

Self Balancing Robot : Into the Embedded Systems

Hands On Workshop

Indian Institute of Technology, Bombay

February 13, 2024



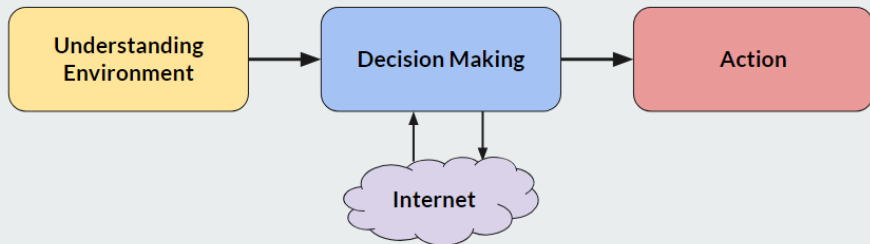
Agenda for Discussion

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

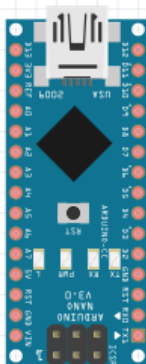


Embedded Systems

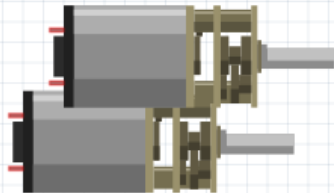
Embedded systems are compact computing systems dedicated to performing specific tasks.



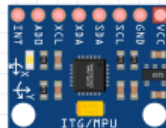
Overview of our System



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

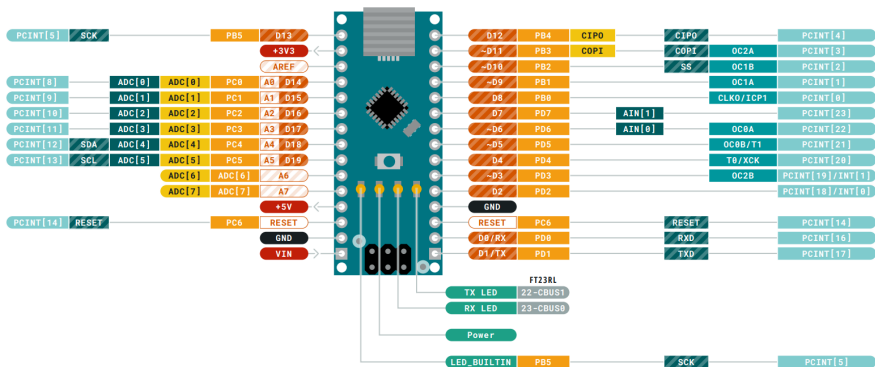


Arduino Nano

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)





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



Embedded System made easy..

Embedded 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

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

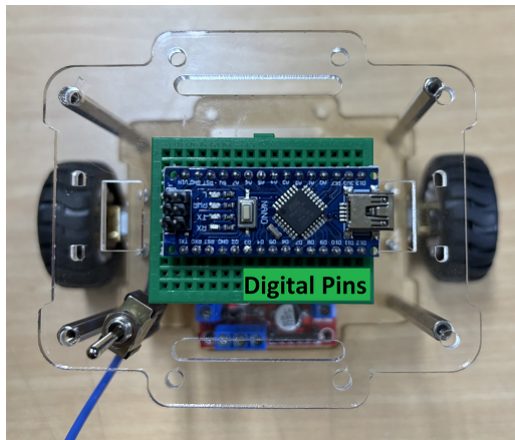
    // Set Pin 3 as input
    pinMode(3, INPUT);
}

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



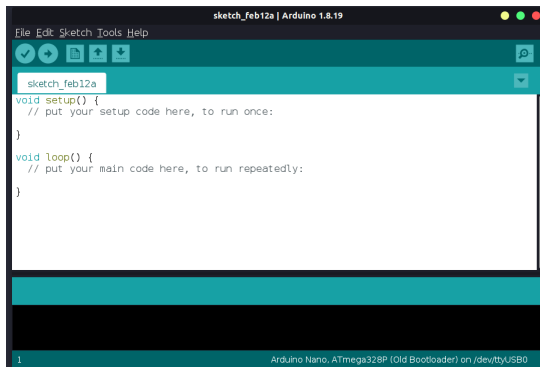
Assembly

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



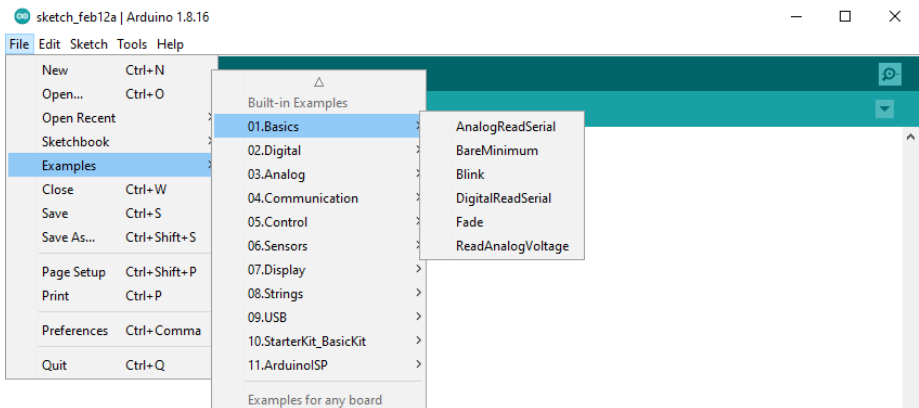
using Arduino IDE

- 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”



