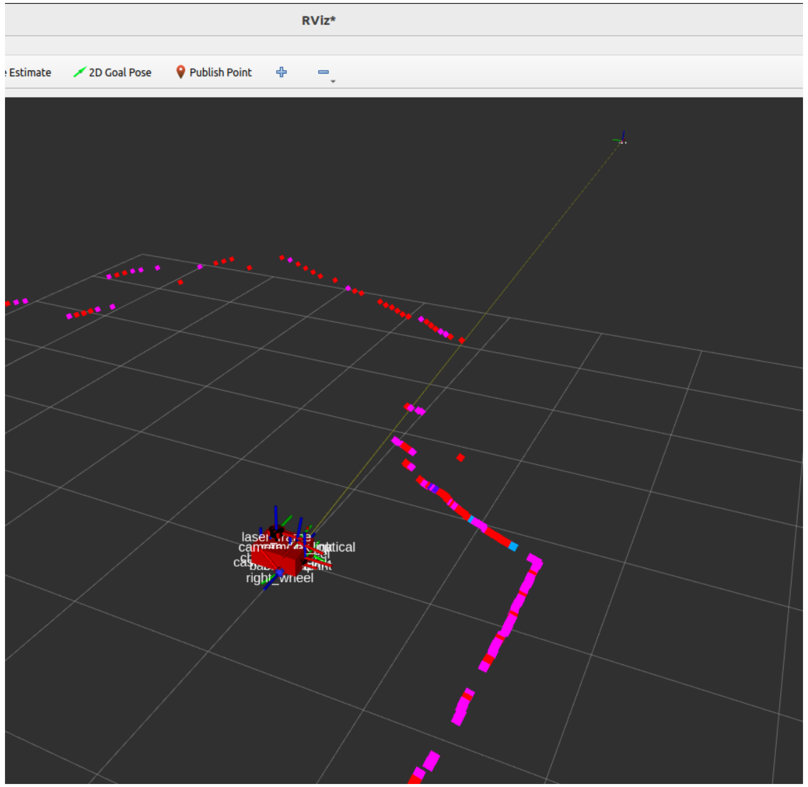
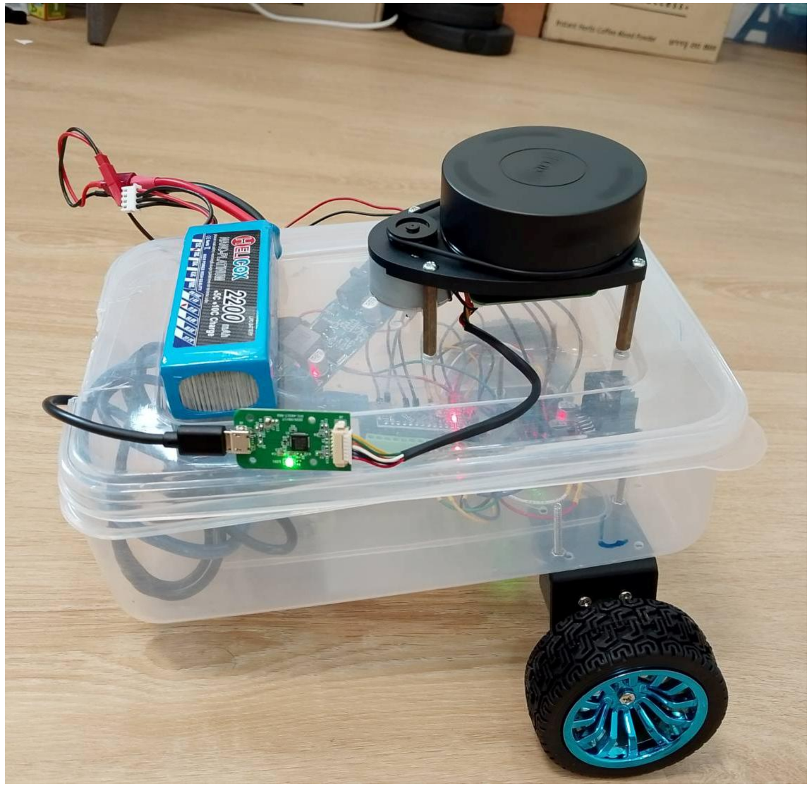




MECHATRONICS
AND AUTOMATION ENGINEERING
UBON RATCHATHANI UNIVERSITY

Lab Work: Basic Robot

Week1-2: Motor Encoder for position and speed

Mechatronics Engineering UBU

Content

1. Motor encoder

   1.1 Motor encoder by Magnet sensor

   1.2 Use of External interrupt for faster reading encoder

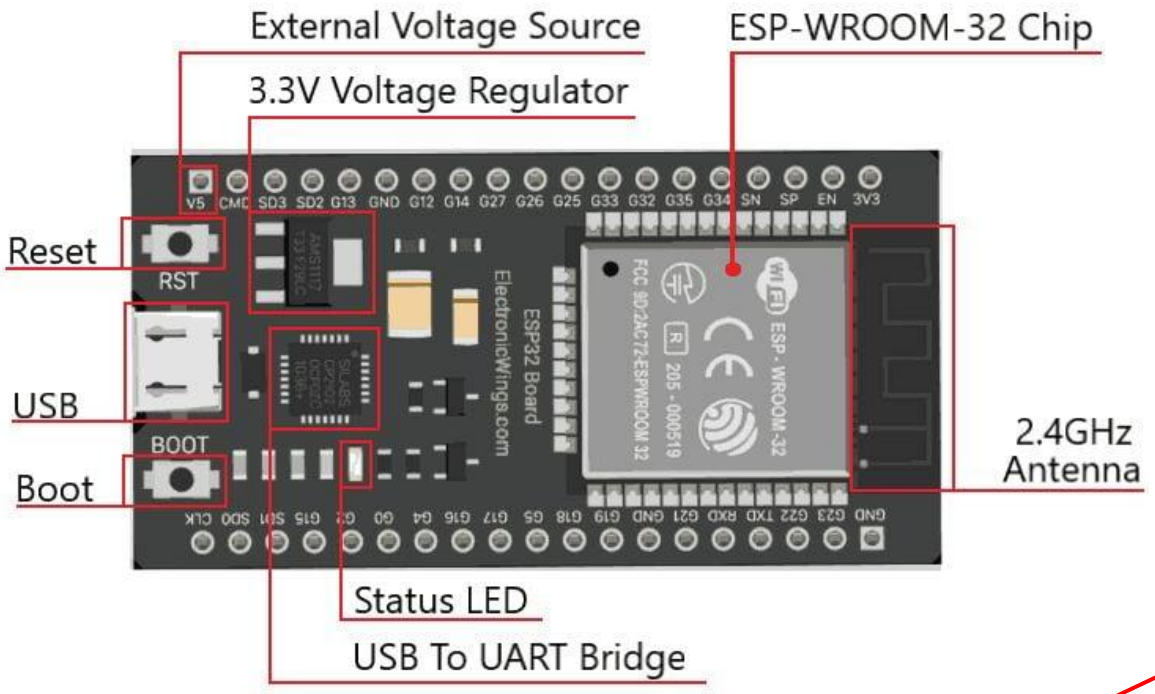
2. Control Movement and Position Motor

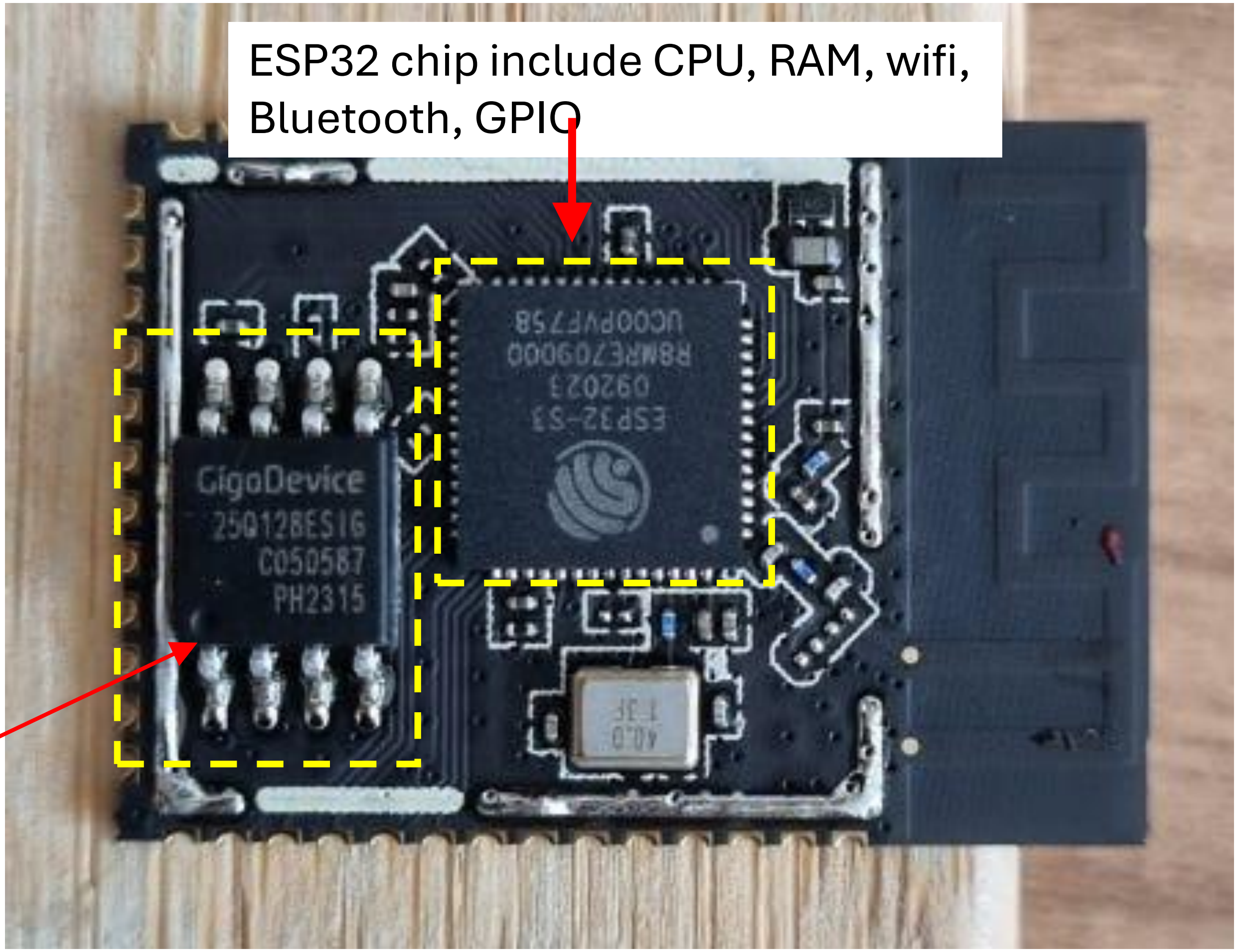
3. Control Speed Motor by Pulse Width Modulation (PWM)

4. Measure speed motor by IR sensors and Oscilloscope

5. Measure speed motor by Encoder Motor

Basic ESP32 structure



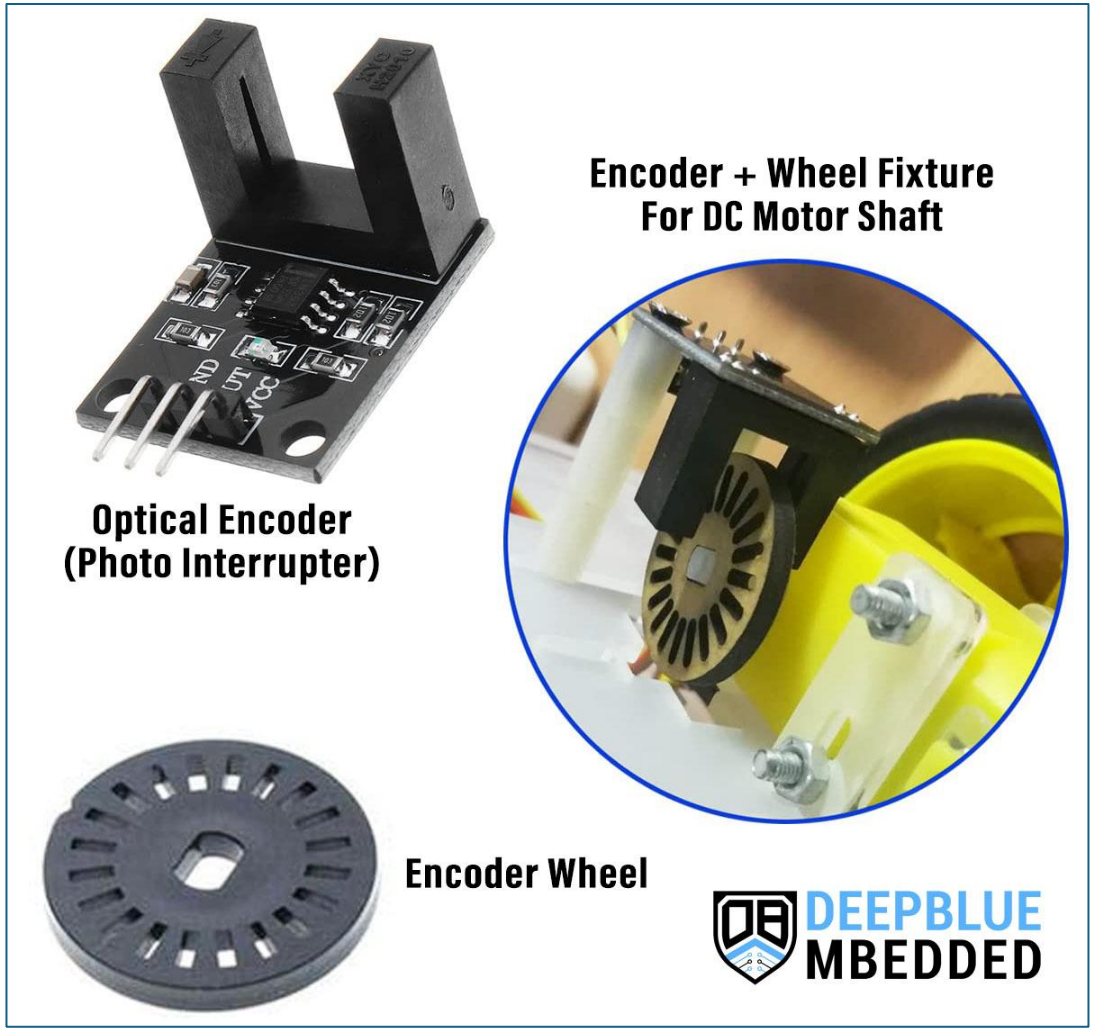

ESP32 chip include CPU, RAM, wifi, Bluetooth, GPIO

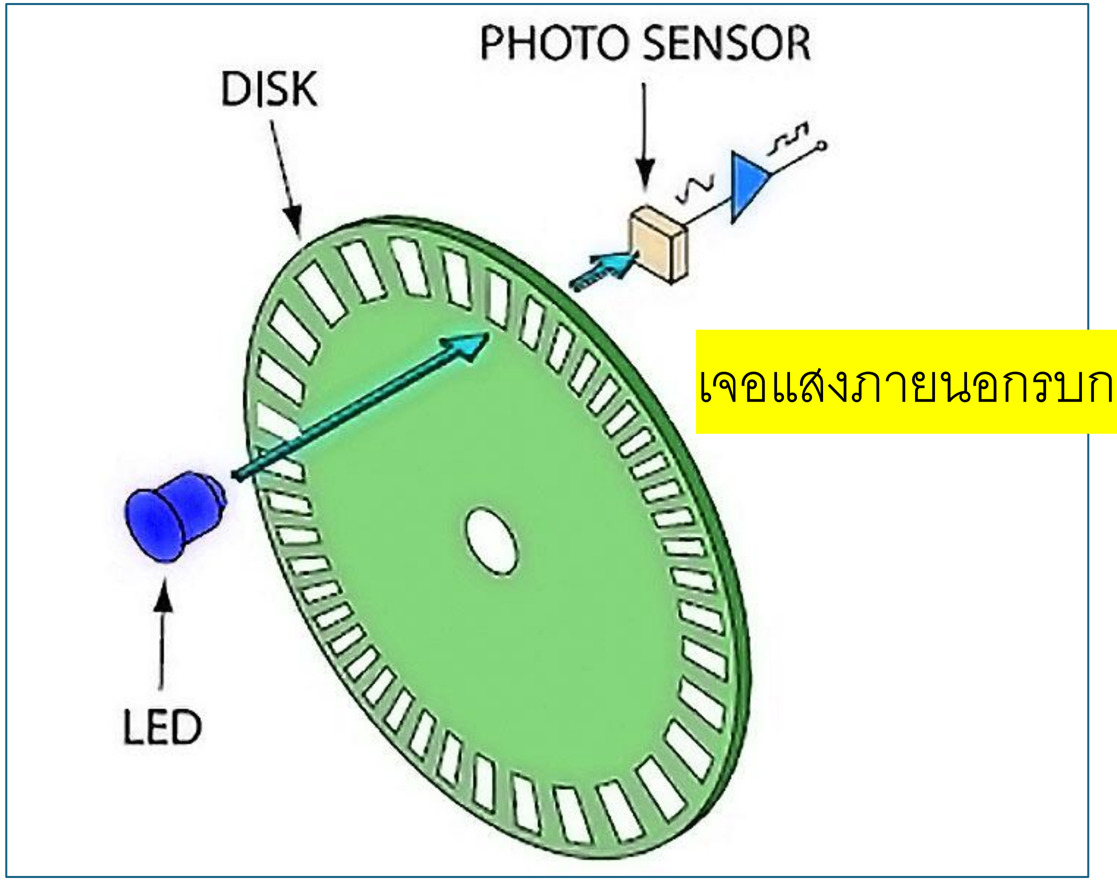
The 4 Mb GigaDevice GD25Q32 flash memory is non-volatile:
it save or delete program code, parameters, and configuration data.

1. Motor encoder

Why we use an encoder along with DC motor?

- The speed and position of a DC motor shaft can be controlled using the encoder readings. This is critical in many applications.



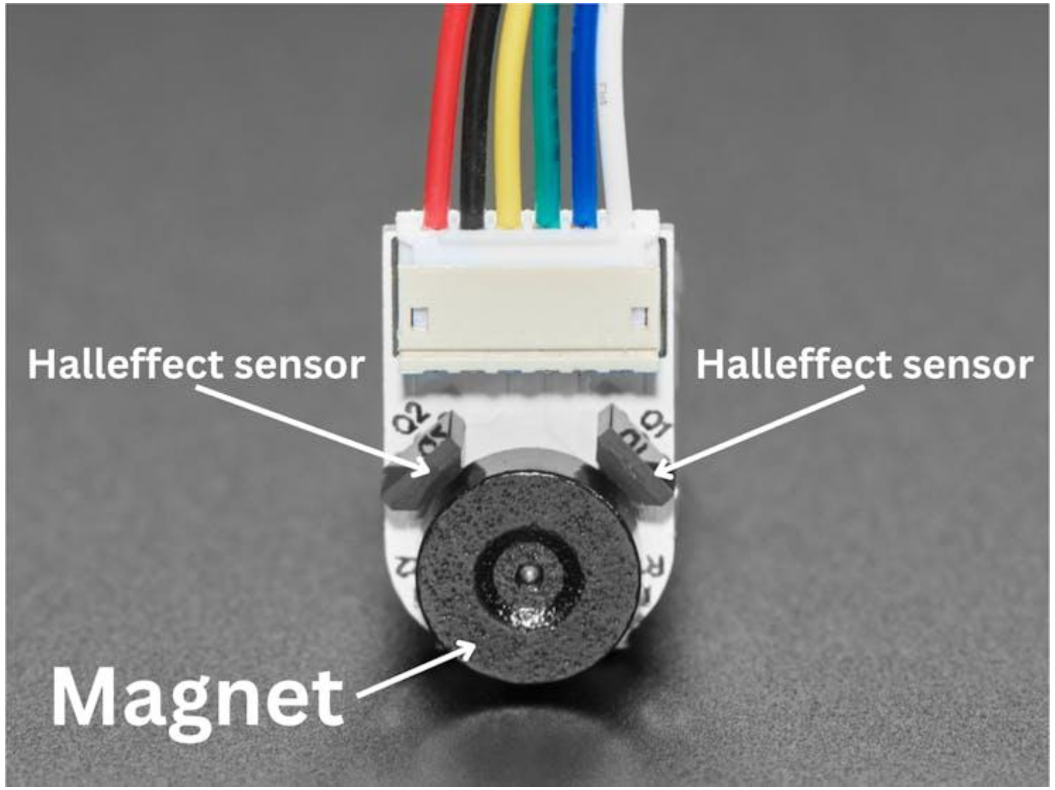


เจอแสงภายนอกรบกวน!!!

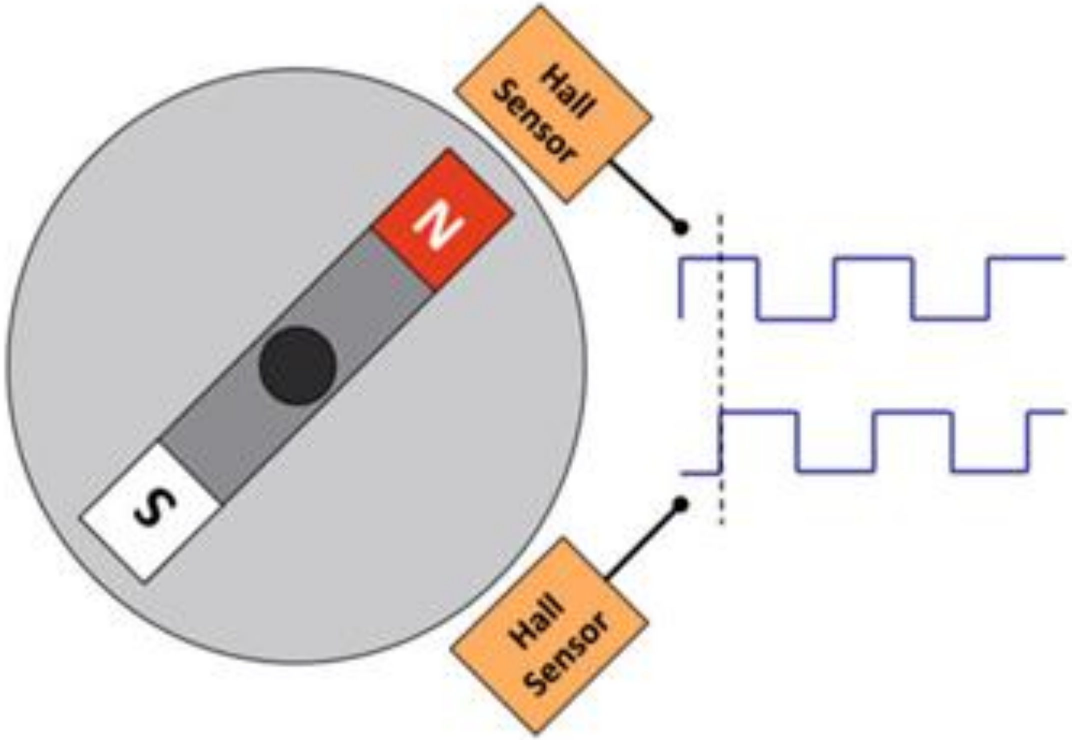
1.1 Motor encoder by Magnet sensor

Why we use an encoder along with DC motor?

