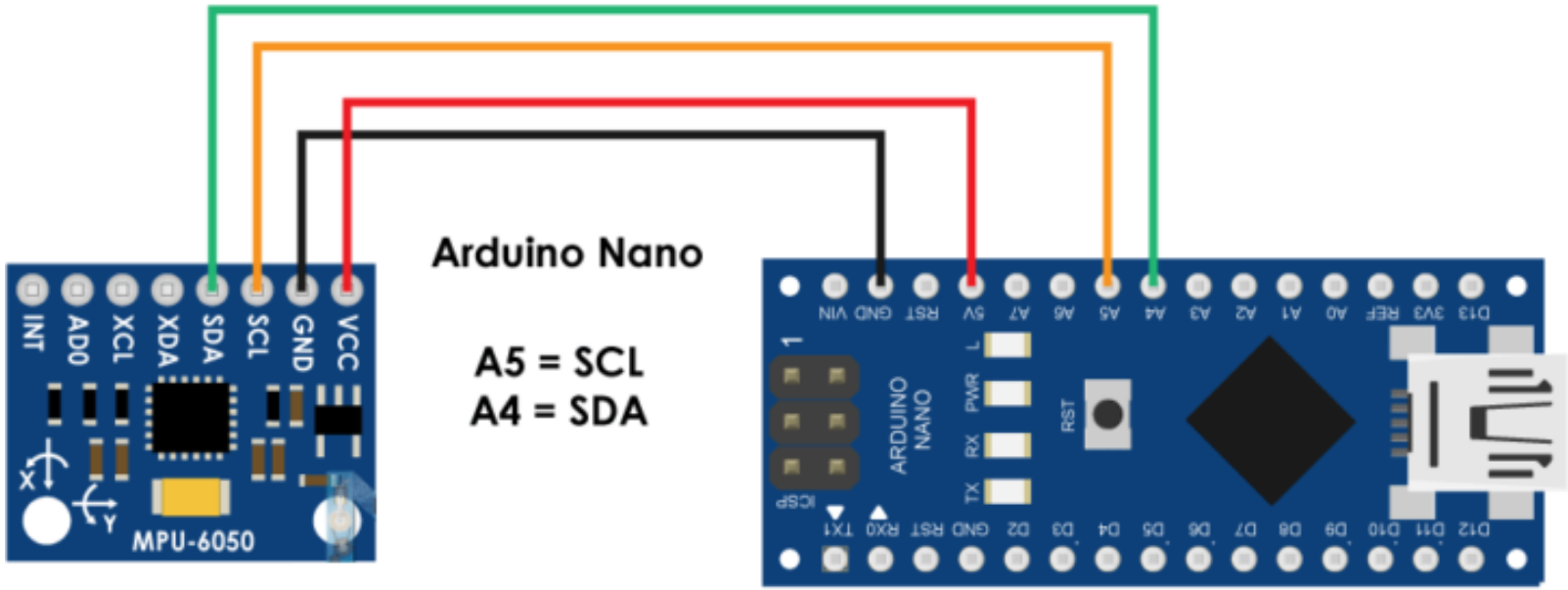
Assembly

Mount the MPU6050 to the bottom of the plate.