Our First Program: Blink LED

- 1 Open Arduino IDE
- 2 Click on File \Rightarrow Examples \Rightarrow 01.Basics \Rightarrow 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:

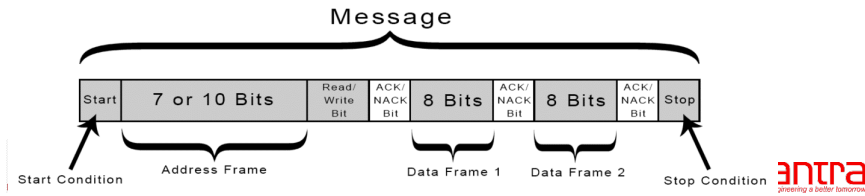
[//lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/](https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/)



Communication

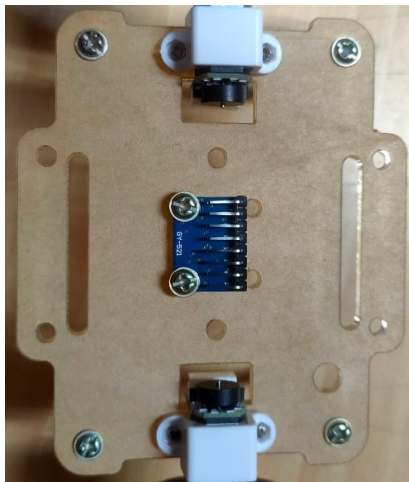
I2C Communication:

- 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.



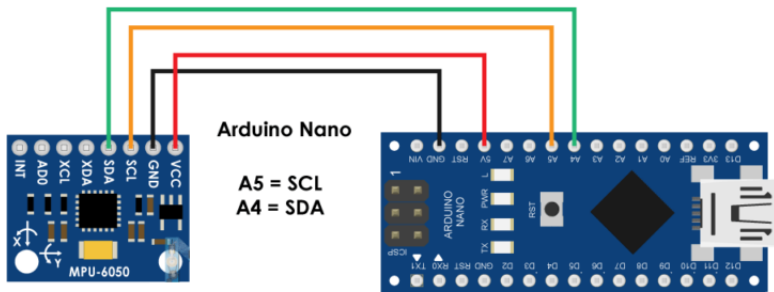
Assembly

Mount the MPU6050 to the bottom of the plate.



MPU6050 Connection

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



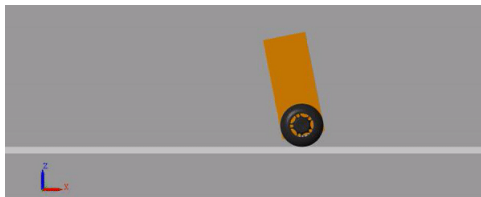
How to get the Data?

- 1 Open Arduino IDE
- 2 Click on File \Rightarrow Examples \Rightarrow MPU6050_light \Rightarrow 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/gonzafernand/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:

$$\tau = mgl\sin(\theta)$$

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.

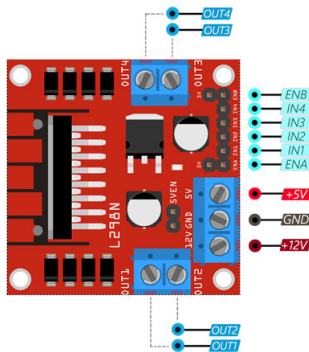


L298n Motor Driver

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.