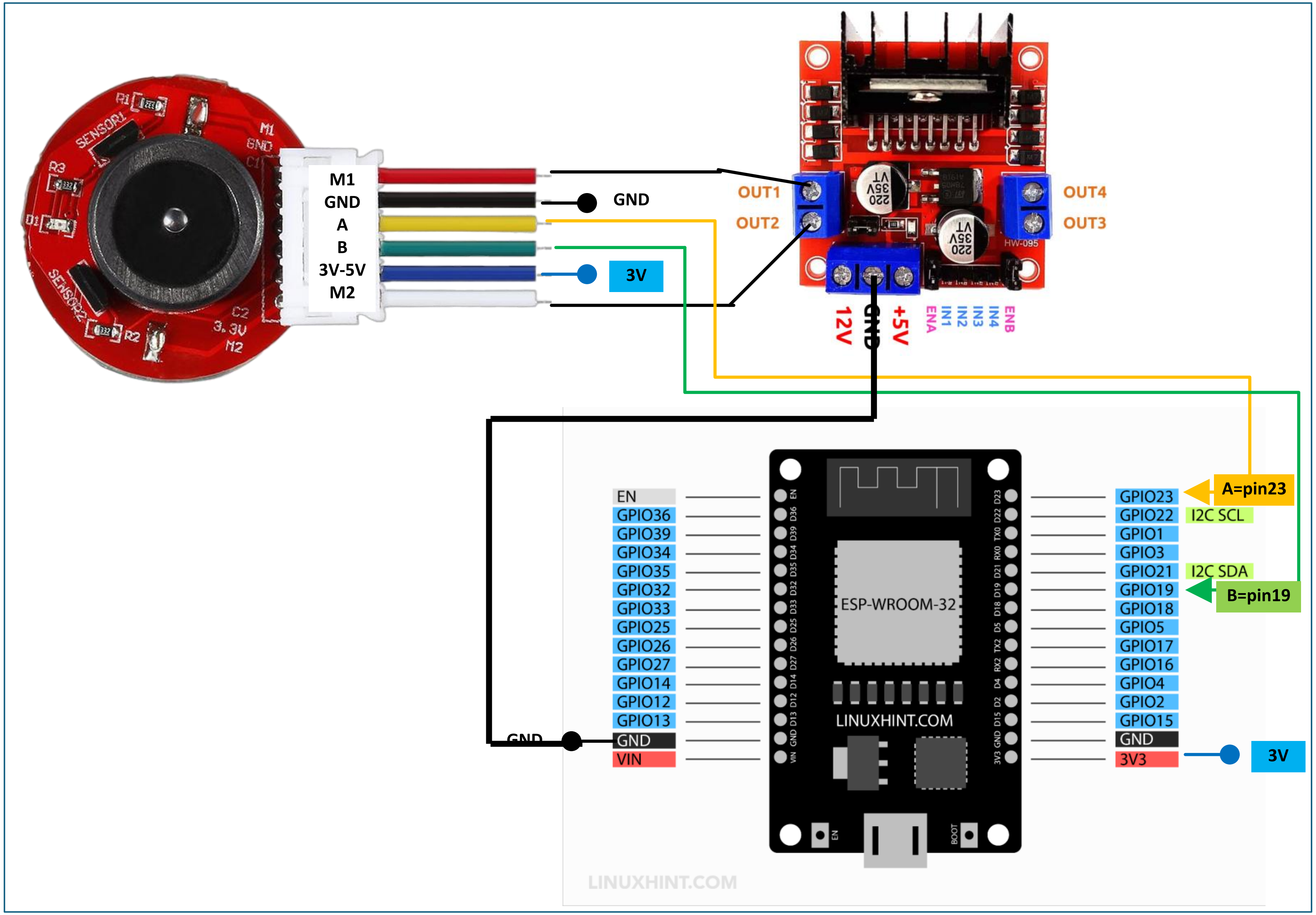
- The speed and position of a DC motor shaft can be controlled using the encoder readings. This is critical in many applications.



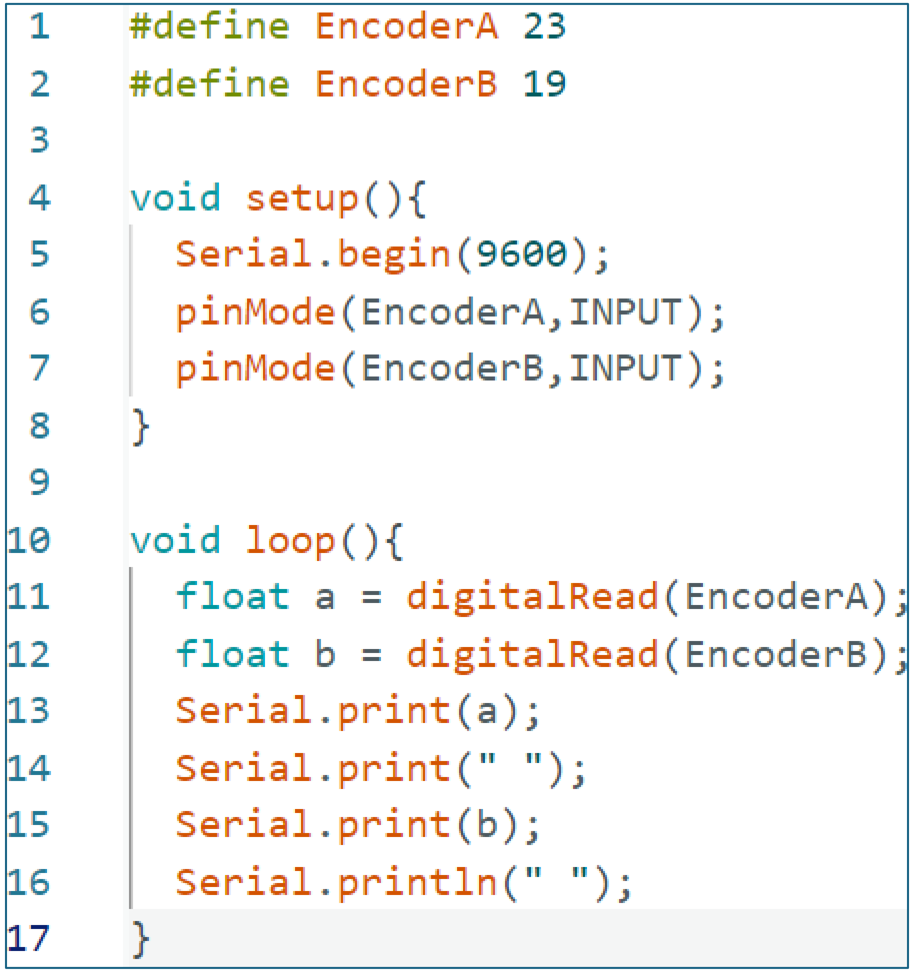
Why are there two sensors?



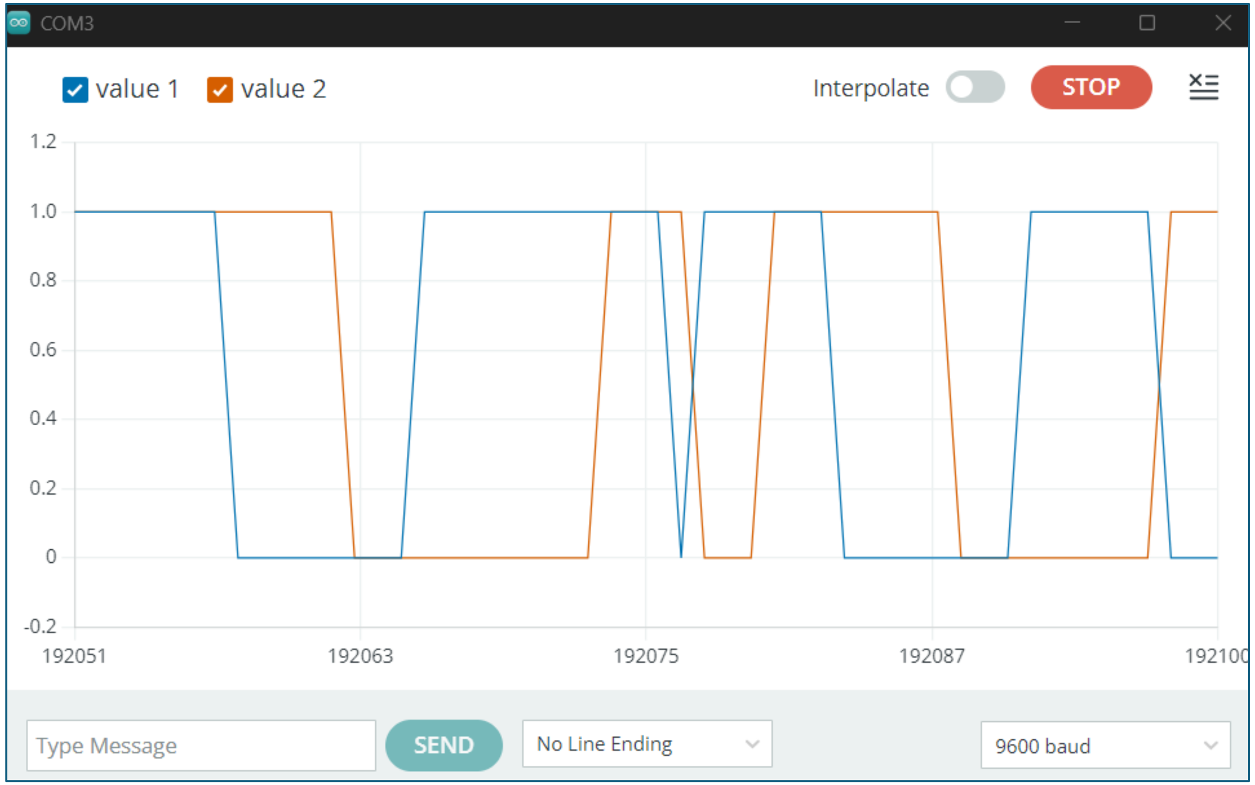
Connecting1.1 Read rotation count of motor from encoder



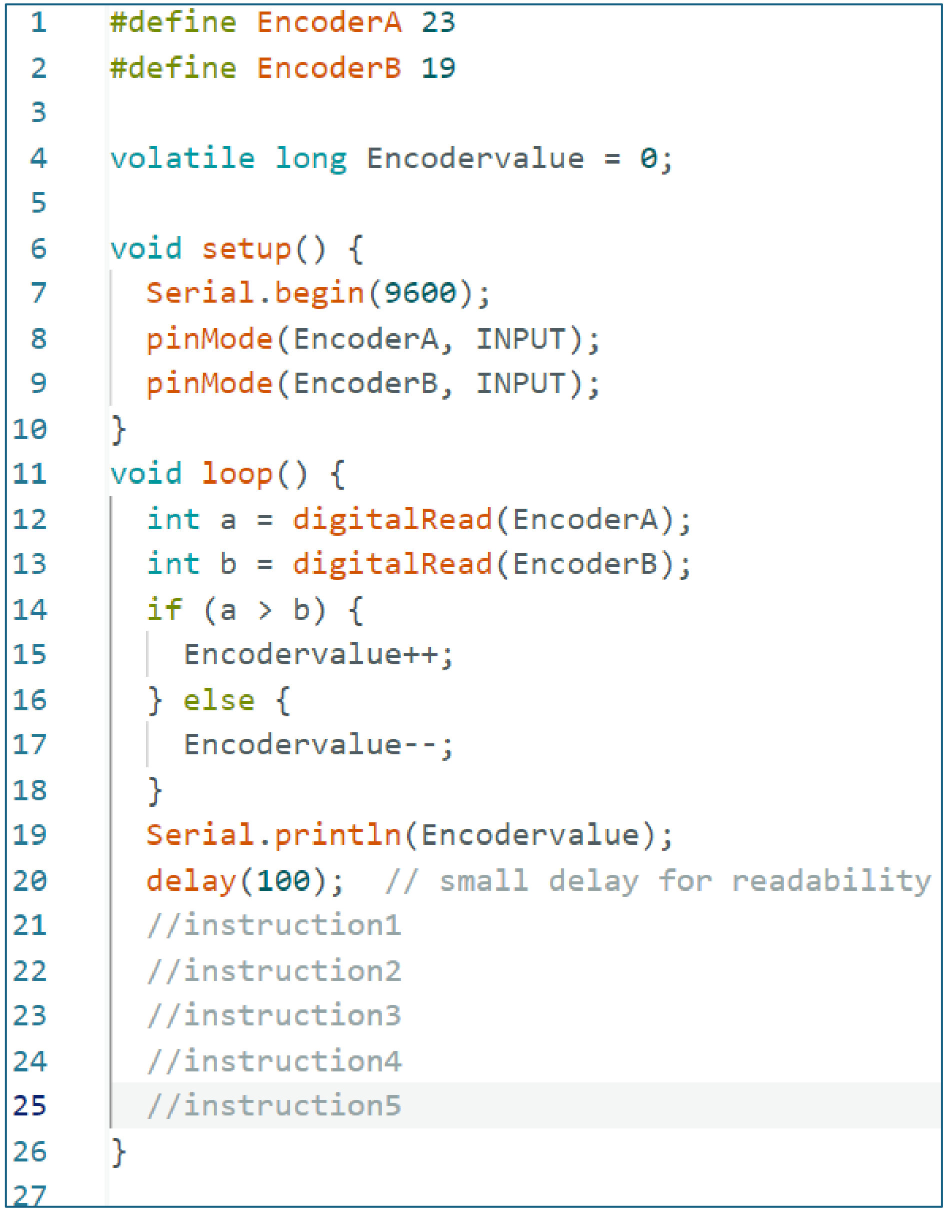
Example1.1 วิธีการนับรอบ encoder เบื้องต้น โดยดูขอบขาขึ้น A จากกราฟ



```
#define EncoderA 23
#define EncoderB 19

void setup(){
  Serial.begin(9600);
  pinMode(EncoderA,INPUT);
  pinMode(EncoderB,INPUT);
}

void loop(){
  float a = digitalRead(EncoderA);
  float b = digitalRead(EncoderB);
  Serial.print(a);
  Serial.print(" ");
  Serial.print(b);
  Serial.println(" ");
}
```



Example1.2 การเขียน code นับรอบ encoder วิธีพื้นฐาน !!!



```
#define EncoderA 23
#define EncoderB 19

volatile long Encodervalue = 0;

void setup() {
  Serial.begin(9600);
  pinMode(EncoderA, INPUT);
  pinMode(EncoderB, INPUT);
}
void loop() {
  int a = digitalRead(EncoderA);
  int b = digitalRead(EncoderB);
  if (a > b) {
    Encodervalue++;
  } else {
    Encodervalue--;
  }
  Serial.println(Encodervalue);
  delay(100);  // small delay for readability
  //instruction1
  //instruction2
  //instruction3
  //instruction4
  //instruction5
}

```

!!! ข้อเสียคือถ้าโค้ดมีหลาย instruction จะทำให้การอ่าน encoder ช้ามากๆเพราะต้องรอ loop

Content

1. Motor encoder

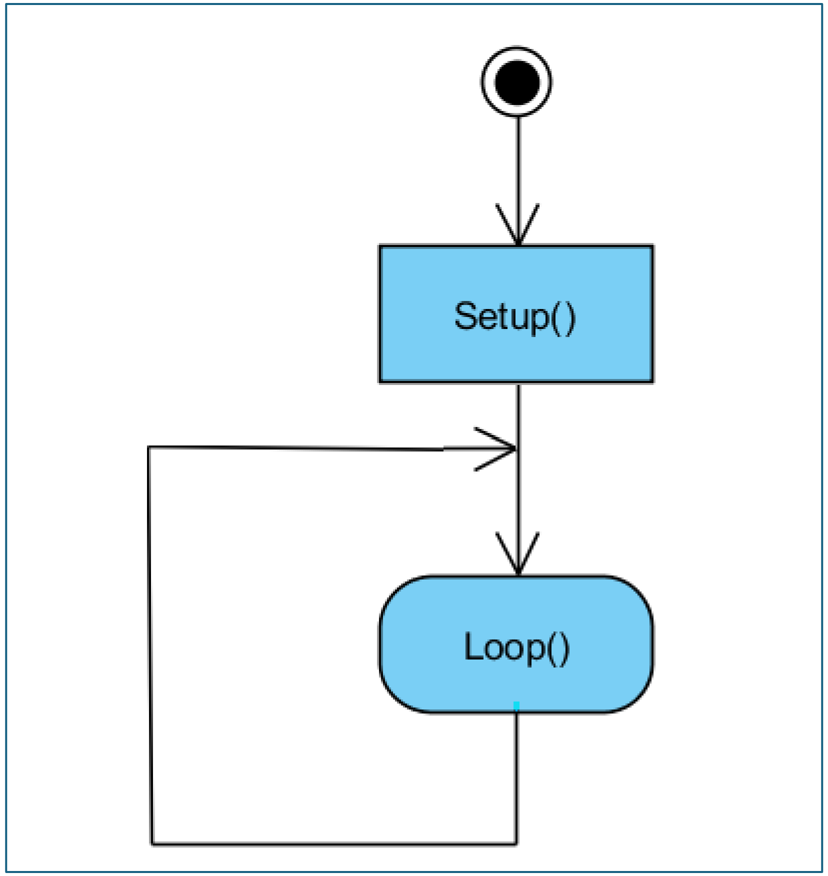
    1.1 Motor encoder by Magnet sensor

    1.2 Use of External interrupt for faster reading encoder

2. Control Movement and Position Motor
3. Control Speed Motor by Pulse Width Modulation (PWM)
4. Measure speed motor by IR sensors and Oscilloscope
5. Measure speed motor by Encoder Motor

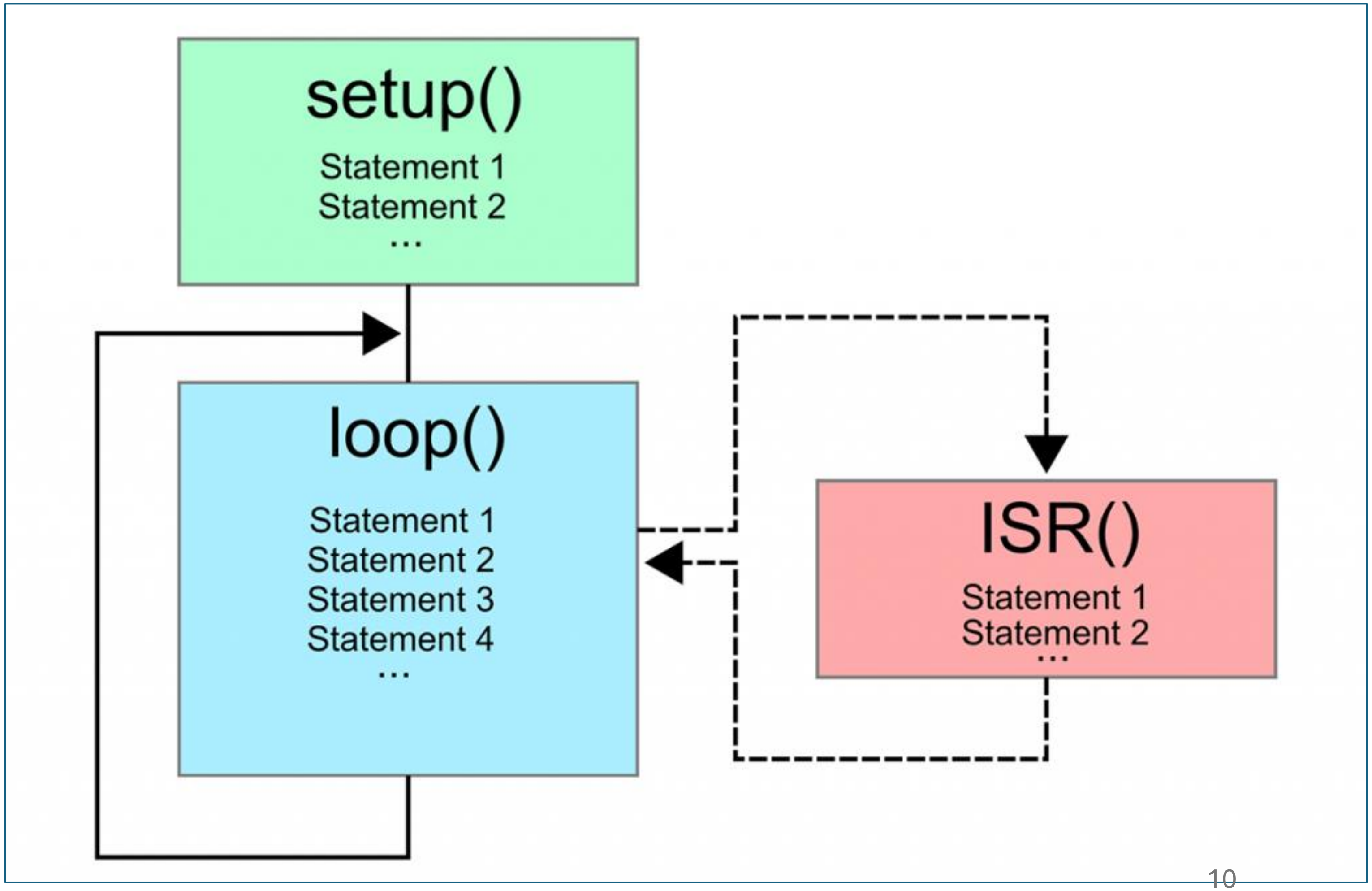
1.2 Use of External interrupt for faster reading encoder