MPU6050 Connection

MPU6050 Pin	Arduino Pin
VCC	5V
GND	GND
SCL	A5
SDA	A4



IIT BOMBAY

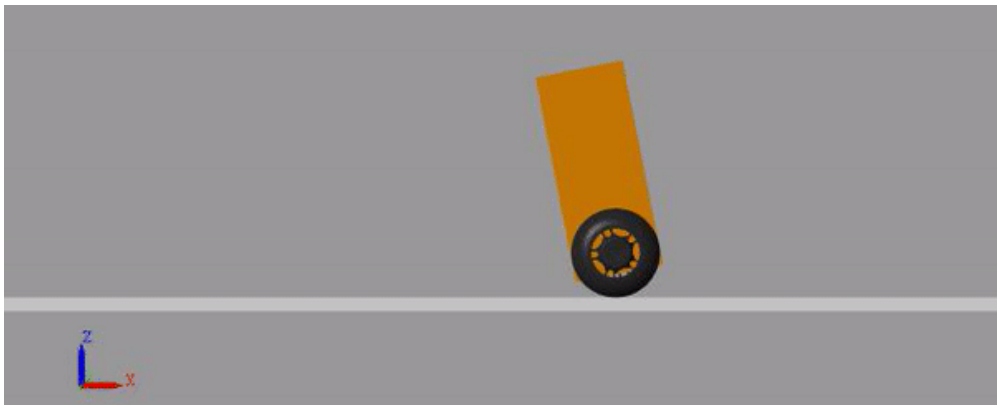
e-Yantra Engineering a better tomorrow

How to get the Data?

1. Open Arduino IDE
2. Click on File ⇒ Examples ⇒ MPU6050_light ⇒ GetAllData

N20 Motors

- DC Mini Metal Geared Motor
- High Torque and Low RPM
- **Specifications**
  - Operating Voltage - 6 - 12V
  - Torque - 0.4 Kg-cm
  - Speed - 300rpm



Reference: `https://github.com/gonzafernan/self-balancing-lqg`

Choice of Motors

Selection of motors must be done considering the Weight, Center of Mass (CM) Position, Max Retrievable Angle.
These points together will define the Torque and RPM of the Motor.
The minimum torque required for retrieving the body from an angle of θ is:

$$\tau = mglsin(\theta)$$

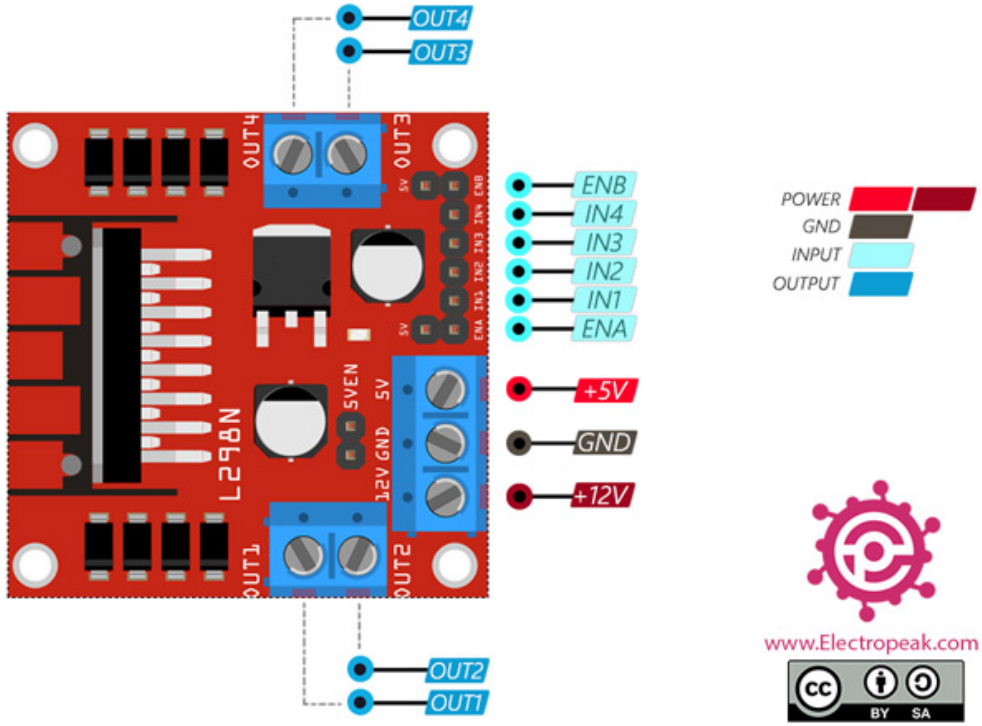
where:
m is mass of the body
l is the shortest distance of CM from the axis of rotation
g is acceleration due to gravity
θ is the angle between the vertical axis and the shortest line joining the CM and axis of rotation.

L298n Motor Driver

Why?