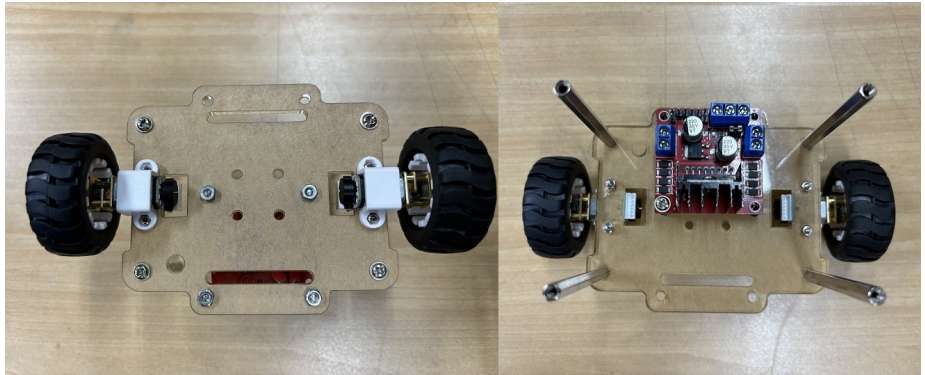
L298n PinOut



www.Electropeak.com



Assembly

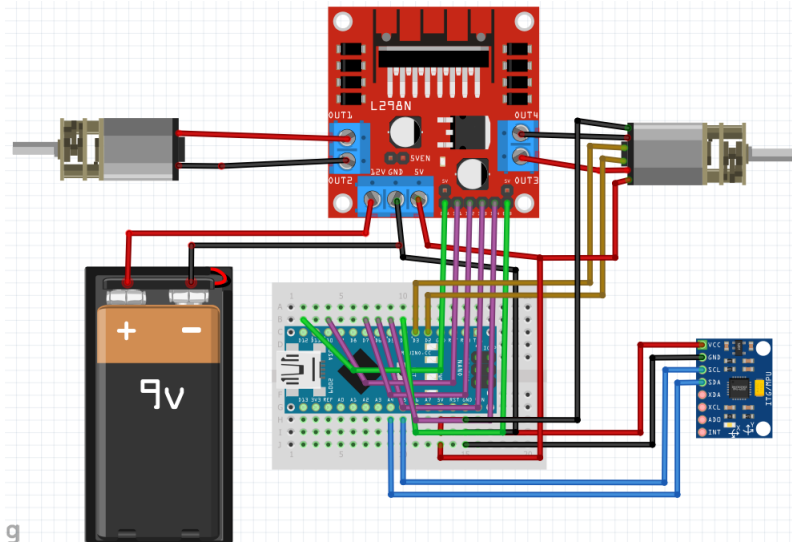


N20 Motor Connection

N20 Motor Pins	Arduino & L298n	L298n Pins	Arduino Pin
VCC	5V	5V	5V
GND	GND	GND	GND & to battery
RM1	L298C1A	ENA	D12
RM2	L298C1B	IN1	D10
LM1	L298C2A	IN2	D7
LM2	L298C2B	IN3	D6
RMC1	L298C1A	IN4	D5
RMC2	L298C1B	ENB	D4



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

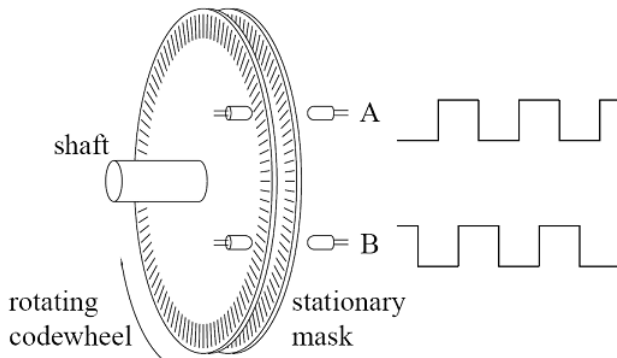
- Open Arduino IDE and Create a New Sketch.
- In the Setup function, set motor pins as output.
- In the loop function, let us do the following.
 - 1 Rotate both the motors forward for 1s.
 - 2 Rotate the right motor forward and the left motor backward for 0.5s
 - 3 Stop the right motor and rotate the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



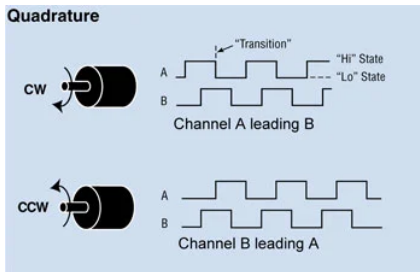
N20 Motor Encoders

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

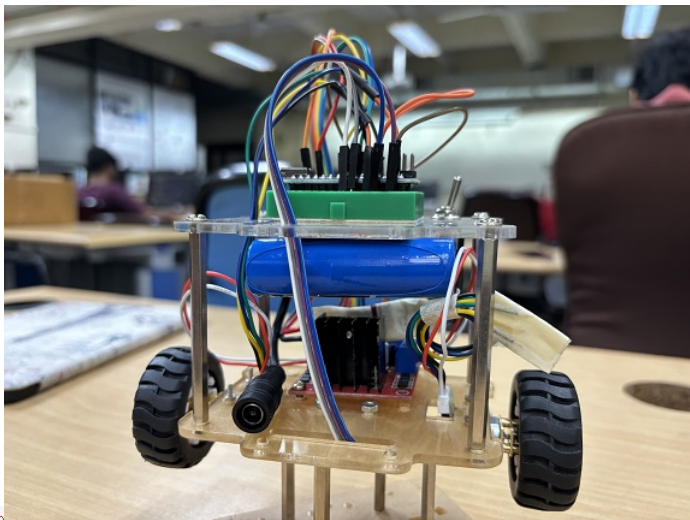


List of Programming APIs

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



Our Bot is Ready



Thank You

Thank You!