General

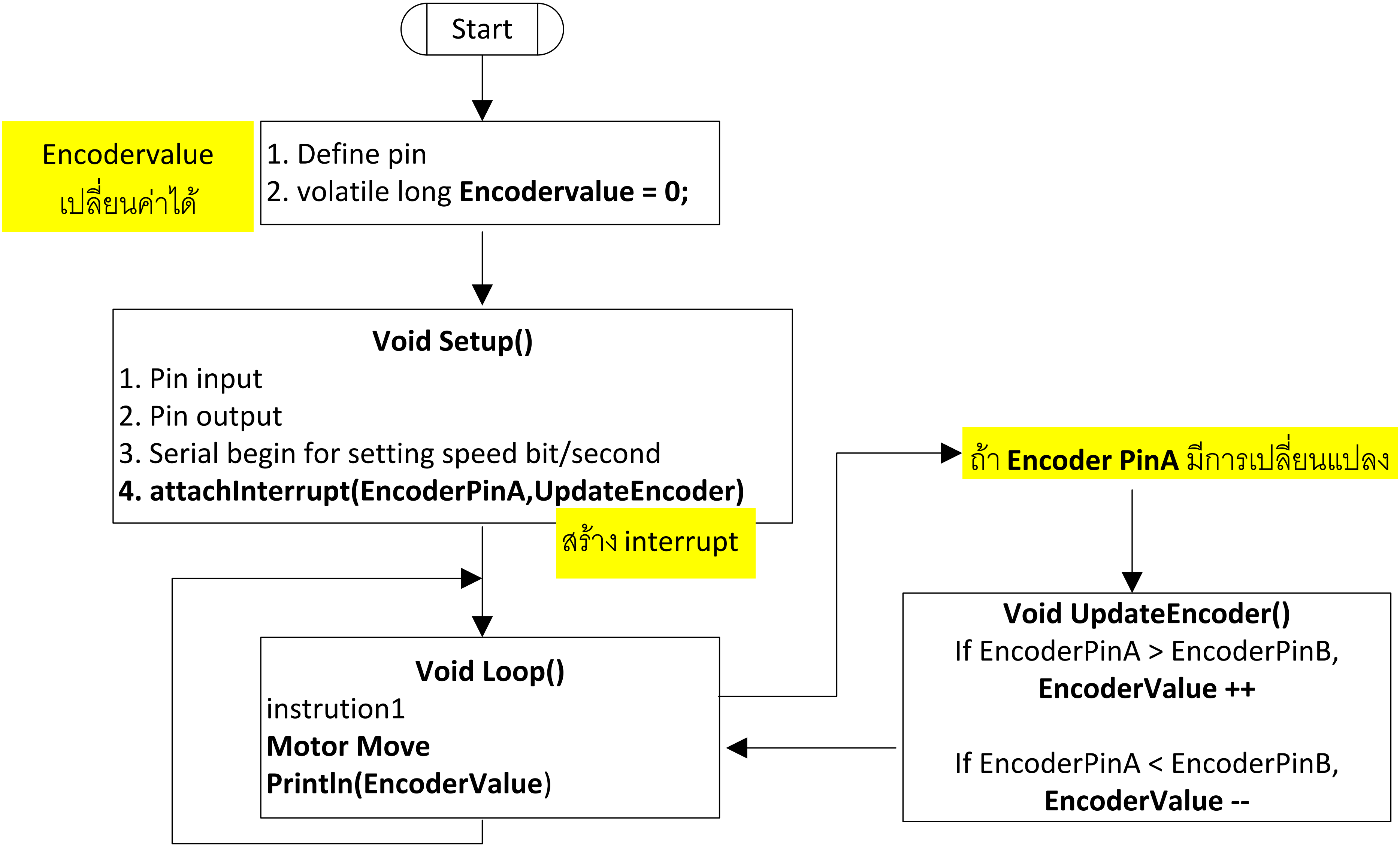


ช้ากว่า เพราะต้องวน **loop** หาค่า **signal**

General with interrupt service routine (ISR)



Flowchart1.3 for Read Encoder by interrupt



Start

Encodervalue
เปลี่ยนค่าได้

1. Define pin
2. volatile long **Encodervalue = 0;**

Void Setup()
1. Pin input
2. Pin output
3. Serial begin for setting speed bit/second
4. attachInterrupt(EncoderPinA,UpdateEncoder)

สร้าง interrupt

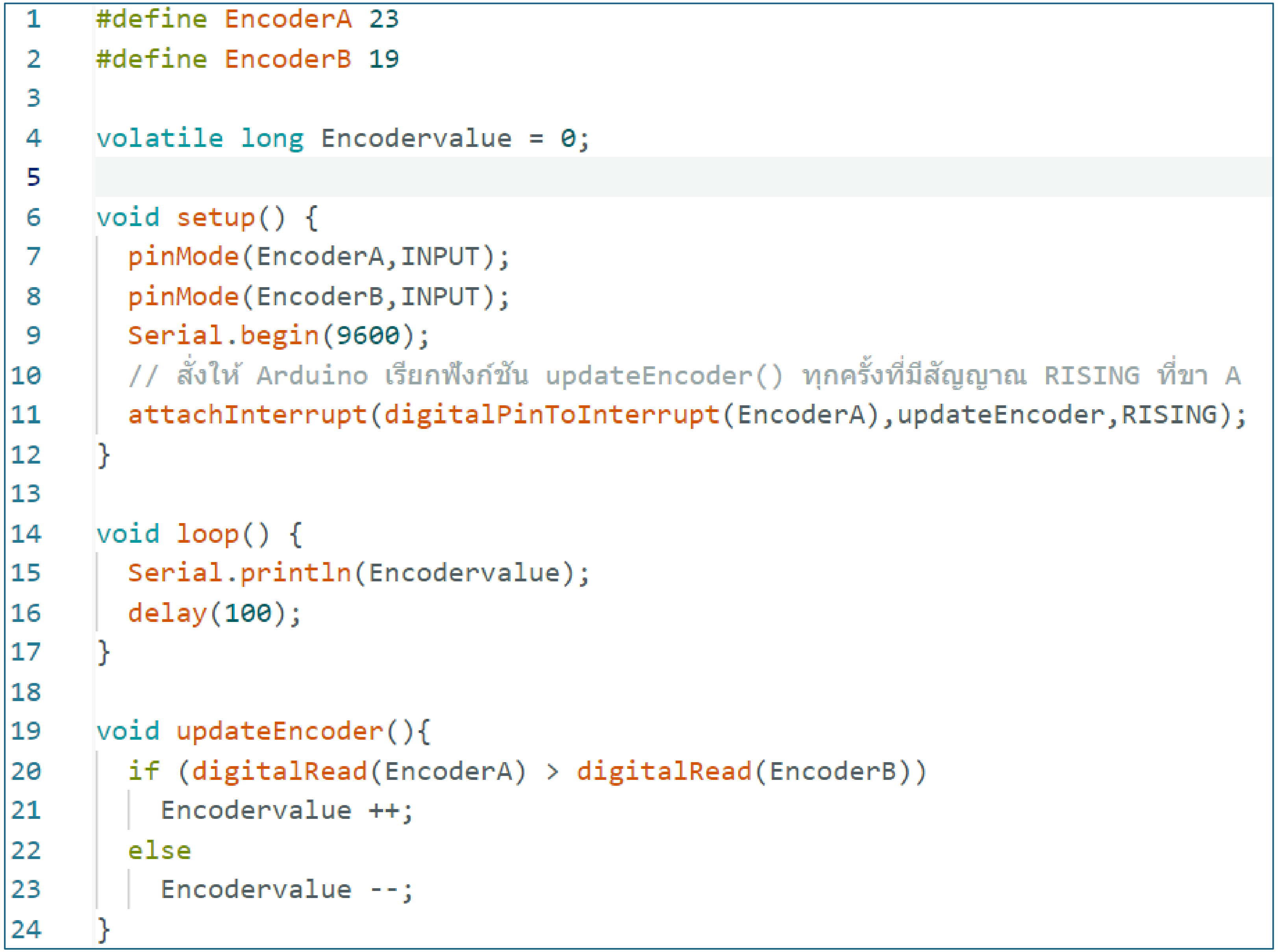
Void Loop()
instrution1
Motor Move
Println(EncoderValue)

ถ้า **Encoder PinA** มีการเปลี่ยนแปลง

Void UpdateEncoder()
If EncoderPinA > EncoderPinB,
EncoderValue ++

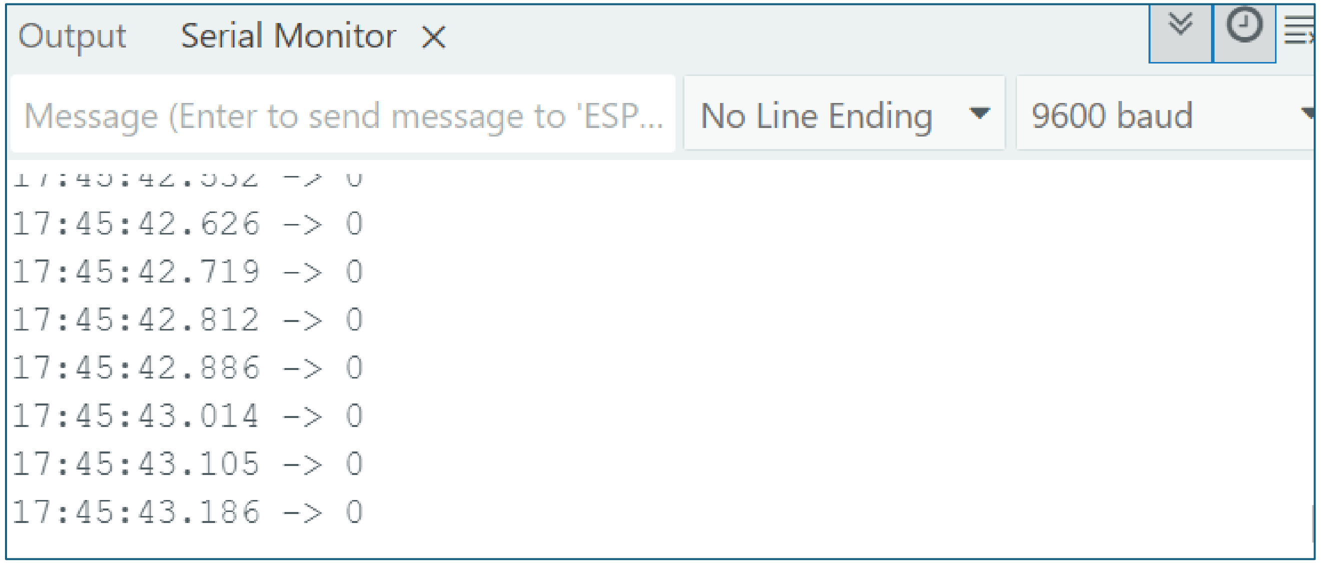
If EncoderPinA < EncoderPinB,
EncoderValue --

Example1.3 Encoder with interrupt service routine (ISR)



```
#define EncoderA 23
#define EncoderB 19

volatile long Encodervalue = 0;

void setup() {
  pinMode(EncoderA,INPUT);
  pinMode(EncoderB,INPUT);
  Serial.begin(9600);
  // สั่งให้ Arduino เรียกฟังก์ชัน updateEncoder() ทุกครั้งที่มีสัญญาณ RISING ที่ขา A
  attachInterrupt(digitalPinToInterrupt(EncoderA),updateEncoder,RISING);
}

void loop() {
  Serial.println(Encodervalue);
  delay(100);
}

void updateEncoder(){
  if (digitalRead(EncoderA) > digitalRead(EncoderB))
    Encodervalue ++;
  else
    Encodervalue --;
}
```

Begin run code



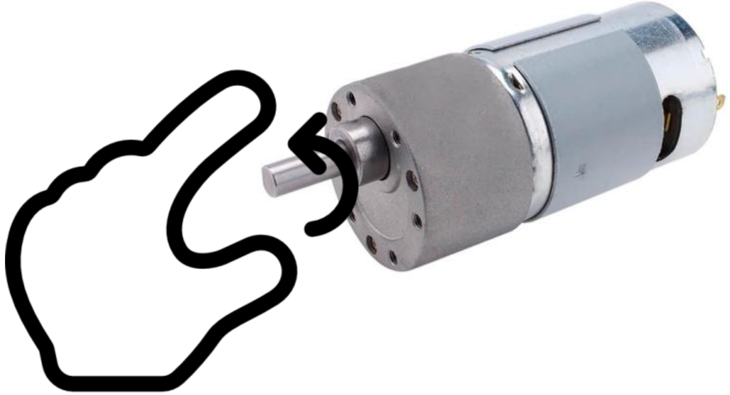
Message (Enter to send message to 'ESP...

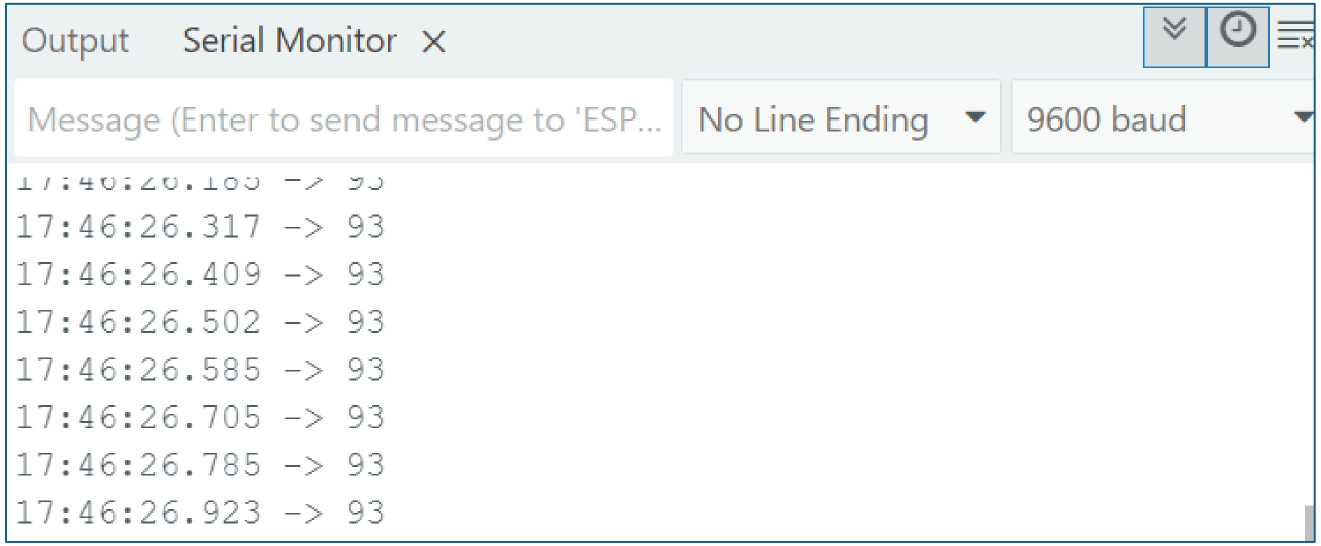
No Line Ending

9600 baud

```
17:45:42.532 -> 0
17:45:42.626 -> 0
17:45:42.719 -> 0
17:45:42.812 -> 0
17:45:42.886 -> 0
17:45:43.014 -> 0
17:45:43.105 -> 0
17:45:43.186 -> 0
```

Rotate motor by hand





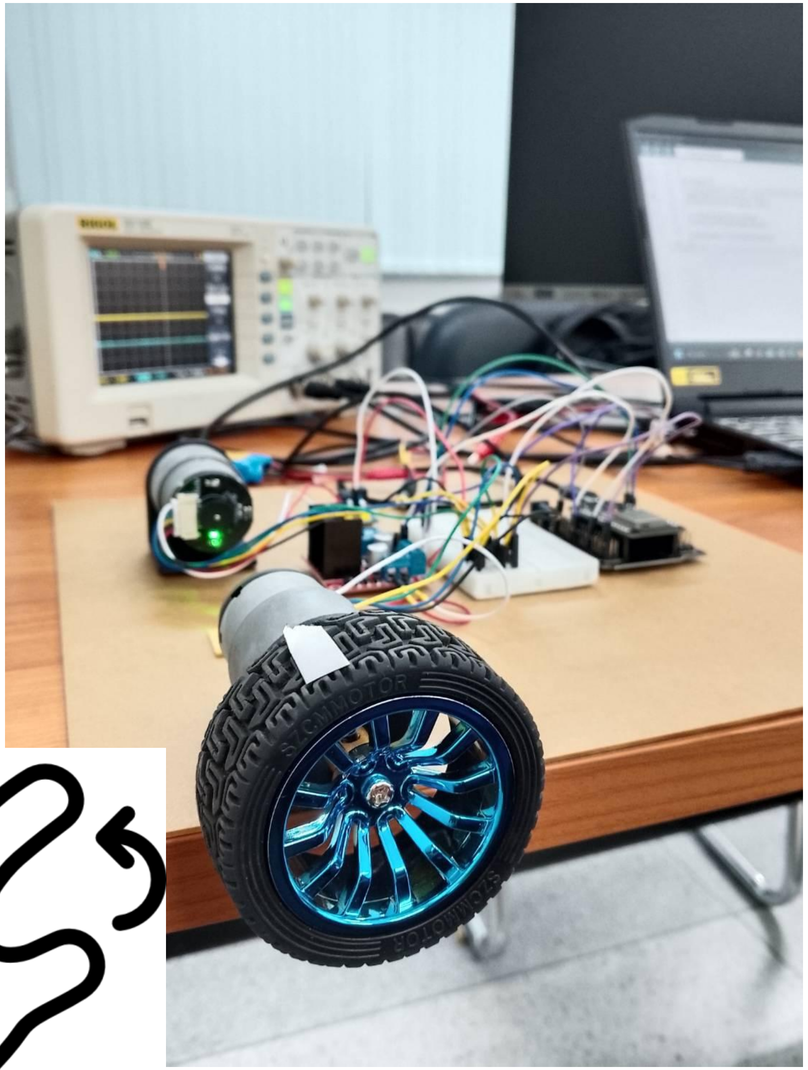
Message (Enter to send message to 'ESP...

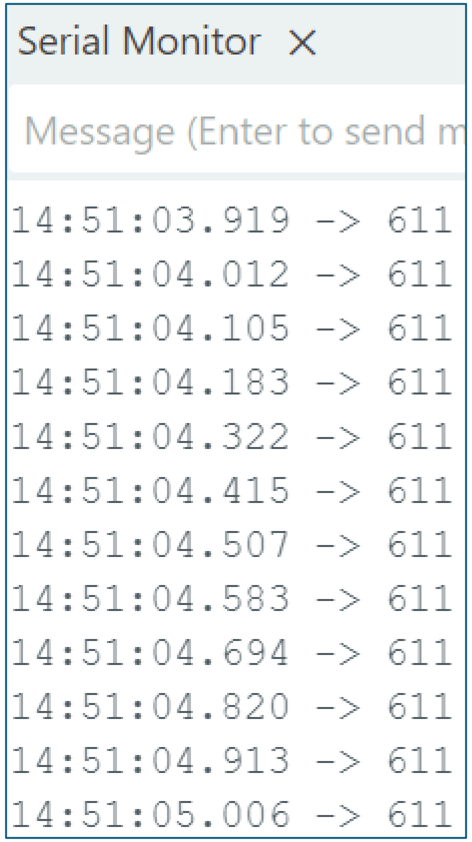
No Line Ending

9600 baud

```
17:46:26.185 -> 93
17:46:26.317 -> 93
17:46:26.409 -> 93
17:46:26.502 -> 93
17:46:26.585 -> 93
17:46:26.705 -> 93
17:46:26.785 -> 93
17:46:26.923 -> 93
```