- Maximum current that a pin can source / sink is 20mA.
- Motors requires a current of up to 1A to attain maximum torque for loads.

L298n PinOut

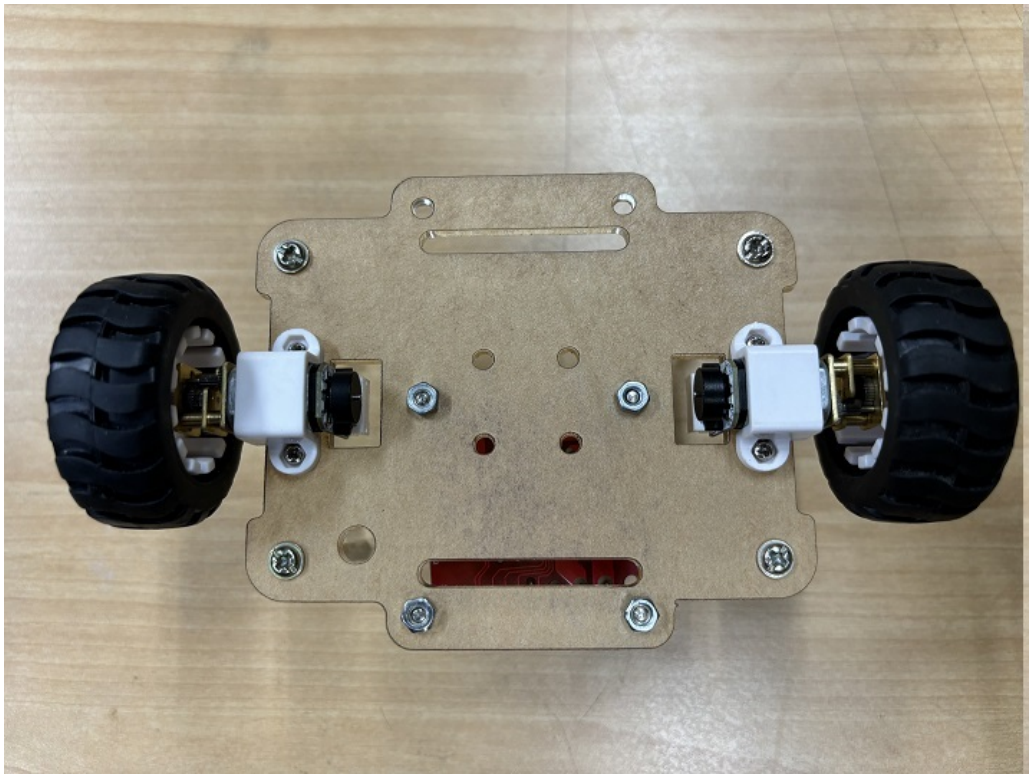


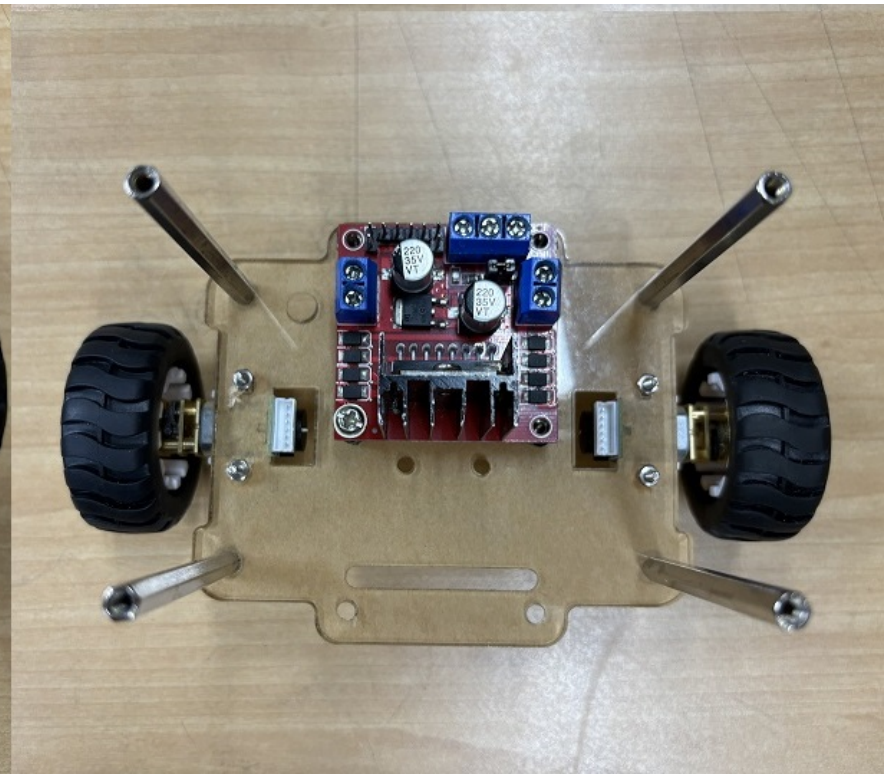
www.Electropeak.com

IIT BOMBAY

e-Yantra Engineering a better tomorrow

Assembly