Turn the wheel once and measure the encoder count.





```
14:51:03.919 -> 611
14:51:04.012 -> 611
14:51:04.105 -> 611
14:51:04.183 -> 611
14:51:04.322 -> 611
14:51:04.415 -> 611
14:51:04.507 -> 611
14:51:04.583 -> 611
14:51:04.694 -> 611
14:51:04.820 -> 611
14:51:04.913 -> 611
14:51:05.006 -> 611
```

$$Round = \frac{Count}{Pulses\ per\ revolutionCounts} = \frac{Count}{611}$$

Content

1. Motor encoder

    1.1 Motor encoder by Magnet sensor

    1.2 Use of External interrupt for faster reading encoder

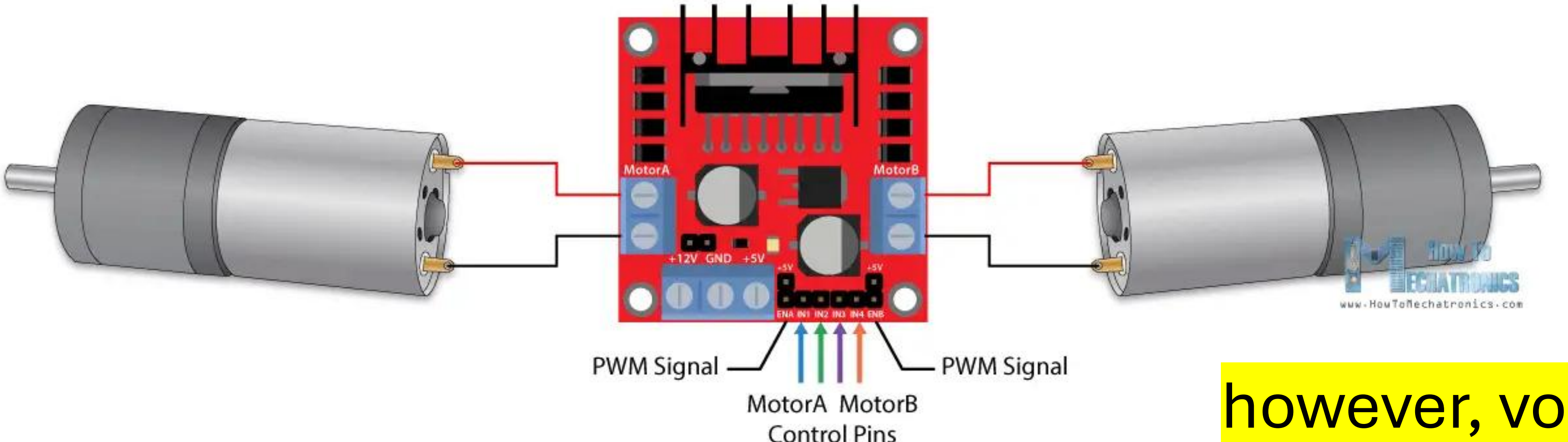
2. Control Movement and Position Motor

3. Control Speed Motor by Pulse Width Modulation (PWM)

4. Measure speed motor by IR sensors and Oscilloscope

5. Measure speed motor by Encoder Motor

2. Motor rotation using the L298N



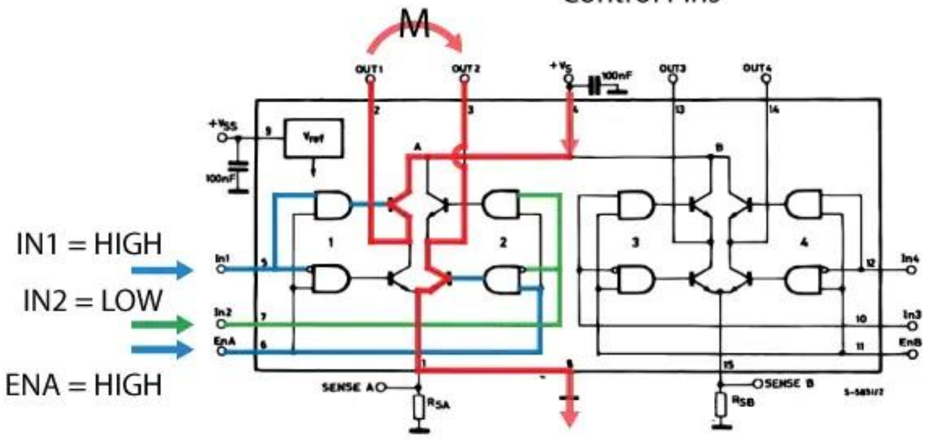
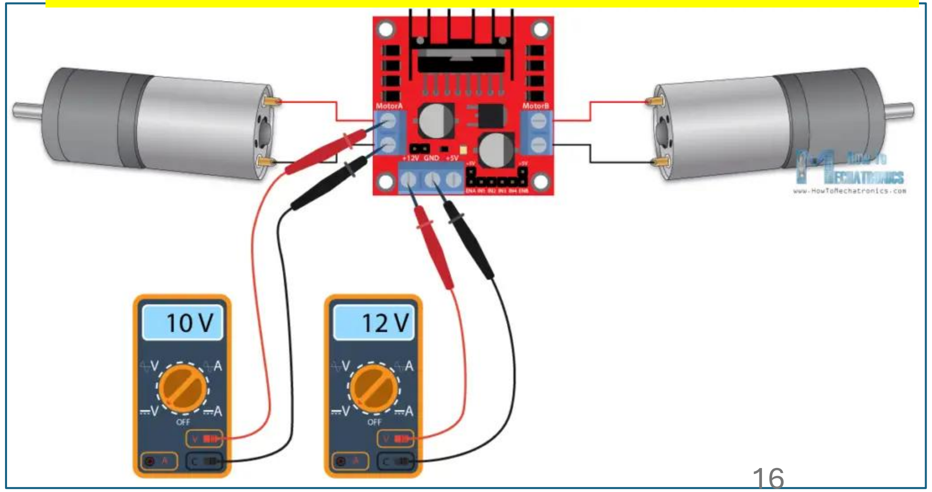


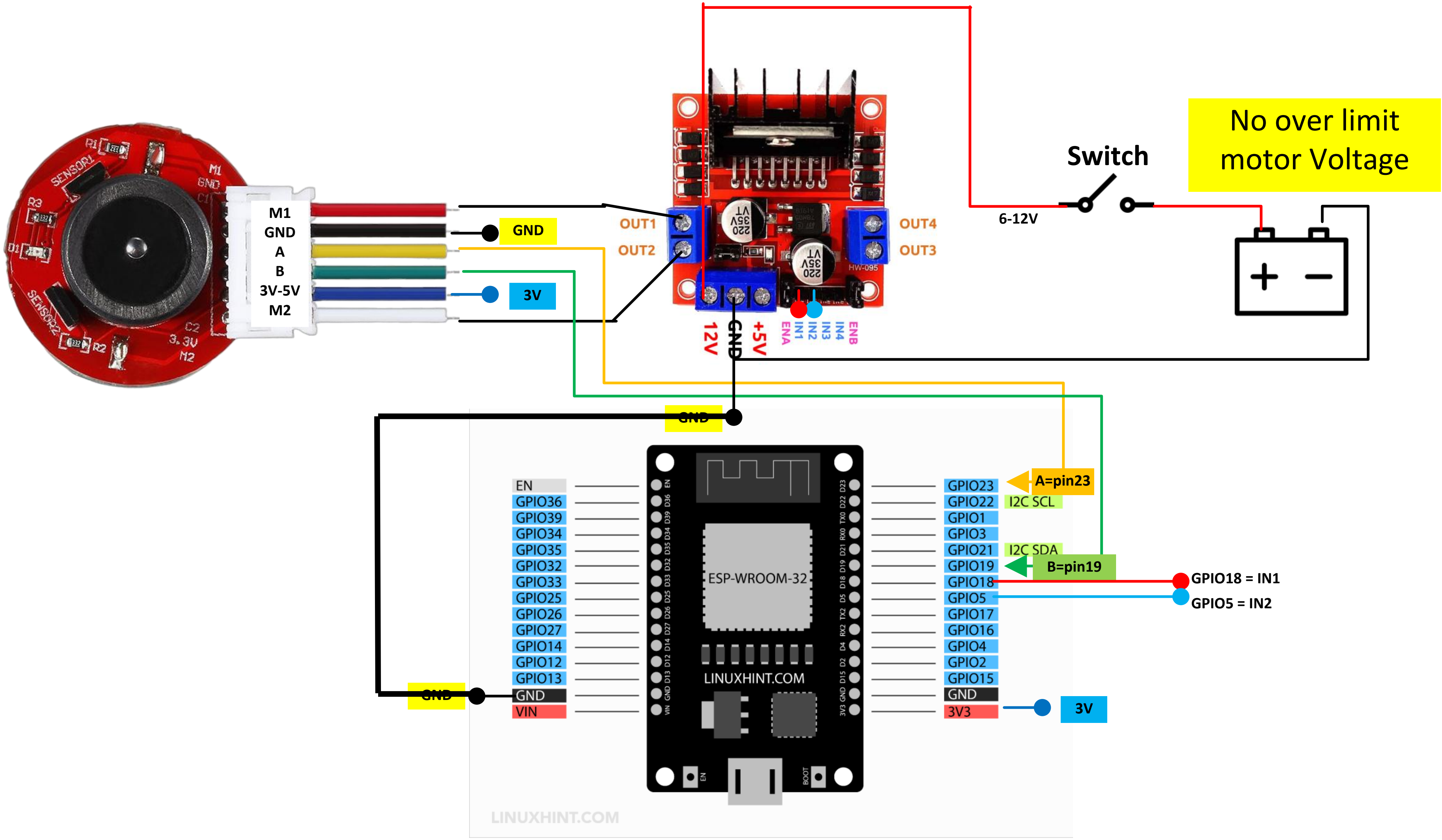
*Figure from L298N datasheet

L298N Block Diagram

however, voltage drop!!!

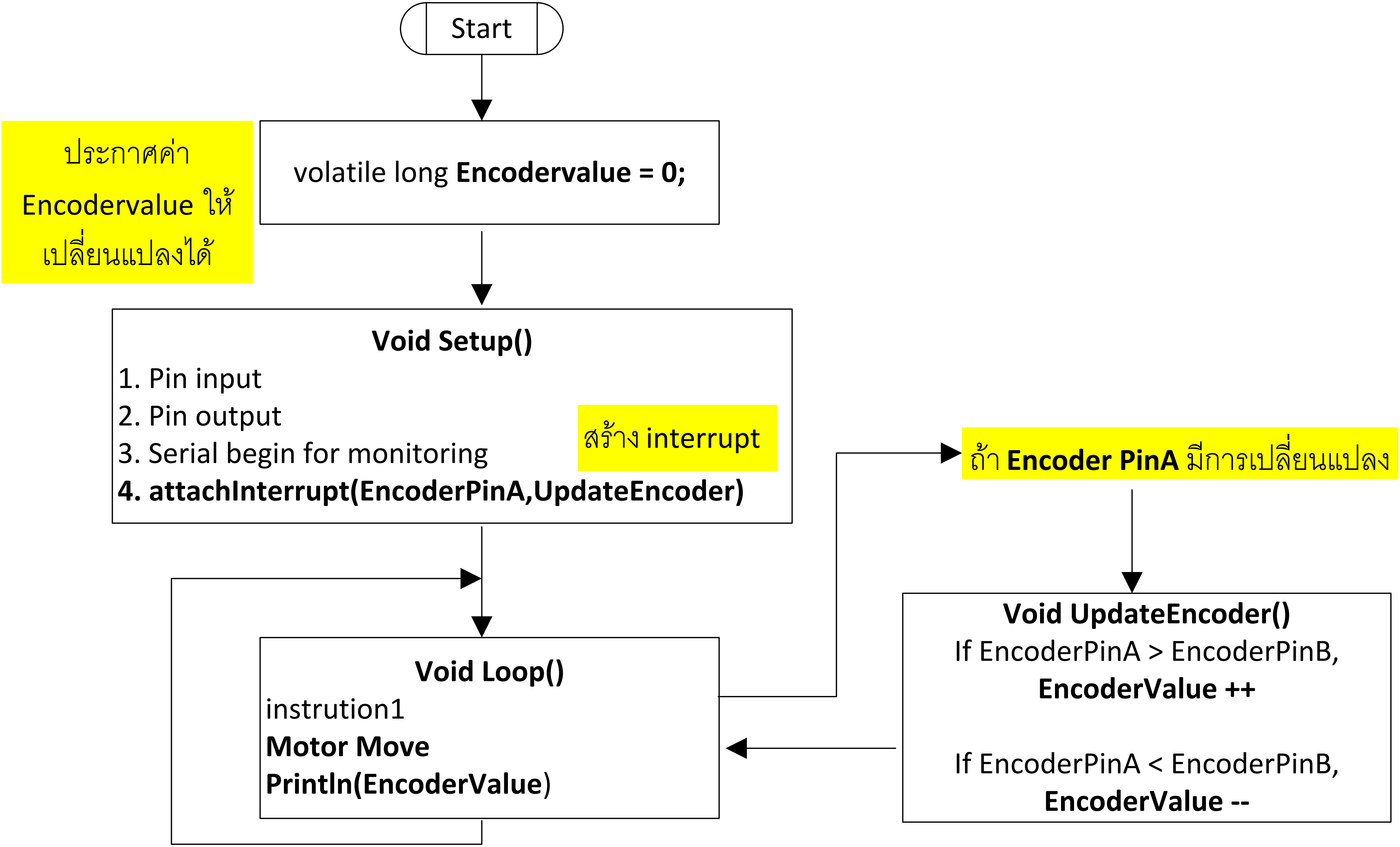


Connecting2.1 Motor rotate by L298N



No over limit motor Voltage

Flowchart2.1 Motor Move and Read Encoder



Start

ประกาศค่า Encodervalue ให้เปลี่ยนแปลงได้

volatile long **Encodervalue = 0;**

Void Setup()

1. Pin input
2. Pin output
3. Serial begin for monitoring
4. **attachInterrupt(EncoderPinA,UpdateEncoder)**

สร้าง interrupt

Void Loop()

instrution1

Motor Move

Println(EncoderValue)

ถ้า **Encoder PinA** มีการเปลี่ยนแปลง

Void UpdateEncoder()

If EncoderPinA > EncoderPinB,

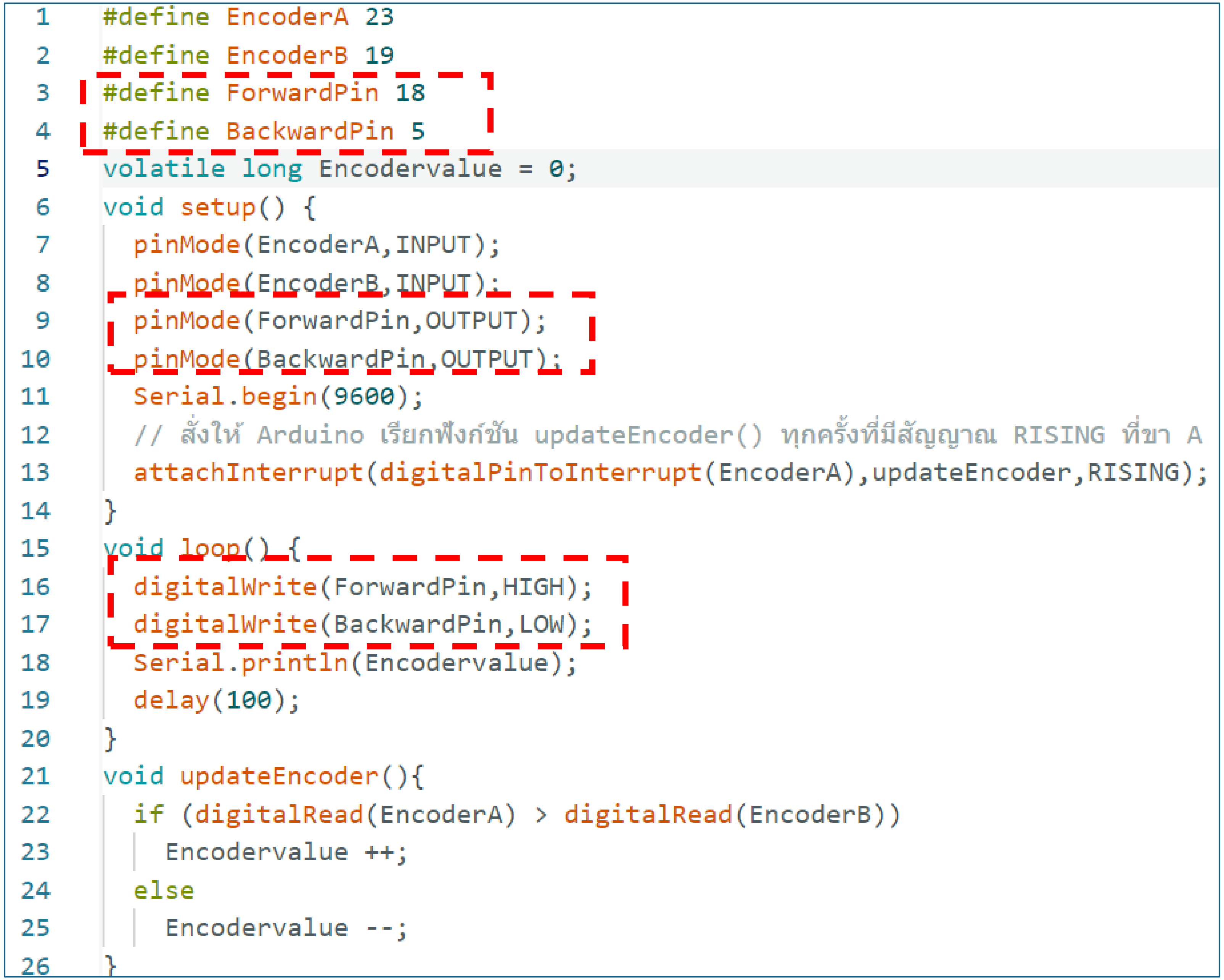
EncoderValue ++

If EncoderPinA < EncoderPinB,

EncoderValue --

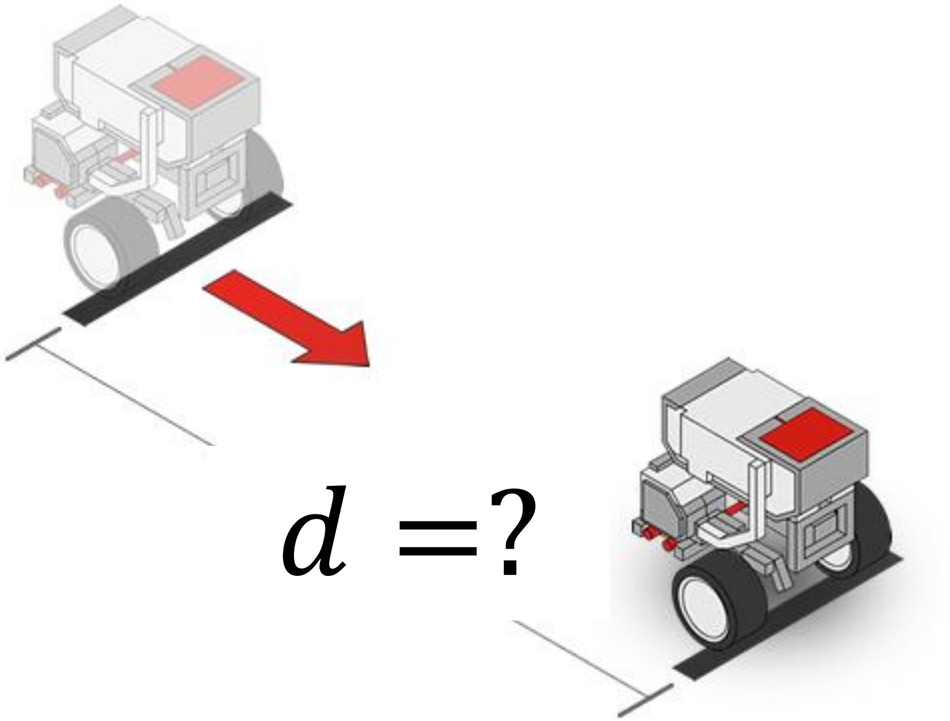
Example2.1

Motor Move



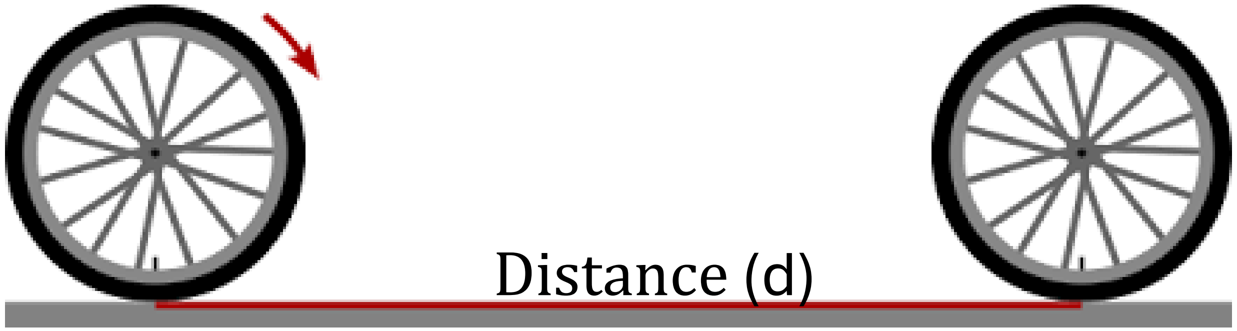
```
#define EncoderA 23
#define EncoderB 19
#define ForwardPin 18
#define BackwardPin 5
volatile long Encodervalue = 0;
void setup() {
  pinMode(EncoderA,INPUT);
  pinMode(EncoderB,INPUT);
  pinMode(ForwardPin,OUTPUT);
  pinMode(BackwardPin,OUTPUT);
  Serial.begin(9600);
  // สั่งให้ Arduino เรียกฟังก์ชัน updateEncoder() ทุกครั้งที่มีสัญญาณ RISING ที่ขา A
  attachInterrupt(digitalPinToInterrupt(EncoderA),updateEncoder,RISING);
}
void loop() {
  digitalWrite(ForwardPin,HIGH);
  digitalWrite(BackwardPin,LOW);
  Serial.println(Encodervalue);
  delay(100);
}
void updateEncoder(){
  if (digitalRead(EncoderA) > digitalRead(EncoderB))
    Encodervalue ++;
  else
    Encodervalue --;
}
```

Homework1: Compute the distance of robot movement from encoder and write the flowchart



$$d = ?$$

$$s = 2\pi r$$

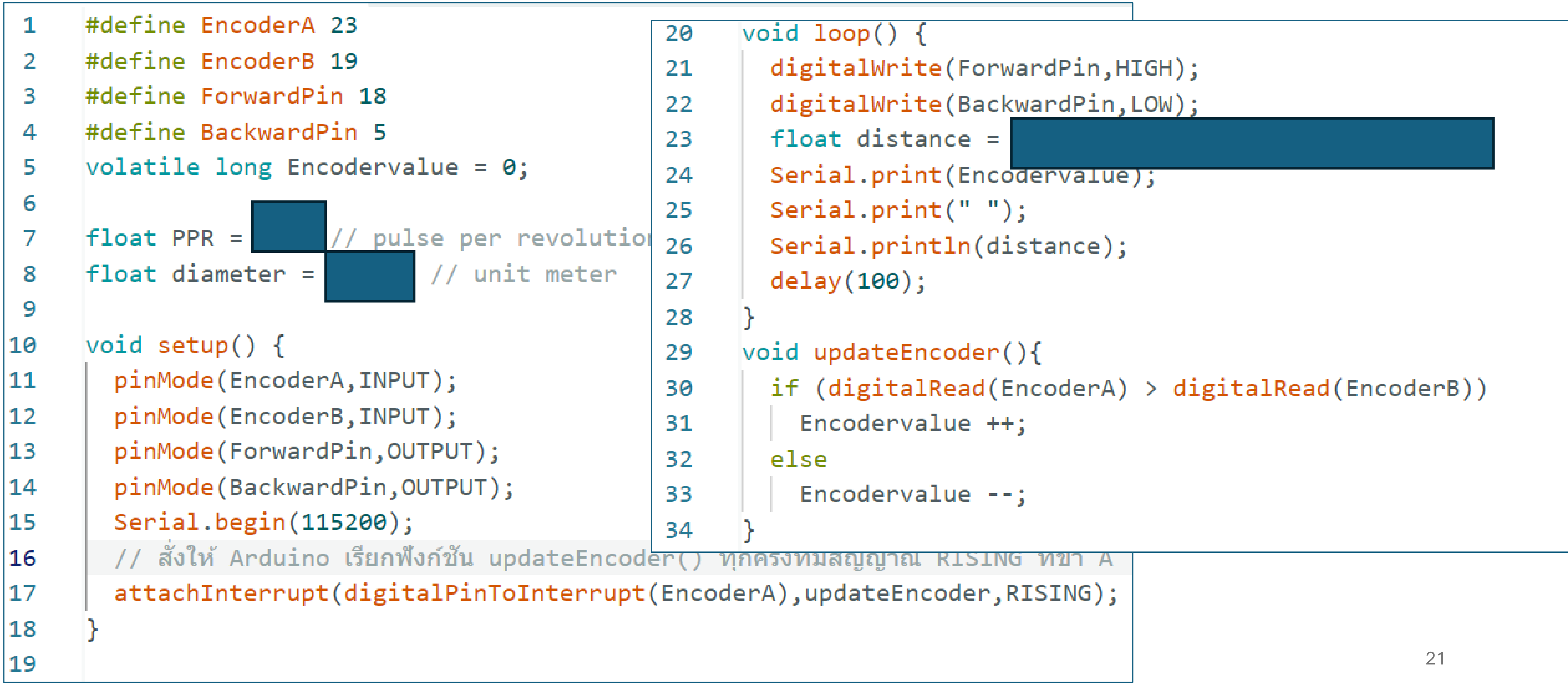


$$d = \frac{\text{EncoderValue}}{PPR} \times 2\pi r$$

Where,

PPR (Pulse Per Revolution) = ค่า encodervalue ที่อ่านได้ใน 1 รอบ

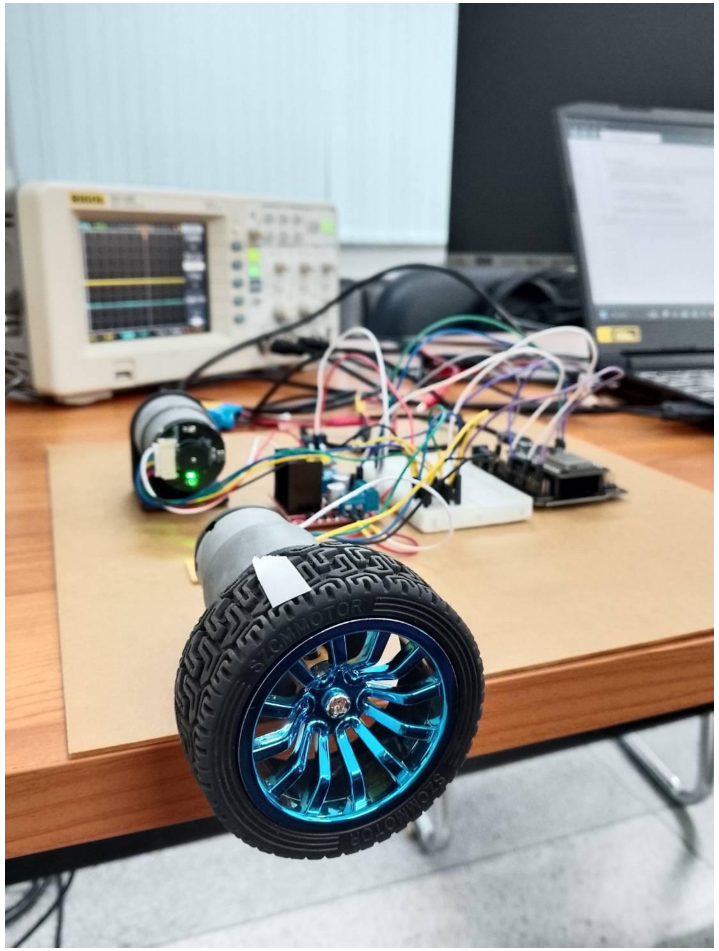
สำหรับล้อนี้ PPR ประมาณ 611

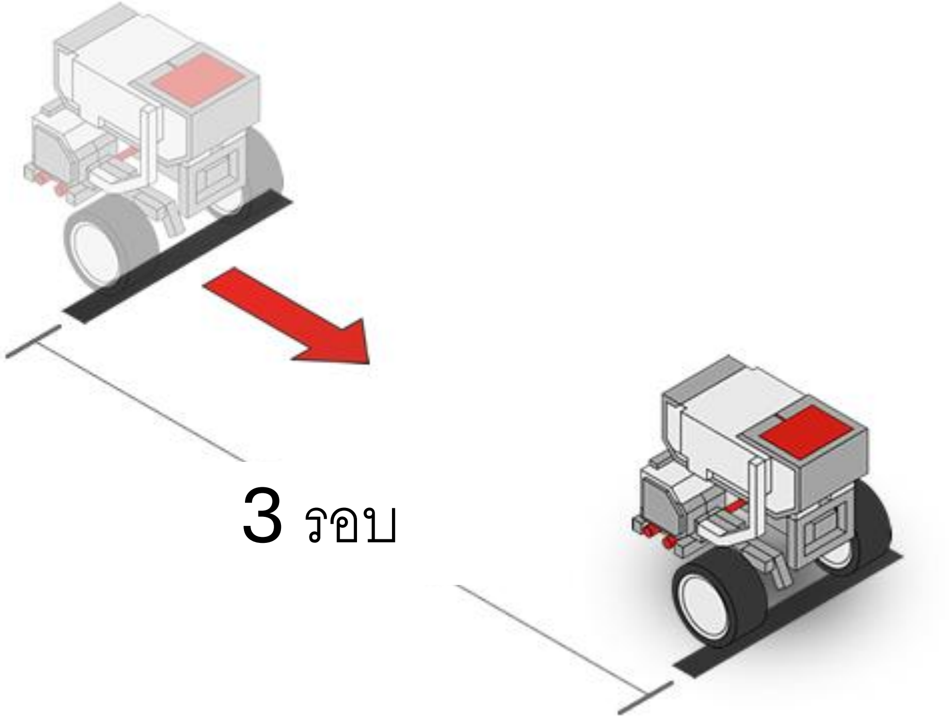


```
#define EncoderA 23
#define EncoderB 19
#define ForwardPin 18
#define BackwardPin 5
volatile long Encodervalue = 0;

float PPR = [redacted] // pulse per revolution
float diameter = [redacted]  // unit meter

void setup() {
  pinMode(EncoderA,INPUT);
  pinMode(EncoderB,INPUT);
  pinMode(ForwardPin,OUTPUT);
  pinMode(BackwardPin,OUTPUT);
  Serial.begin(115200);
  // สั่งให้ Arduino เรียกฟังก์ชัน updateEncoder() ทุกครั้งที่มีสัญญาณ RISING ที่ขา A
  attachInterrupt(digitalPinToInterrupt(EncoderA),updateEncoder,RISING);
}
```

```
void loop() {
  digitalWrite(ForwardPin,HIGH);
  digitalWrite(BackwardPin,LOW);
  float distance = [redacted]
  Serial.print(Encodervalue);
  Serial.print(" ");
  Serial.println(distance);
  delay(100);
}
void updateEncoder(){
  if (digitalRead(EncoderA) > digitalRead(EncoderB))
    Encodervalue ++;
  else
    Encodervalue --;
}
```

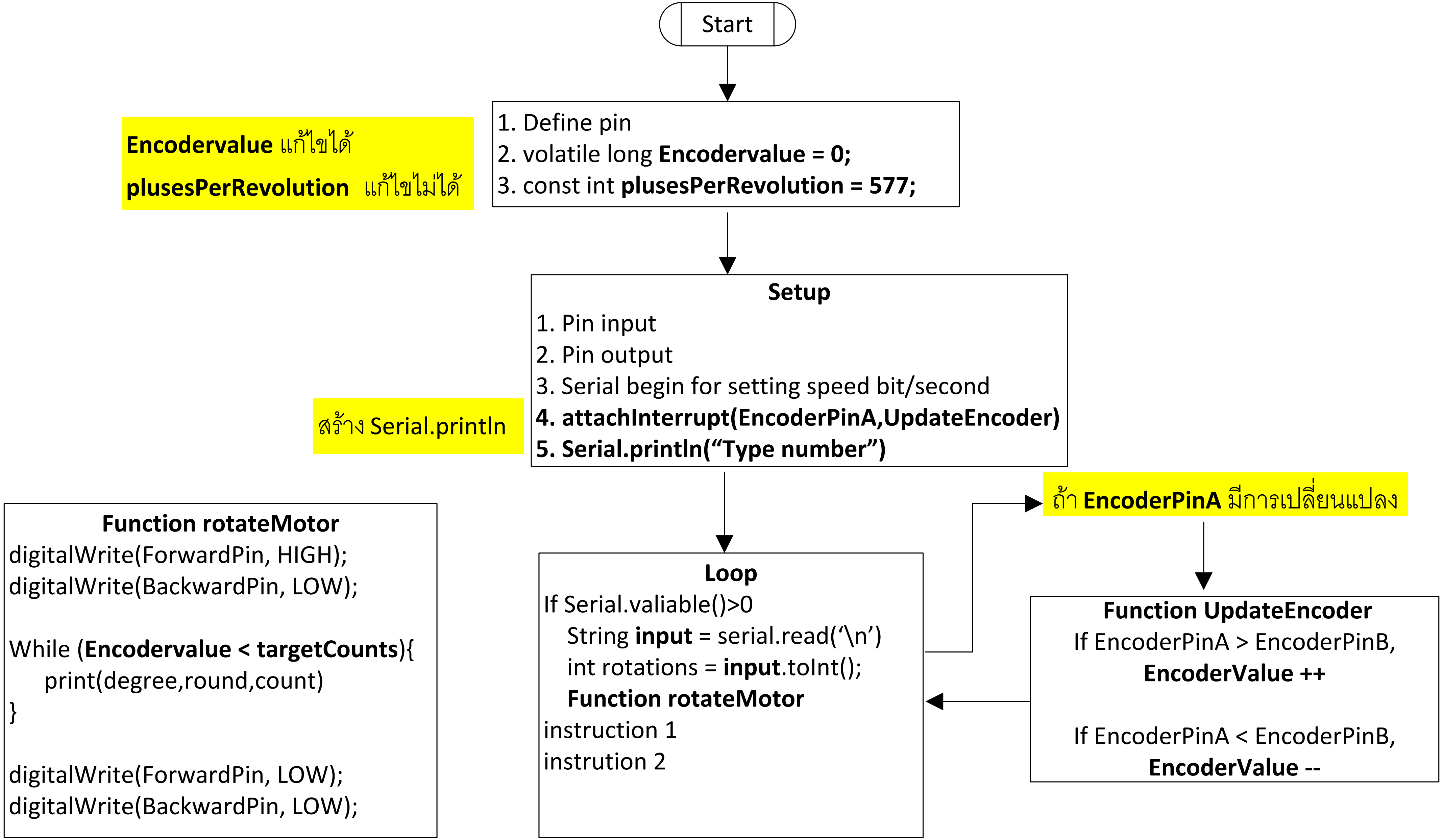
Example2.2 Determine the number of round for move





3 รอบ

Flowchart2.2 for control round of motor



Start

1. Define pin
2. volatile long **Encodervalue = 0;**
3. const int **plusesPerRevolution = 577;**

Encodervalue แก้ไขได้

plusesPerRevolution แก้ไขไม่ได้

Setup

1. Pin input
2. Pin output
3. Serial begin for setting speed bit/second
4. **attachInterrupt(EncoderPinA,UpdateEncoder)**
5. **Serial.println("Type number")**

สร้าง Serial.println

ถ้า **EncoderPinA** มีการเปลี่ยนแปลง

Function rotateMotor

```
digitalWrite(ForwardPin, HIGH);
digitalWrite(BackwardPin, LOW);

While (Encodervalue < targetCounts){
     print(degree,round,count)
}

digitalWrite(ForwardPin, LOW);
digitalWrite(BackwardPin, LOW);
```

Loop

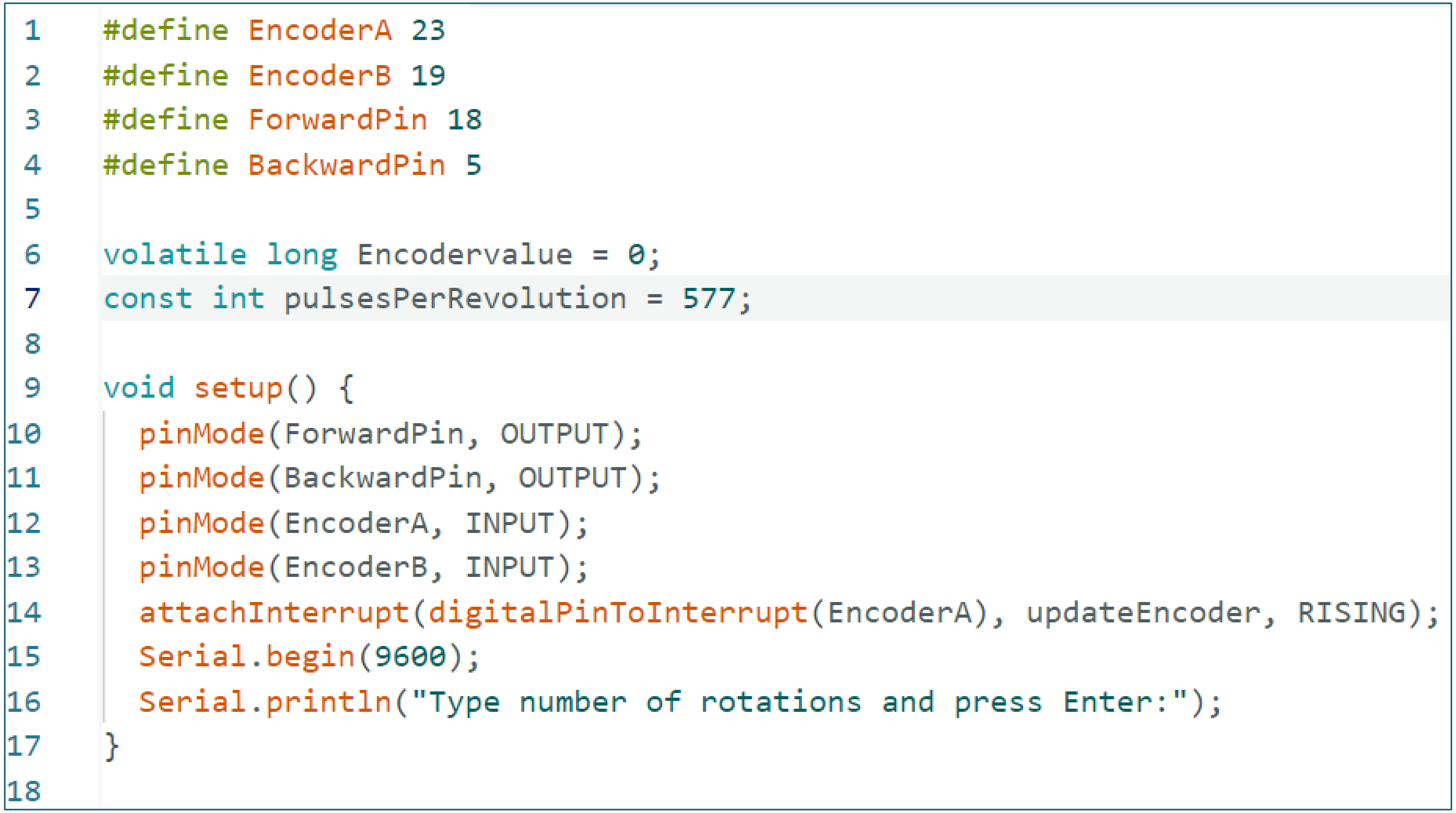
```
If Serial.valiable()>0
   String input = serial.read('\n')
   int rotations = input.toInt();
   Function rotateMotor
instruction 1
instrution 2
```

Function UpdateEncoder

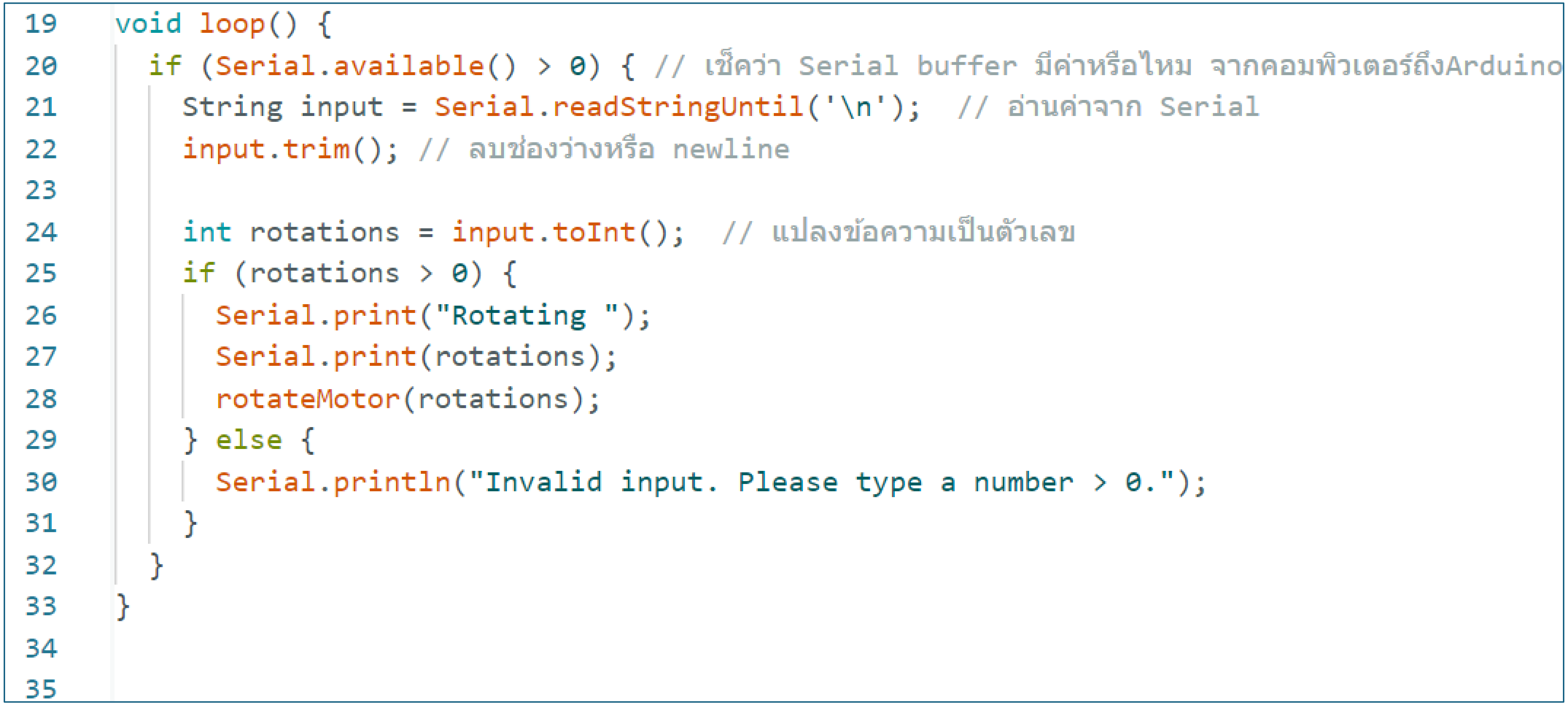
If EncoderPinA > EncoderPinB,
EncoderValue ++

If EncoderPinA < EncoderPinB,
EncoderValue --

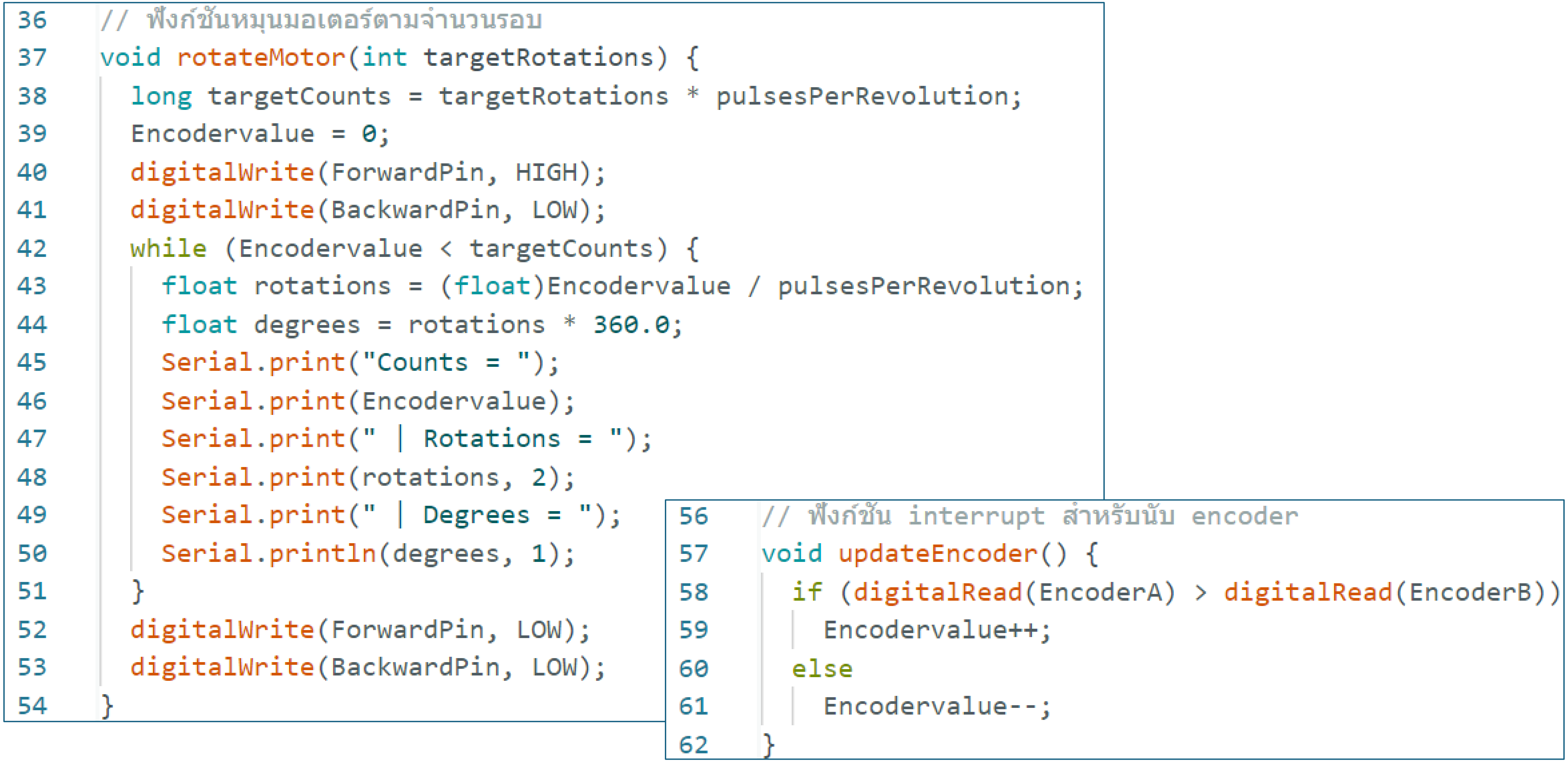
Code of Example2.2 control round of motor



```
#define EncoderA 23
#define EncoderB 19
#define ForwardPin 18
#define BackwardPin 5

volatile long Encodervalue = 0;
const int pulsesPerRevolution = 577;

void setup() {
  pinMode(ForwardPin, OUTPUT);
  pinMode(BackwardPin, OUTPUT);
  pinMode(EncoderA, INPUT);
  pinMode(EncoderB, INPUT);
  attachInterrupt(digitalPinToInterrupt(EncoderA), updateEncoder, RISING);
  Serial.begin(9600);
  Serial.println("Type number of rotations and press Enter:");
}