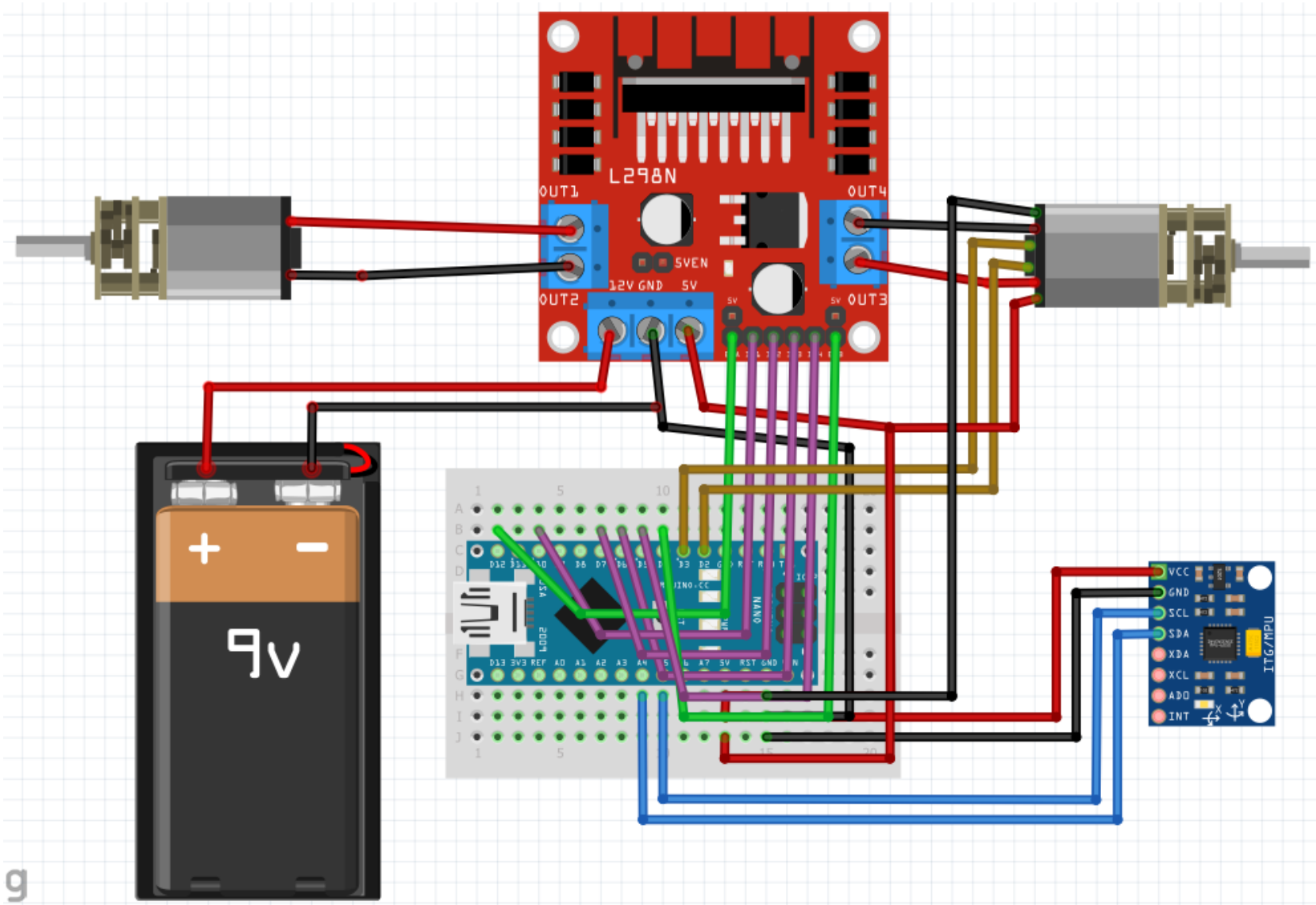


N20 Motor Connection

N20 Motor Pins	Arduino & L298n
VCC	5V
GND	GND
RM M1	L298n OUT1
RM M2	L298n OUT2
LM M1	L298n OUT1
LM M2	L298n OUT2
RM C1	Arduino D2
RM C2	Arduino D3

L298n Pins	Arduino Pin
5V	5V
GND	GND & to battery
ENA	D5
IN1	D4
IN2	D7
IN3	D10
IN4	D12
ENB	D6

Component connection Diagram



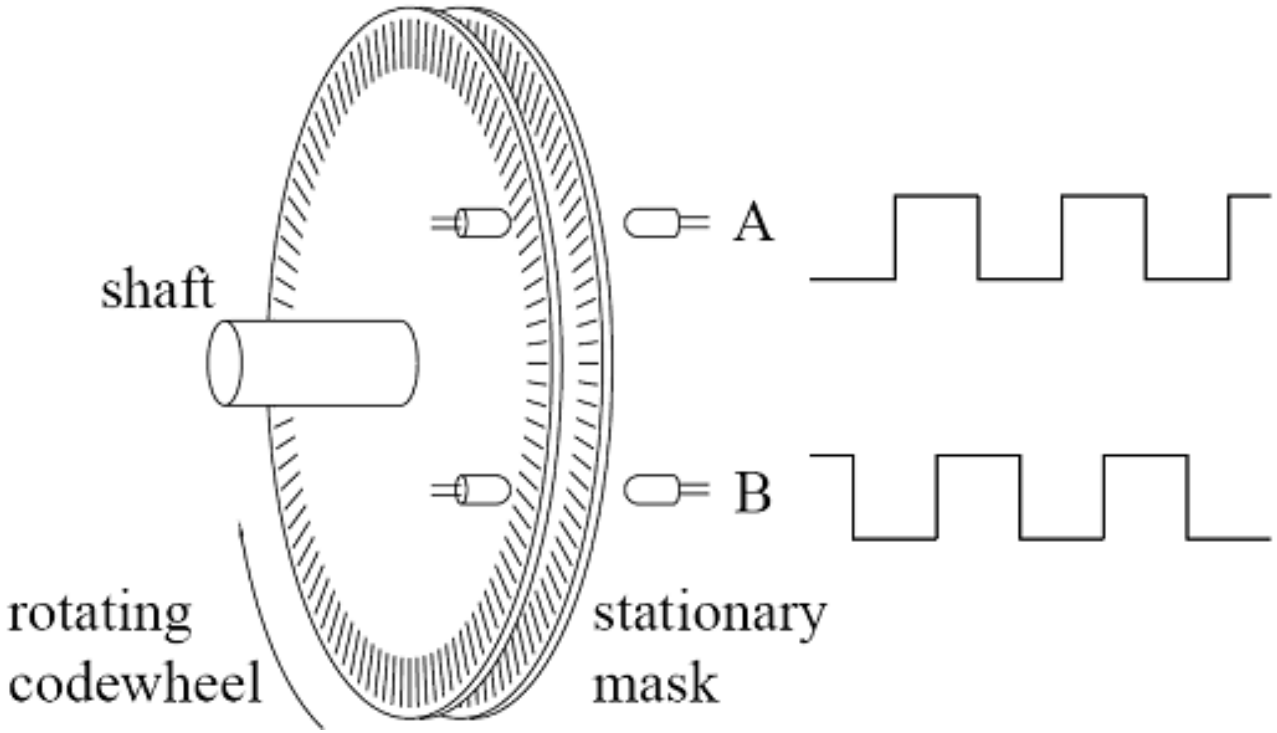
Take control of Motors

Direction	EnA	In1	In2	In3	In4	EnB
Forward	255	1	0	1	0	255
Backward	255	0	1	0	1	255
Right	255	1	0	0	1	0
Left	0	0	0	1	0	255

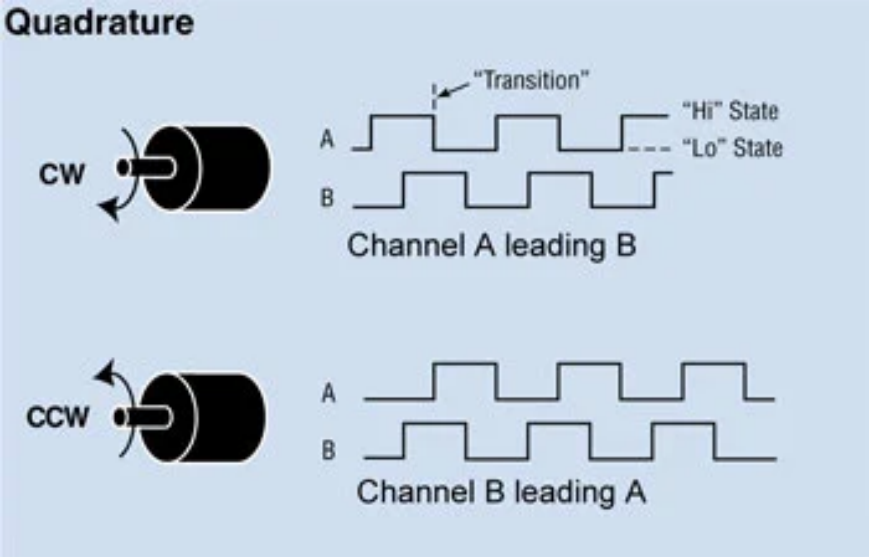
- Open Arduino IDE and Create a New Sketch.
- In the Setup function, set motor pins as output.
- In the loop function, let us do the following.
  1. Rotate both the motors forward for 1s.
  2. Rotate the right motor forward and the left motor backward for 0.5s
  3. Stop the right motor and rotate the left motor forward for 1s.
  4. Stop both the motors for 1.5s.

N20 Motor Encoders

For getting the data on position and velocity of the body we use **Quadrature Encoders** coupled with the Motors.
It has a slotted disk and two IR transceiver.



How much rotations?



- Open Arduino IDE and Create a New Sketch.
- In the setup function, set encoder pins as input.
- In the setup function, set interrupt for the encoder Pin A for rising edge.
- In the loop function, let us print the number of pulses counted.