```



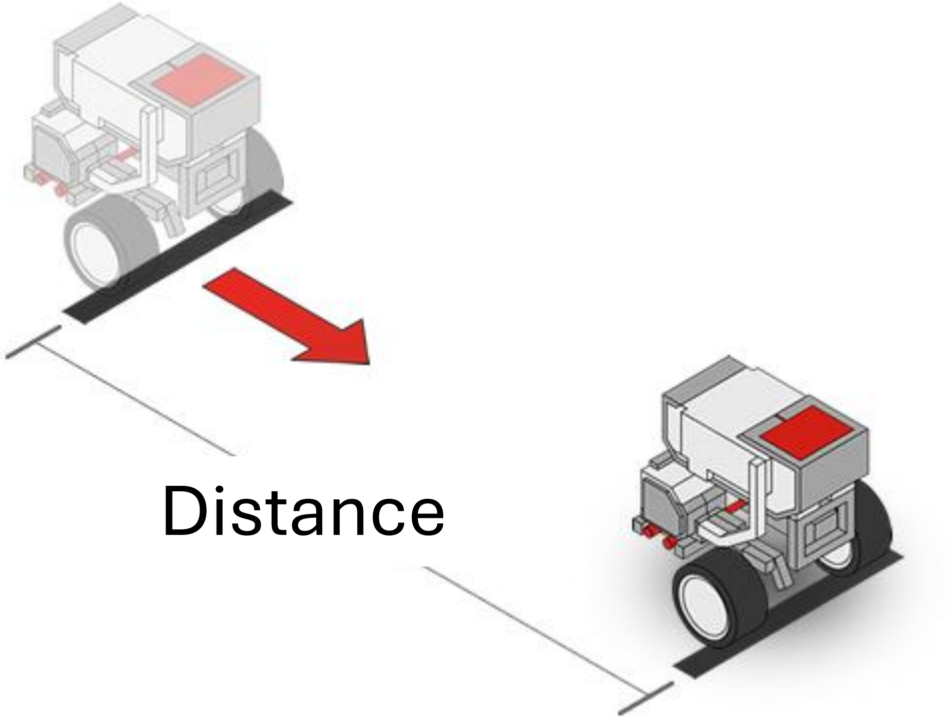
```
void loop() {
  if (Serial.available() > 0) { // เช็คว่า Serial buffer มีค่าหรือไหม จากคอมพิวเตอร์ถึงArduino
    String input = Serial.readStringUntil('\n');  // อ่านค่าจาก Serial
    input.trim(); // ลบช่องว่างหรือ newline

    int rotations = input.toInt();  // แปลงข้อความเป็นตัวเลข
    if (rotations > 0) {
      Serial.print("Rotating ");
      Serial.print(rotations);
      rotateMotor(rotations);
    } else {
      Serial.println("Invalid input. Please type a number > 0.");
    }
  }
}
```



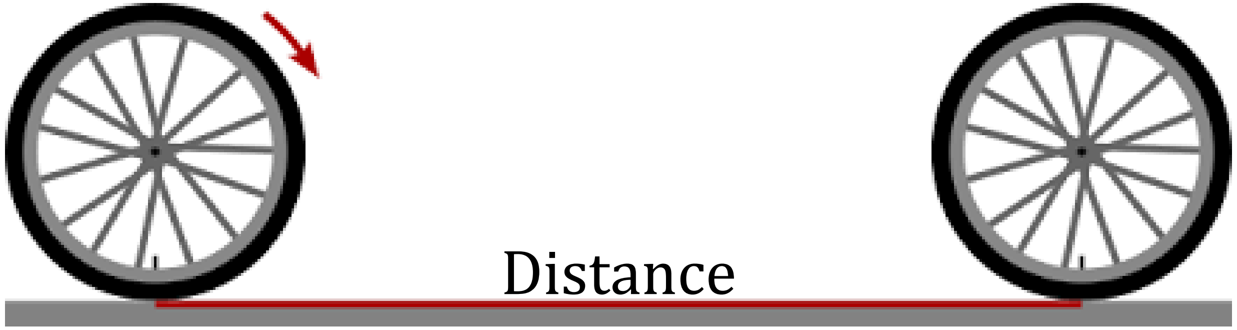
```
// ฟังก์ชันหมุนมอเตอร์ตามจำนวนรอบ
void rotateMotor(int targetRotations) {
  long targetCounts = targetRotations * pulsesPerRevolution;
  Encodervalue = 0;
  digitalWrite(ForwardPin, HIGH);
  digitalWrite(BackwardPin, LOW);
  while (Encodervalue < targetCounts) {
    float rotations = (float)Encodervalue / pulsesPerRevolution;
    float degrees = rotations * 360.0;
    Serial.print("Counts = ");
    Serial.print(Encodervalue);
    Serial.print(" | Rotations = ");
    Serial.print(rotations, 2);
    Serial.print(" | Degrees = ");
    Serial.println(degrees, 1);
  }
  digitalWrite(ForwardPin, LOW);
  digitalWrite(BackwardPin, LOW);
}
```

```
// ฟังก์ชัน interrupt สำหรับนับ encoder
void updateEncoder() {
  if (digitalRead(EncoderA) > digitalRead(EncoderB))
    Encodervalue++;
  else
    Encodervalue--;
}
```

Homework2. Define distance range for robot stop and write the flowchart



$$s = 2\pi r$$



$$\text{Wheel\_Round} = \frac{Distance}{s}$$, แต่ Round ที่ได้เป็นจำนวนทศนิยม

$$\text{pulseCount} = \text{Wheel\_Round} \times PPR$$

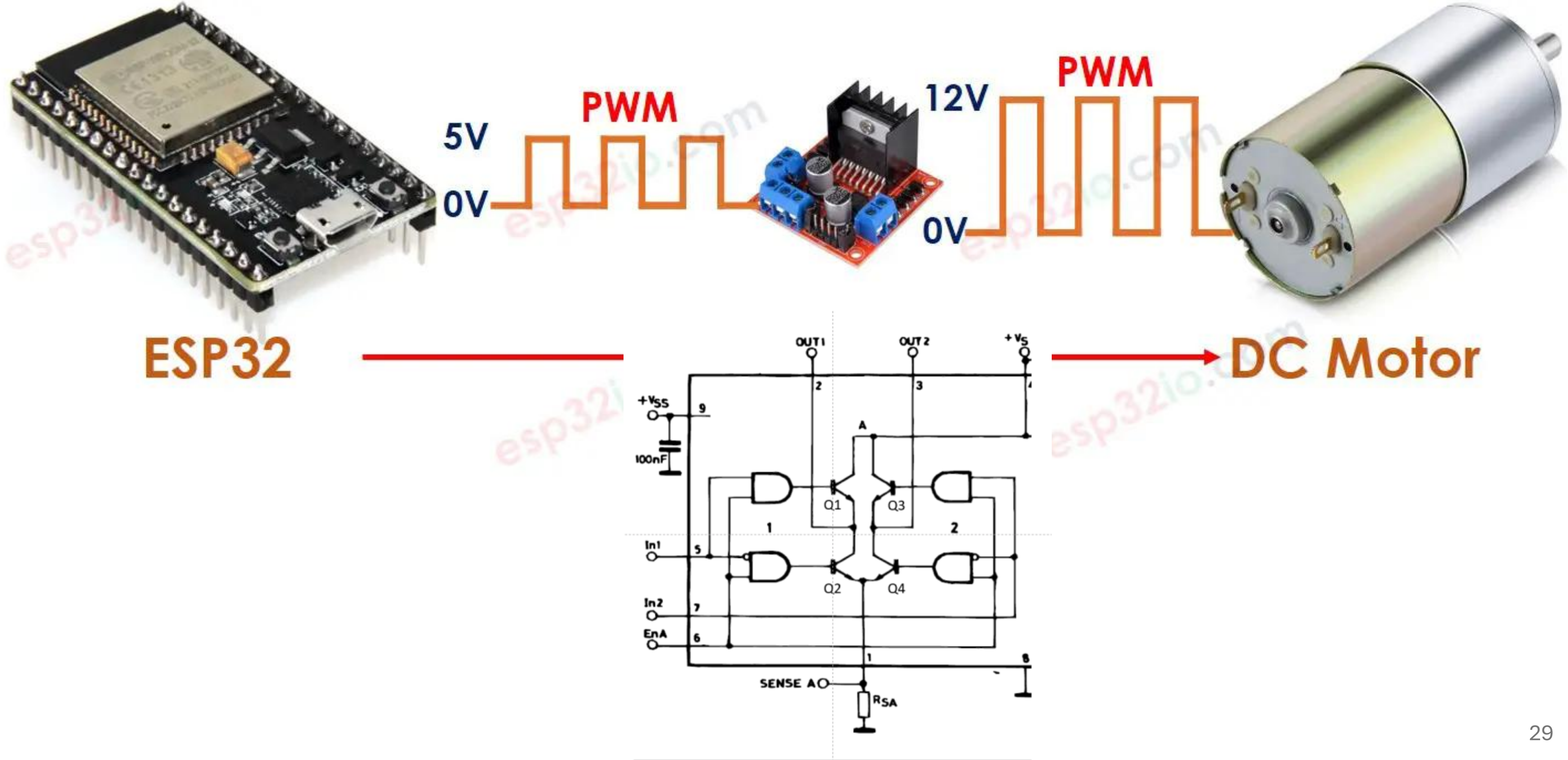
$$\boxed{\text{pulseCount} = \frac{Distance}{s} \times PPR}$$

สำหรับล้อนี้ PPR คือค่าที่ encoder อ่านได้ใน 1 รอบ คือ 611

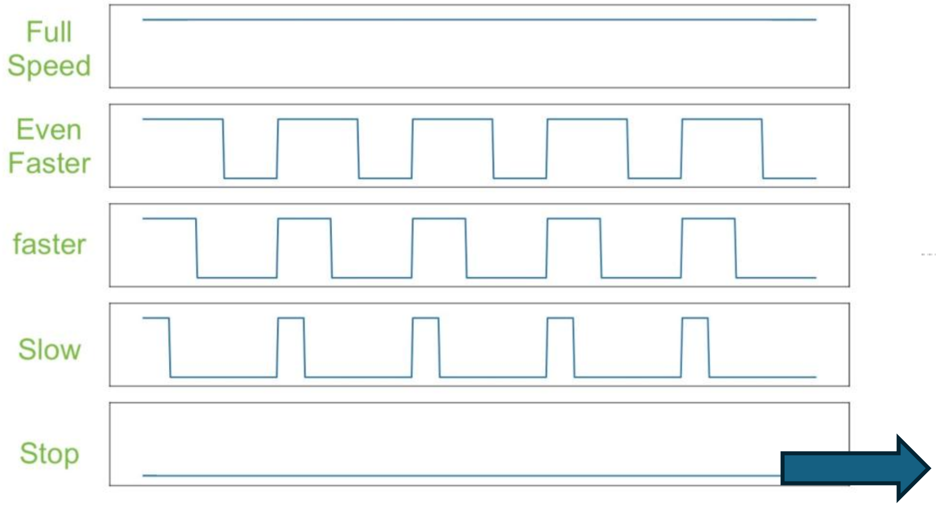
Content

1. Motor encoder
    - 1.1 Motor encoder by Magnet sensor
    - 1.2 Use of External interrupt for faster reading encoder
2. **Control Movement and Position Motor**
3. **Control Speed Motor by Pulse Width Modulation (PWM)**
4. Measure speed motor by IR sensors and Oscilloscope
5. Measure speed motor by Encoder Motor

3. Control speed Motor by Pulse Width Modulation (PWM)

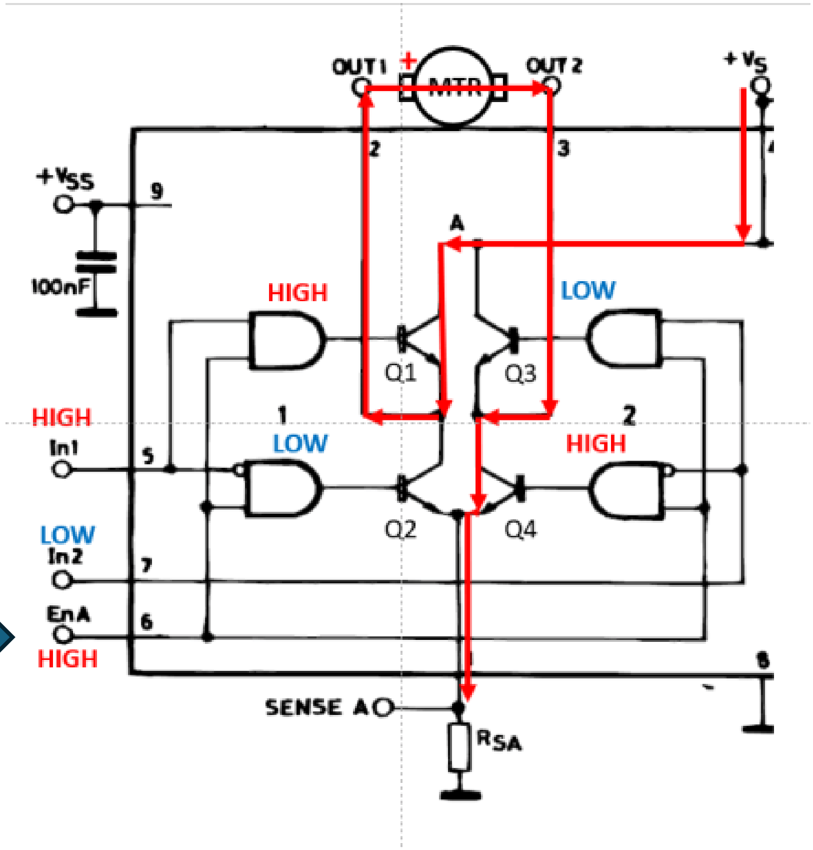


What is Enable Pin of L298N

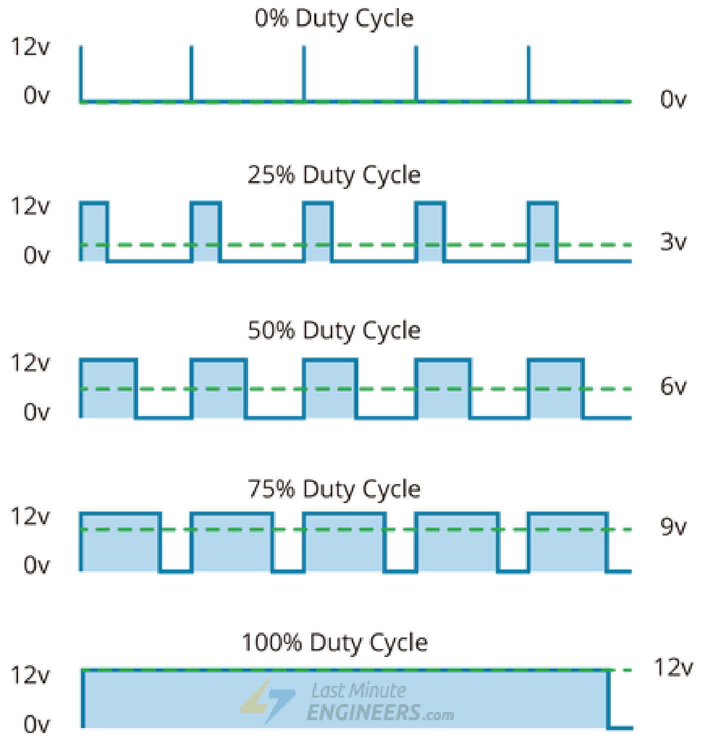


ENA

L298N

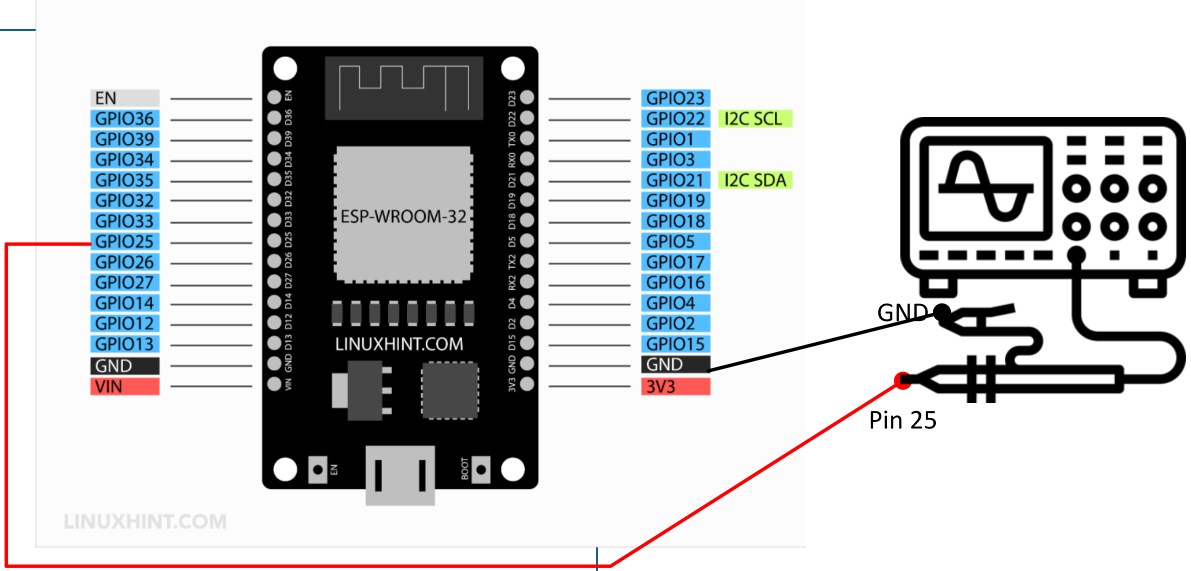


Motor voltage



Example3.1 Measuring PWM Signal

```
int enable1Pin = 25;

const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 100;

void setup() {
  pinMode(enable1Pin, OUTPUT);
  ledcAttachChannel(enable1Pin, freq, resolution, pwmChannel);
  ledcWrite(enable1Pin, dutyCycle);
}
void loop() {

}
```

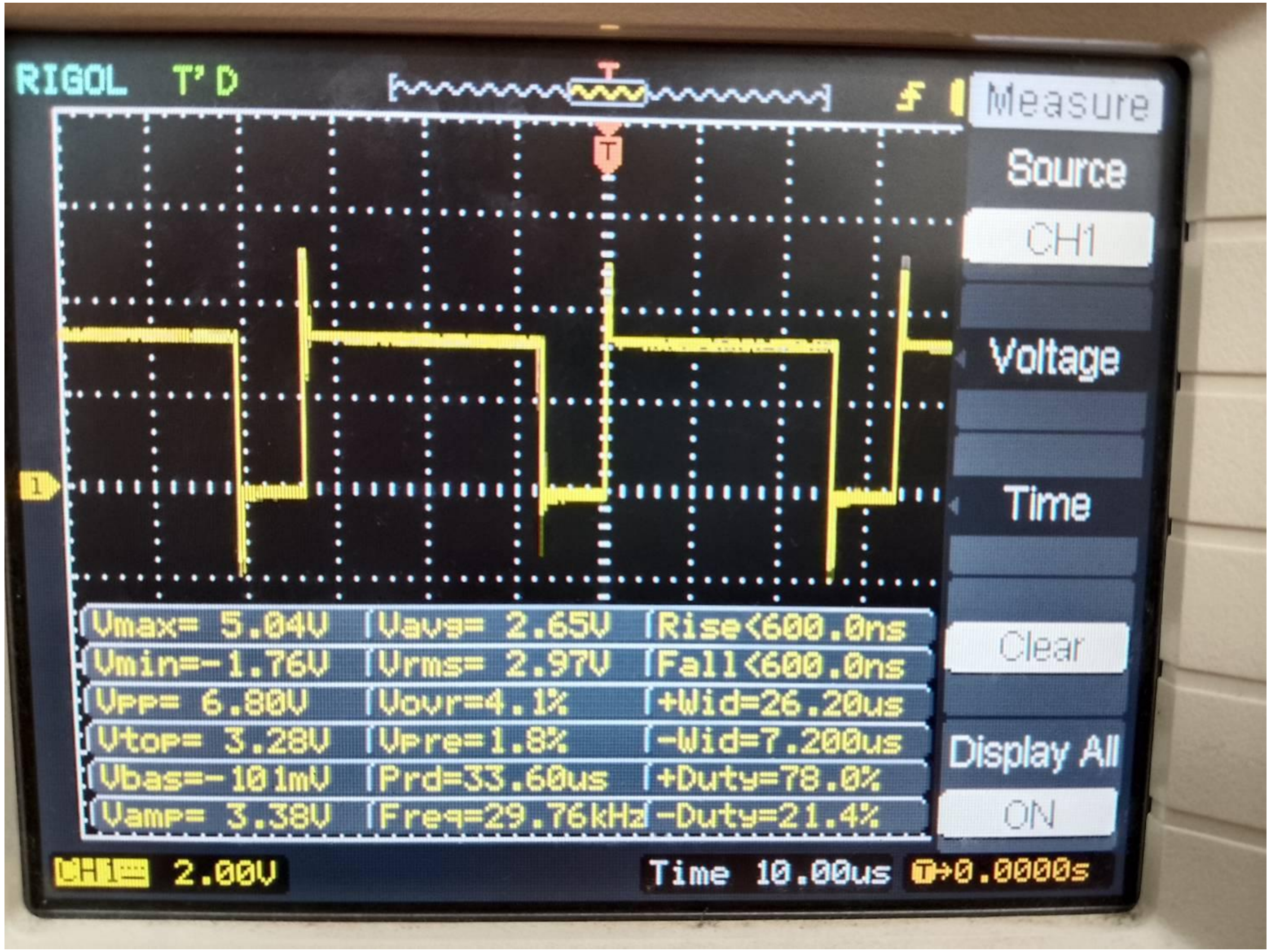


Choosing f=30 kHz

1.Avoid the typical range of human hearing (20-20 kHz)

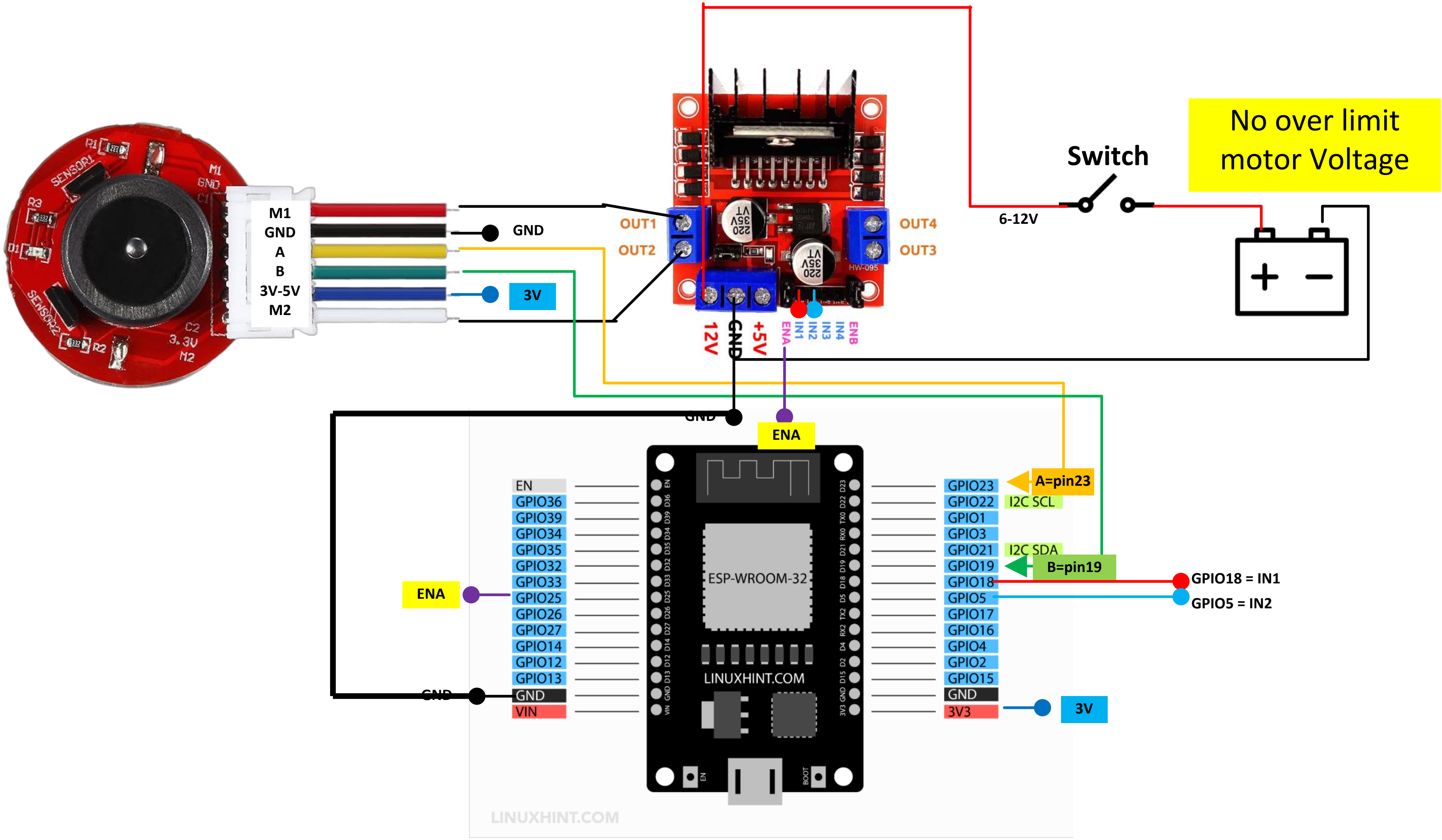
2. Reduce ripple noise.

Measure the PWM signal from ESP32

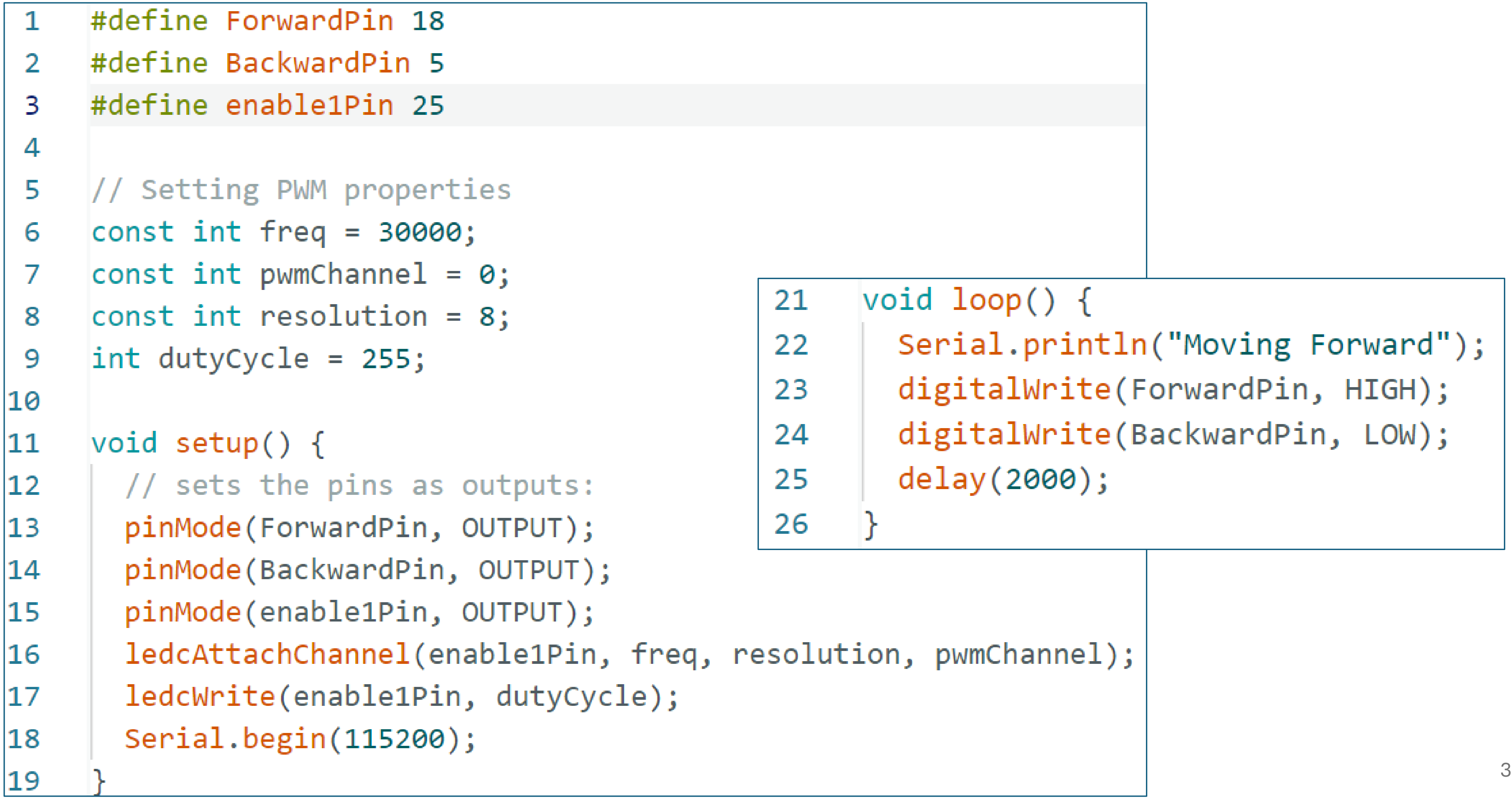


```
Vmax= 5.04V    Vavg= 2.65V    Rise<600.0ns
Vmin=-1.76V    Vrms= 2.97V    Fall<600.0ns
Vpp= 6.80V     Vovr=4.1%      +Wid=26.20us
Vtop= 3.28V    Vpre=1.8%      -Wid=7.200us
Vbas=-101mV    Prd=33.60us    +Duty=78.0%
Vamp= 3.38V    Freq=29.76kHz  -Duty=21.4%
CH1 2.00V              Time 10.00us  T→0.0000s
```

Connecting3.2 GPIO25 to ENA for control speed motor



Code of Connecting3.2 : GPIO25 to ENA for control speed motor



```
#define ForwardPin 18
#define BackwardPin 5
#define enable1Pin 25

// Setting PWM properties
const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 255;

void setup() {
  // sets the pins as outputs:
  pinMode(ForwardPin, OUTPUT);
  pinMode(BackwardPin, OUTPUT);
  pinMode(enable1Pin, OUTPUT);
  ledcAttachChannel(enable1Pin, freq, resolution, pwmChannel);
  ledcWrite(enable1Pin, dutyCycle);
  Serial.begin(115200);
}
```

```
void loop() {
  Serial.println("Moving Forward");
  digitalWrite(ForwardPin, HIGH);
  digitalWrite(BackwardPin, LOW);
  delay(2000);
}
```

Content

1. Motor encoder
    - 1.1 Motor encoder by Magnet sensor
    - 1.2 Use of External interrupt for faster reading encoder

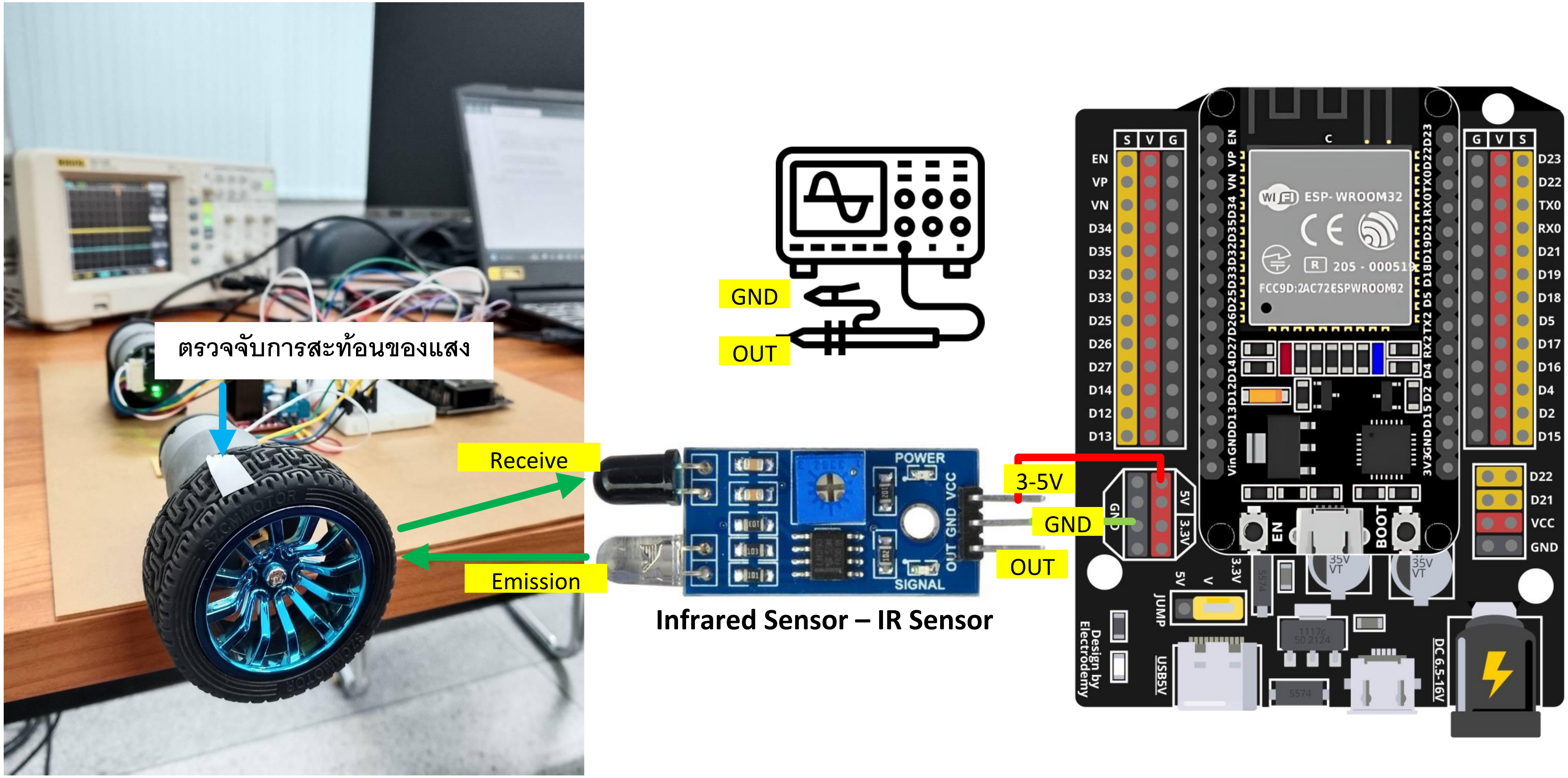
2. Control Movement and Position Motor

3. Control Speed Motor by Pulse Width Modulation (PWM)

4. Measure speed motor by IR sensors and Oscilloscope

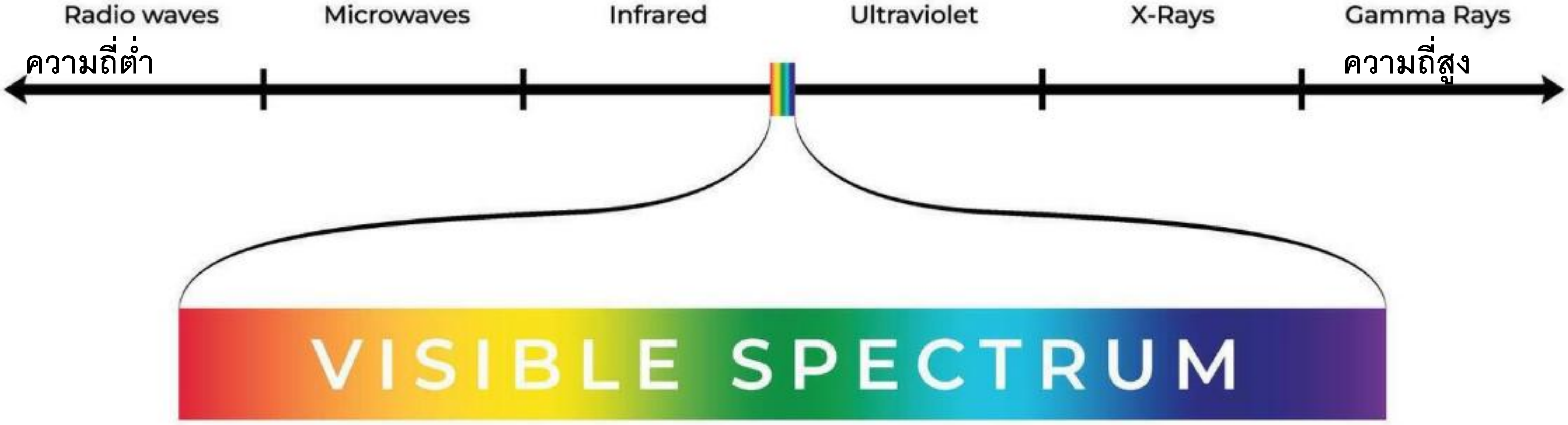
5. Measure speed motor by Encoder Motor

4. Measure speed motor by IR sensors and Oscilloscope



Electromagnetic frequency range

IR sensor



ความถี่ต่ำ

ความถี่สูง

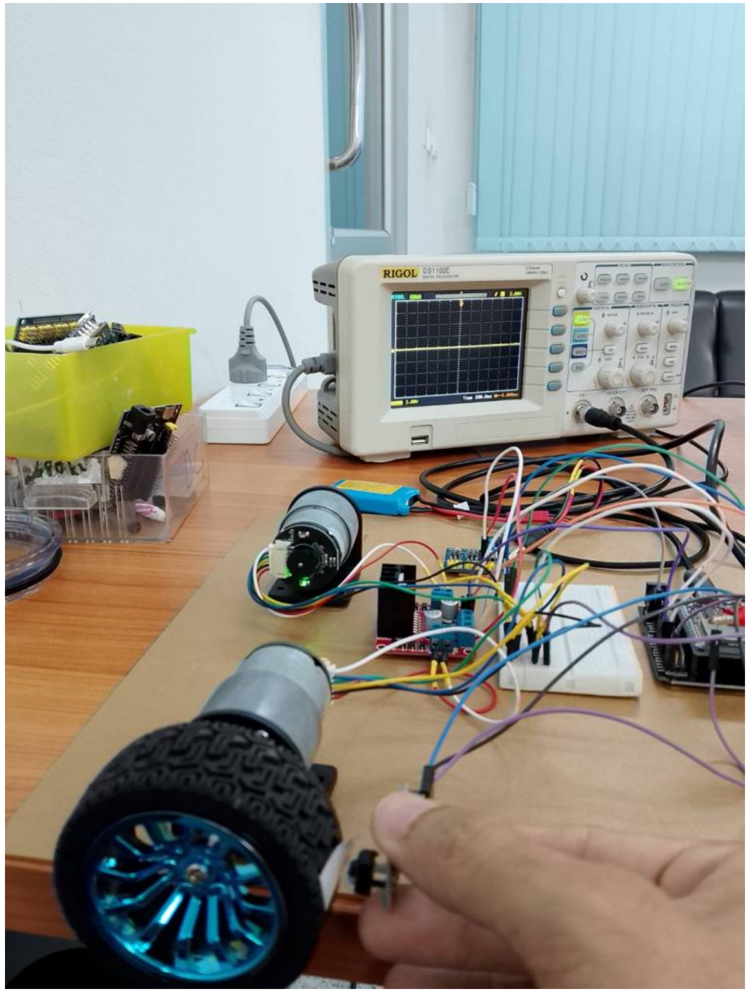
ตามองเห็น

Why we use IR sensor for detecting line

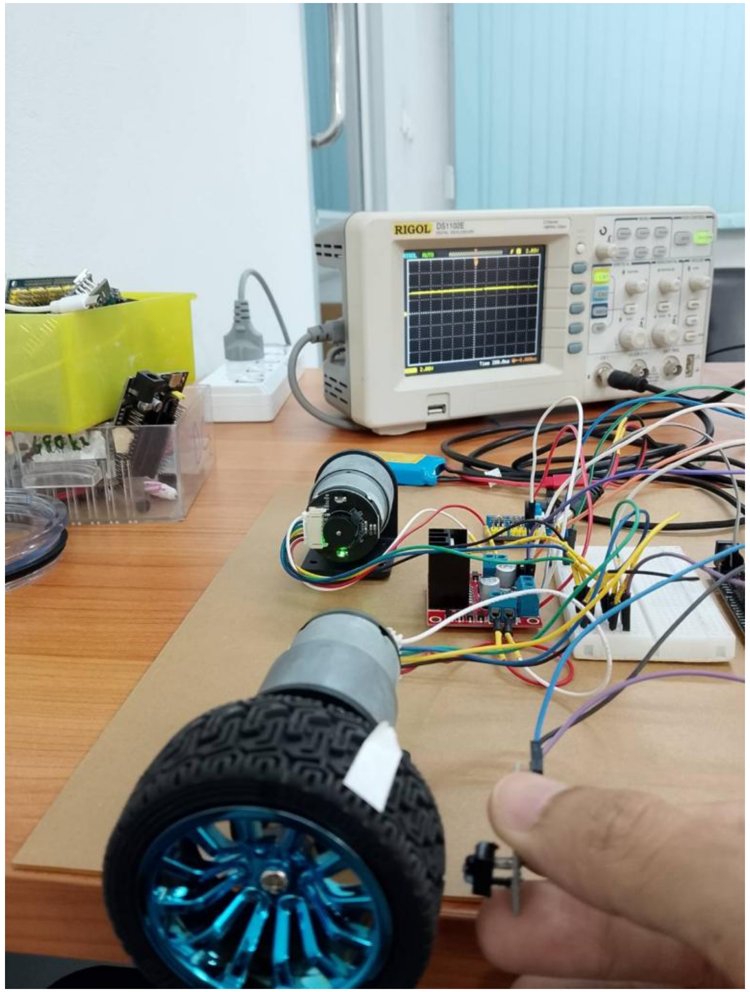
Region	Frequency Range	Wavelength Range	Typical Sources / Detectors	Detection Characteristics	Common Uses
Radio wave	3 kHz – 300 MHz	100 km – 1 m	Antenna, RF receiver	Passes through most materials; low resolution; long range	Communication, RFID, remote key fobs
Microwave	300 MHz – 300 GHz	1 m – 1 mm	Radar antenna, Gunn diode, MMIC radar sensor	Can penetrate fog/plastic; used for motion/range; medium resolution	Radar, motion detection, satellite link, speed gun
Infrared (IR)	300 GHz – 400 THz	1 mm – 0.7 µm	IR LED + photodiode/phototransistor	Needs line of sight; very fast; short range; sensitive to heat	Object detection, thermal sensing, motor encoder, TV remote
Visible light	400 THz – 790 THz	700 nm – 380 nm	Photodiode, phototransistor, camera CCD/CMOS	Works only in clear line of sight; affected by ambient light	Cameras, optical encoders, barcode scanners
Ultraviolet (UV)	790 THz – 30 PHz	380 nm – 10 nm	UV photodiode, photomultiplier tube	Higher energy, can cause fluorescence; hazardous to eyes/skin	Flame sensors, material inspection, sterilization
X-ray	30 PHz – 30 EHz	10 nm – 0.01 nm	Scintillation detector, X-ray CCD	Very high energy; penetrates most materials except metals	Medical imaging, industrial inspection
Gamma ray (γ)	> 30 EHz	< 0.01 nm	Geiger counter, scintillator	Extremely high energy, nuclear origin; needs shielding	Radiation detection, nuclear medicine, astrophysics