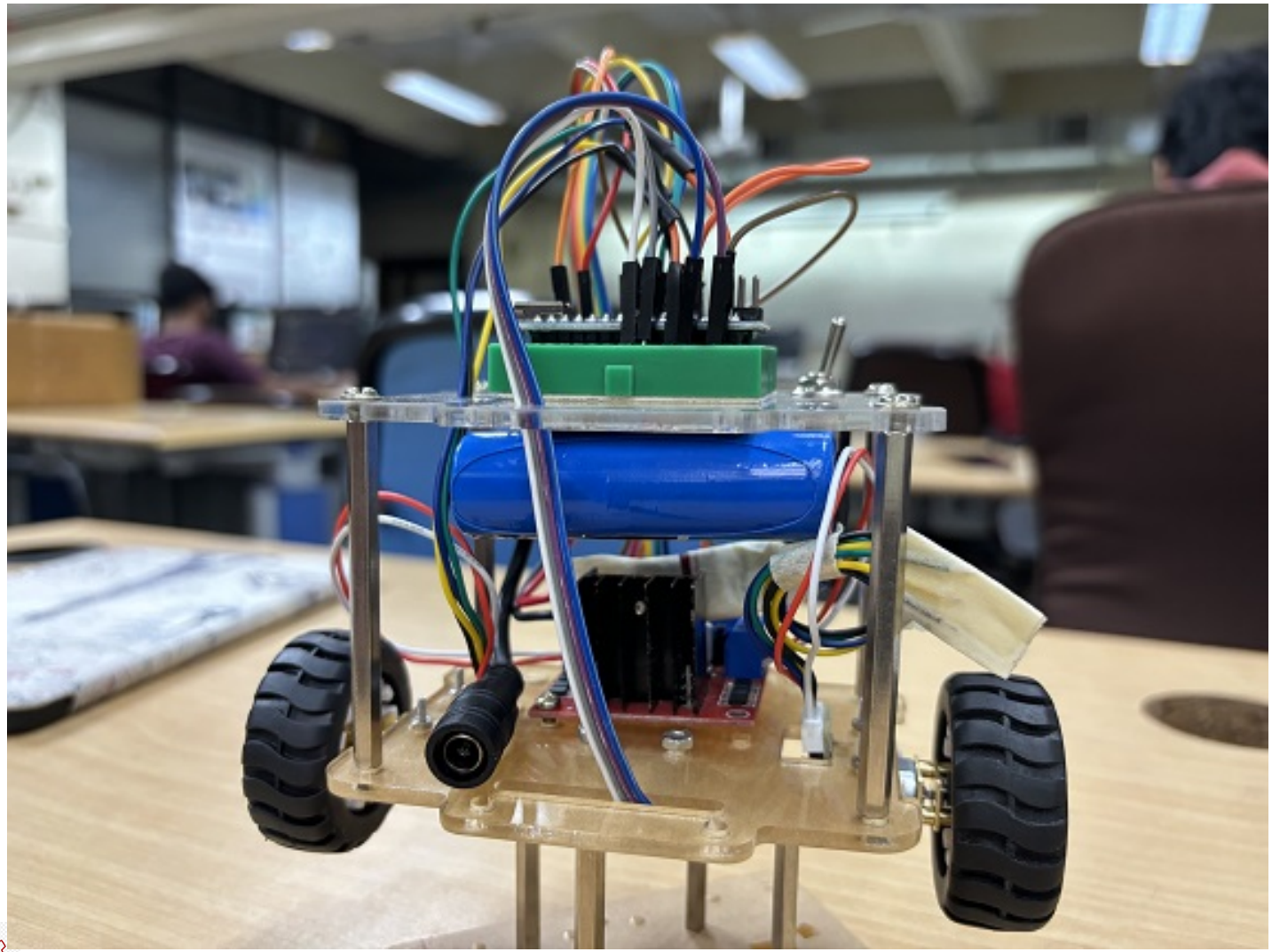
List of Programming APIs

Function	Description
`pinMode(pin, mode)`	Configures a pin as either an input or an output.
`digitalWrite(pin, value)`	Writes a digital value (HIGH or LOW) to a specified pin.
`digitalRead(pin)`	Reads the digital value (HIGH or LOW) from a specified pin.
`analogWrite(pin, value)`	Writes a PWM value to a specified pin for analog output.
`analogRead(pin)`	Reads an analog value (0-1023) from an analog pin.

List of Programming APIs

Function	Description
`Serial.begin(baudrate)`	Initializes serial communication at a specified baud rate.
`Serial.print(data)`	Prints data (numbers, characters, strings) to the serial port.
`Serial.read()`	Reads incoming serial data from the serial buffer.
`delay(ms)`	Pauses the program execution for the specified number of milliseconds.
`millis()`	Returns the number of milliseconds since the Arduino board began running the current program.

Our Bot is Ready



Thank You

Thank You!