Calibration

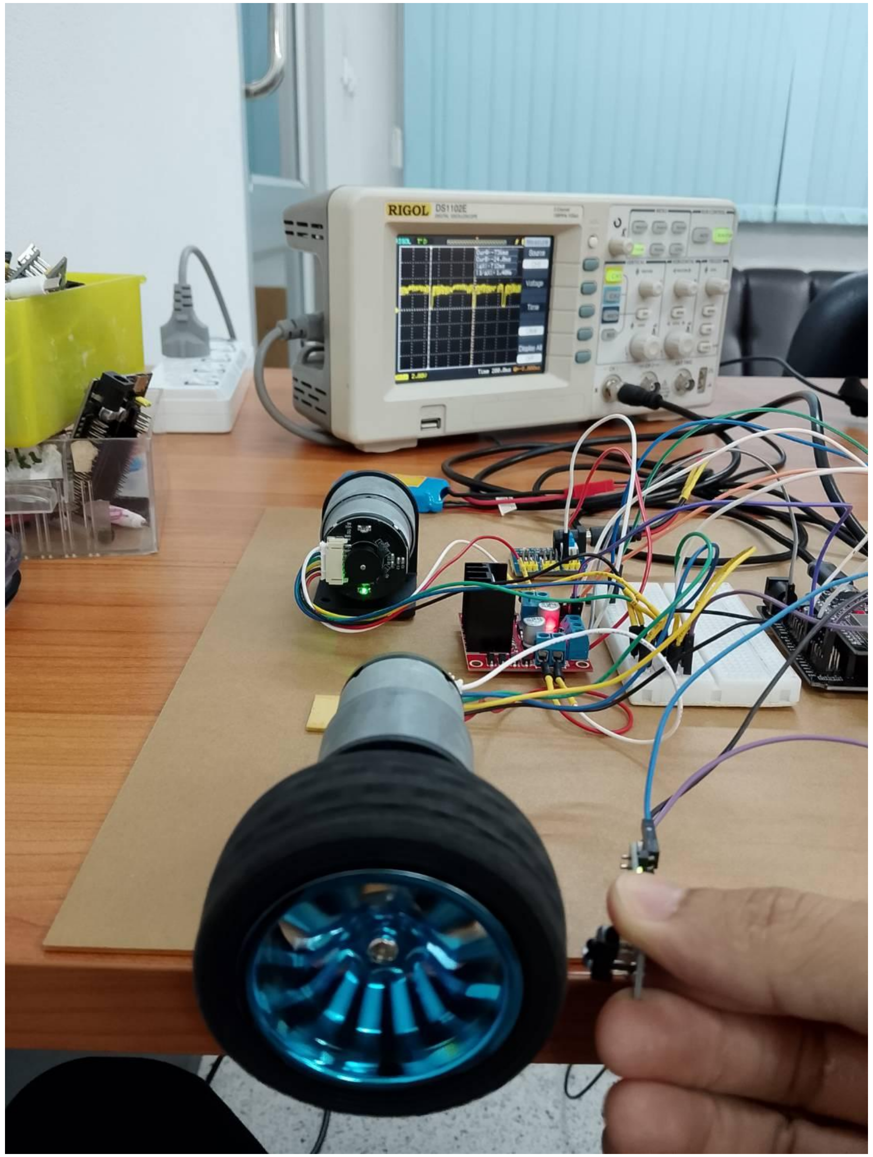
Detect white color: IR read 0V

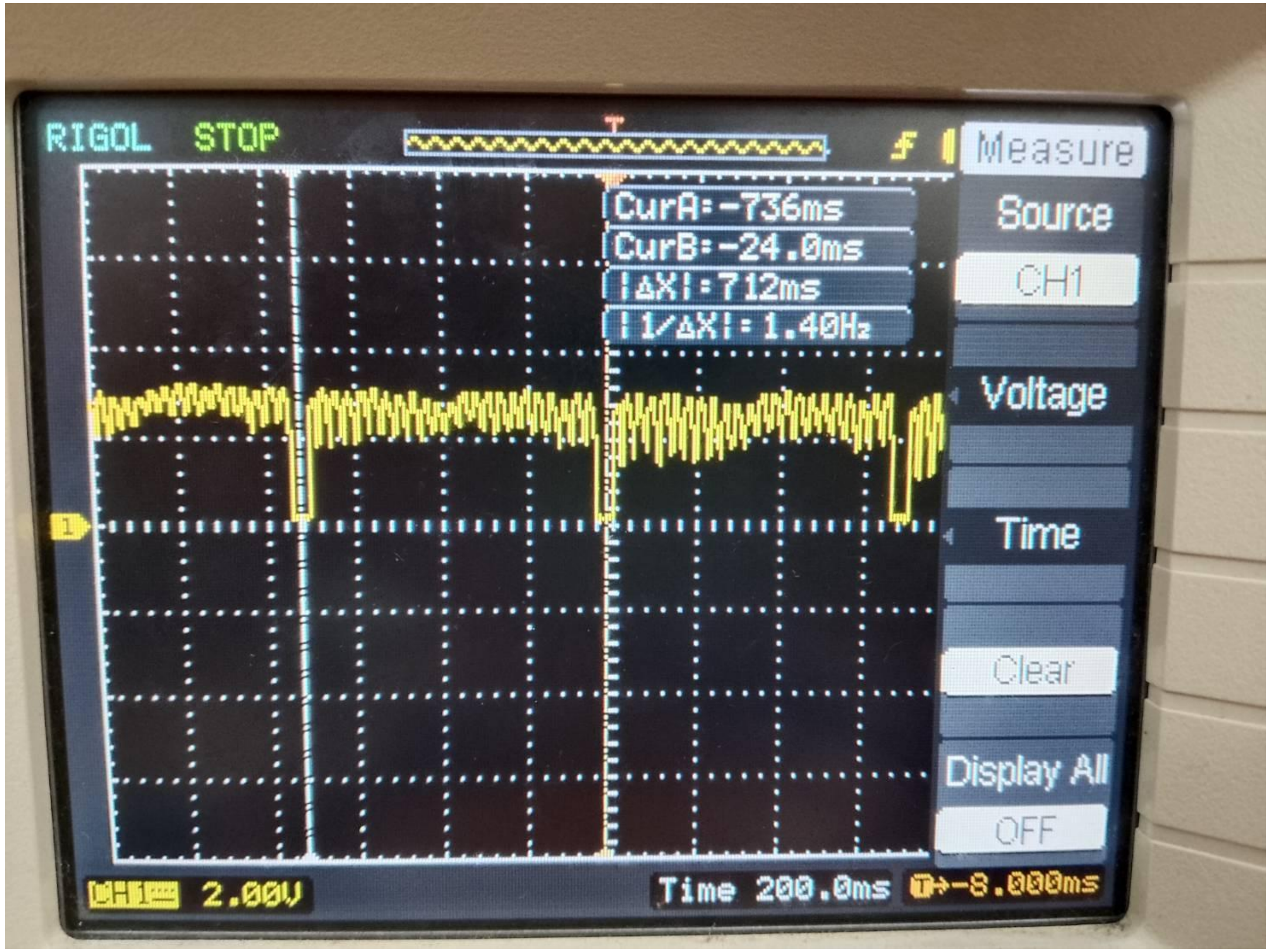


Detect black color: IR read 3V



Measure Speed or Revolution Per Minute (RPM)





CurA:-736ms
CurB:-24.0ms
|ΔX|:712ms
|1/ΔX|:1.40Hz

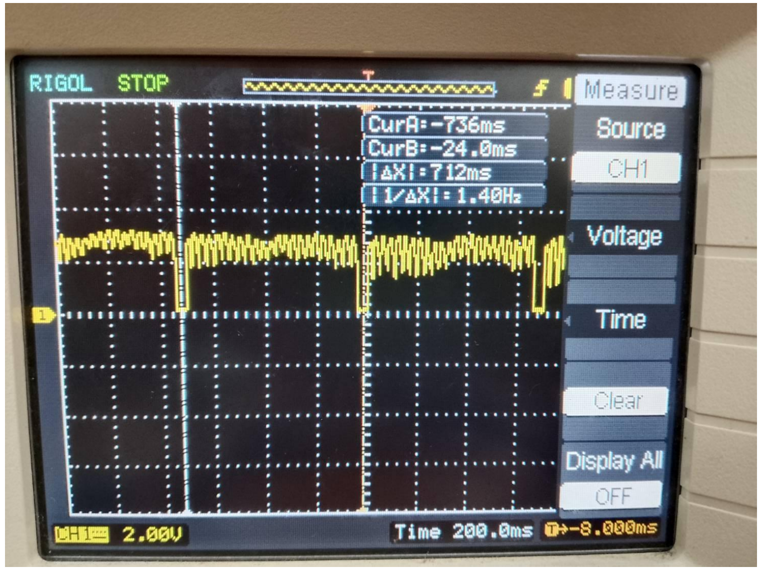
CH1 2.00V
Time 200.0ms T→-8.000ms

$$f = \frac{1}{dt} = \frac{1}{712\text{m}} = 1.4\ \text{Hz}\ (\text{รอบ/วินาที})$$

$$\text{Revolution per minute, RPM}\ \left(\frac{\text{รอบ}}{\text{นาที}}\right) = 1.4 * 60 = 84\ \text{RPM}$$

Homework3: measure revolution per minute (RPM) of your motor

PWM	PWM (%)	Periodic (T)	Frequency (Hz)	RPM
255	100			
	75			
	50			
	25			
	0			



CurA:-736ms
CurB:-24.0ms
|ΔX|:712ms
|1/ΔX|:1.40Hz

CH1 2.00V
Time 200.0ms
T→-8.000ms

Content

1. Motor encoder
    - 1.1 Motor encoder by Magnet sensor
    - 1.2 Use of External interrupt for faster reading encoder
2. **Control Movement and Position Motor**
3. **Control Speed Motor by Pulse Width Modulation (PWM)**
4. Measure speed motor by IR sensors and Oscilloscope
5. **Measure speed motor by Encoder Motor**

5. Measure RPM by Encoder Motor

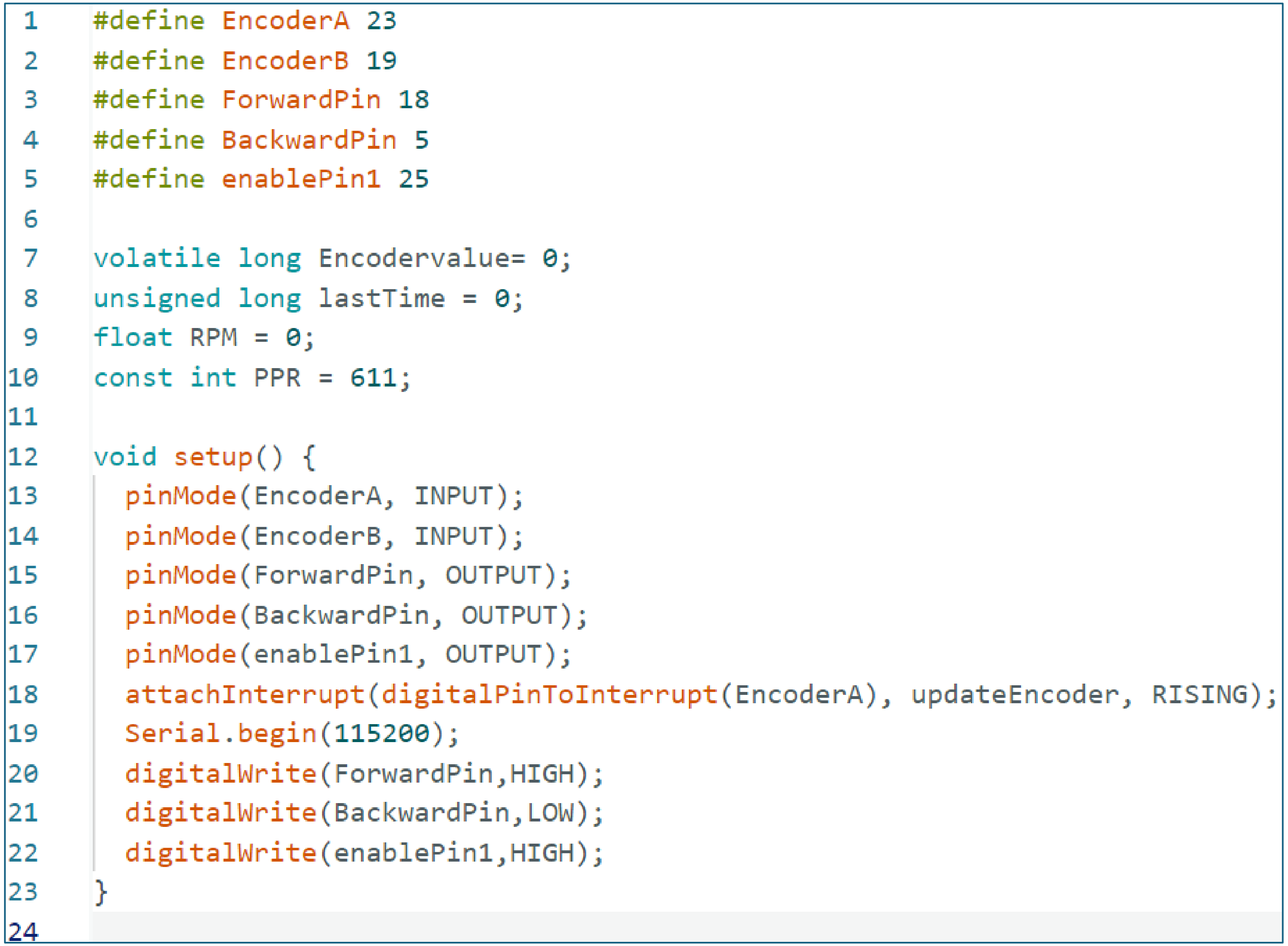
$$RPM = \frac{PulseCount}{PPR} \times 60$$

RPM (Revolution per Minute) = จำนวนรอบที่ล้อหมุนใน 1 นาที

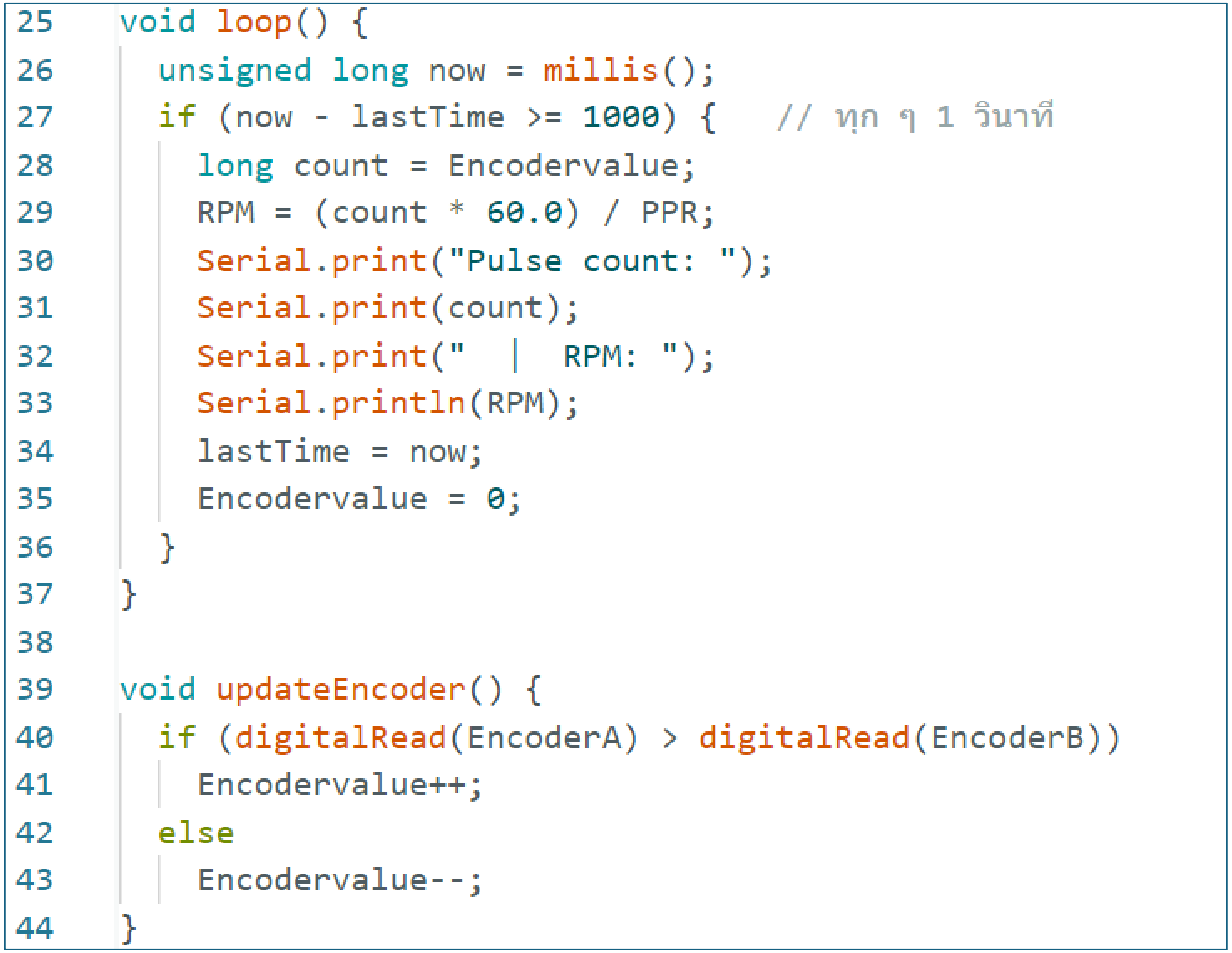
PulseCount = ค่าที่ encoderValue อ่านได้ใน 1 วินาที

PPR (Pulse per Revolution) = ค่าที่ encoderValue อ่านได้ใน 1 รอบ

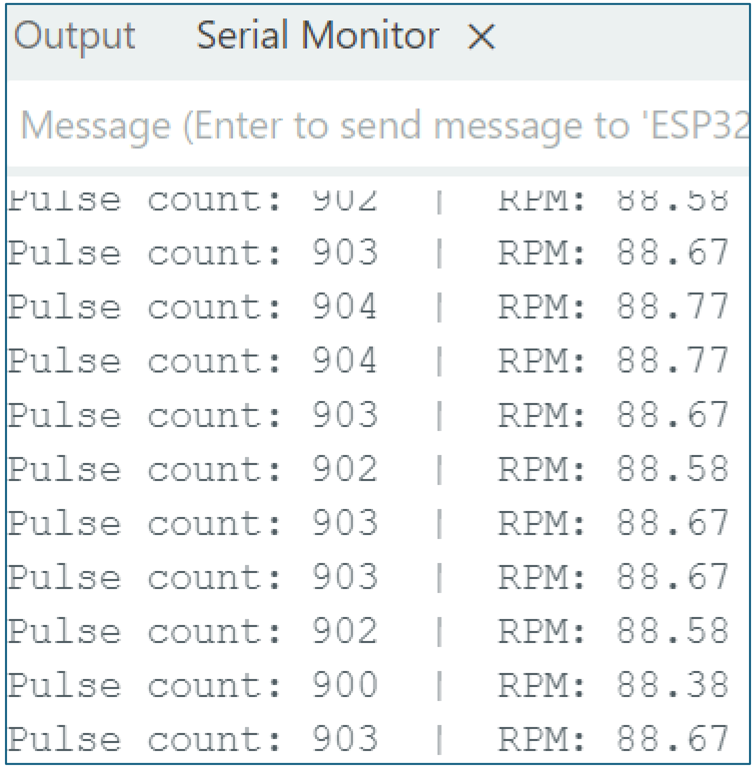
Code5.1 Measure RPM with Fixed PWM=255



```
#define EncoderA 23
#define EncoderB 19
#define ForwardPin 18
#define BackwardPin 5
#define enablePin1 25

volatile long Encodervalue= 0;
unsigned long lastTime = 0;
float RPM = 0;
const int PPR = 611;

void setup() {
  pinMode(EncoderA, INPUT);
  pinMode(EncoderB, INPUT);
  pinMode(ForwardPin, OUTPUT);
  pinMode(BackwardPin, OUTPUT);
  pinMode(enablePin1, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(EncoderA), updateEncoder, RISING);
  Serial.begin(115200);
  digitalWrite(ForwardPin,HIGH);
  digitalWrite(BackwardPin,LOW);
  digitalWrite(enablePin1,HIGH);
}

```



```
void loop() {
  unsigned long now = millis();
  if (now - lastTime >= 1000) {   // ทุก ๆ 1 วินาที
    long count = Encodervalue;
    RPM = (count * 60.0) / PPR;
    Serial.print("Pulse count: ");
    Serial.print(count);
    Serial.print("  |  RPM: ");
    Serial.println(RPM);
    lastTime = now;
    Encodervalue = 0;
  }
}

void updateEncoder() {
  if (digitalRead(EncoderA) > digitalRead(EncoderB))
    Encodervalue++;
  else
    Encodervalue--;
}
```



```
Pulse count: 902  |  RPM: 88.58
Pulse count: 903  |  RPM: 88.67
Pulse count: 904  |  RPM: 88.77
Pulse count: 904  |  RPM: 88.77
Pulse count: 903  |  RPM: 88.67
Pulse count: 902  |  RPM: 88.58
Pulse count: 903  |  RPM: 88.67
Pulse count: 903  |  RPM: 88.67
Pulse count: 902  |  RPM: 88.58
Pulse count: 900  |  RPM: 88.38
Pulse count: 903  |  RPM: 88.67
```

Homework4: Measuring each RPM from **Motor Encoder**

PWM	PWM (%)	Periodic (T)	Frequency (Hz)	RPM
255	100			
	75			
	50			
	25			
	0			

Conclusions

1. Read Encoder as Pulse Per Revolution (PPR) for follows

   1.1) Distance

   1.2) Revolution per Minute (RPM)

   1.3) Measure RPM by IR sensor

2. Control speed by PWM

3. Control Revolution

4. Control Position