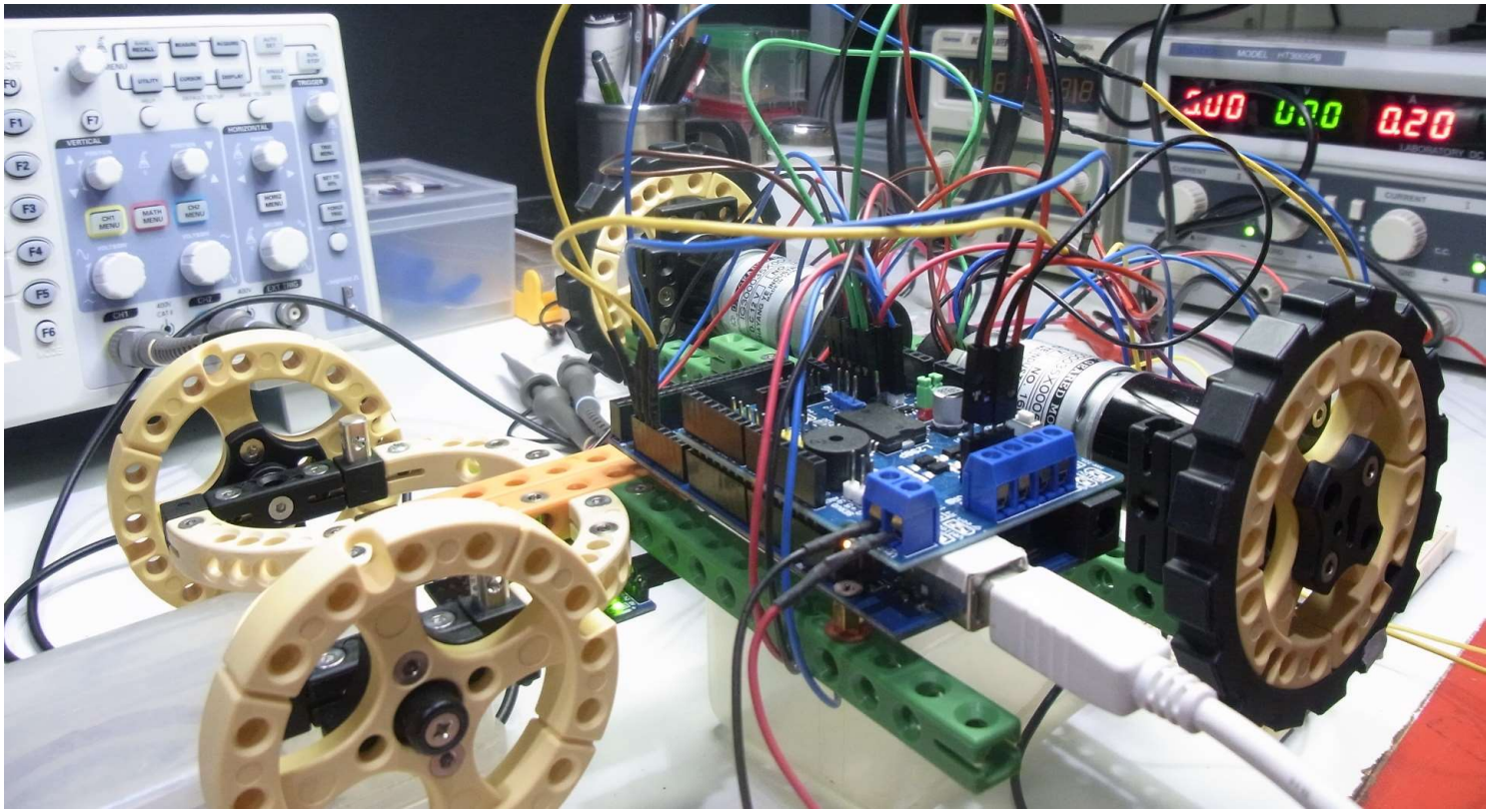


Micorocontroller for Robot Car

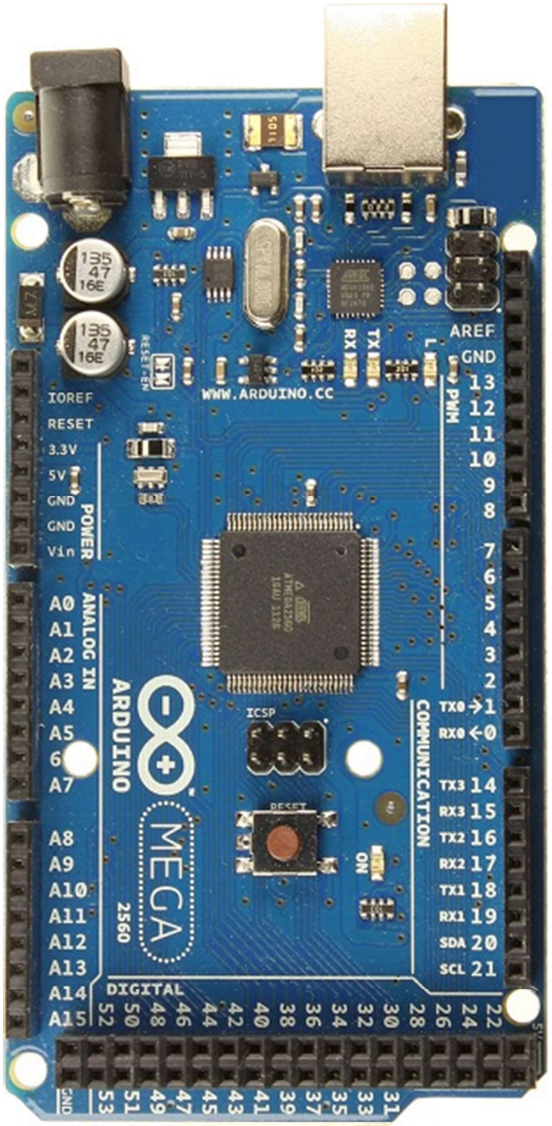
กฤตินันท์ ชอบตรง

krittinunt@gmail.com

GitHub https://github.com/krittinunt

Medium https://krittinunt.medium.com/

YouTube https://www.youtube.com/chobtrong



Arduino Mega 2560

Microcontroller: ATmega2560

Operating Voltage: 5V

Input Voltage (recommended): 7-12V

Input Voltage (limit) : 6-20V

Digital I/O Pins: 54 (of which 15 provide PWM output)

Analog Input Pins: 16

DC Current per I/O Pin: 20 mA

DC Current for 3.3V Pin: 50 mA+

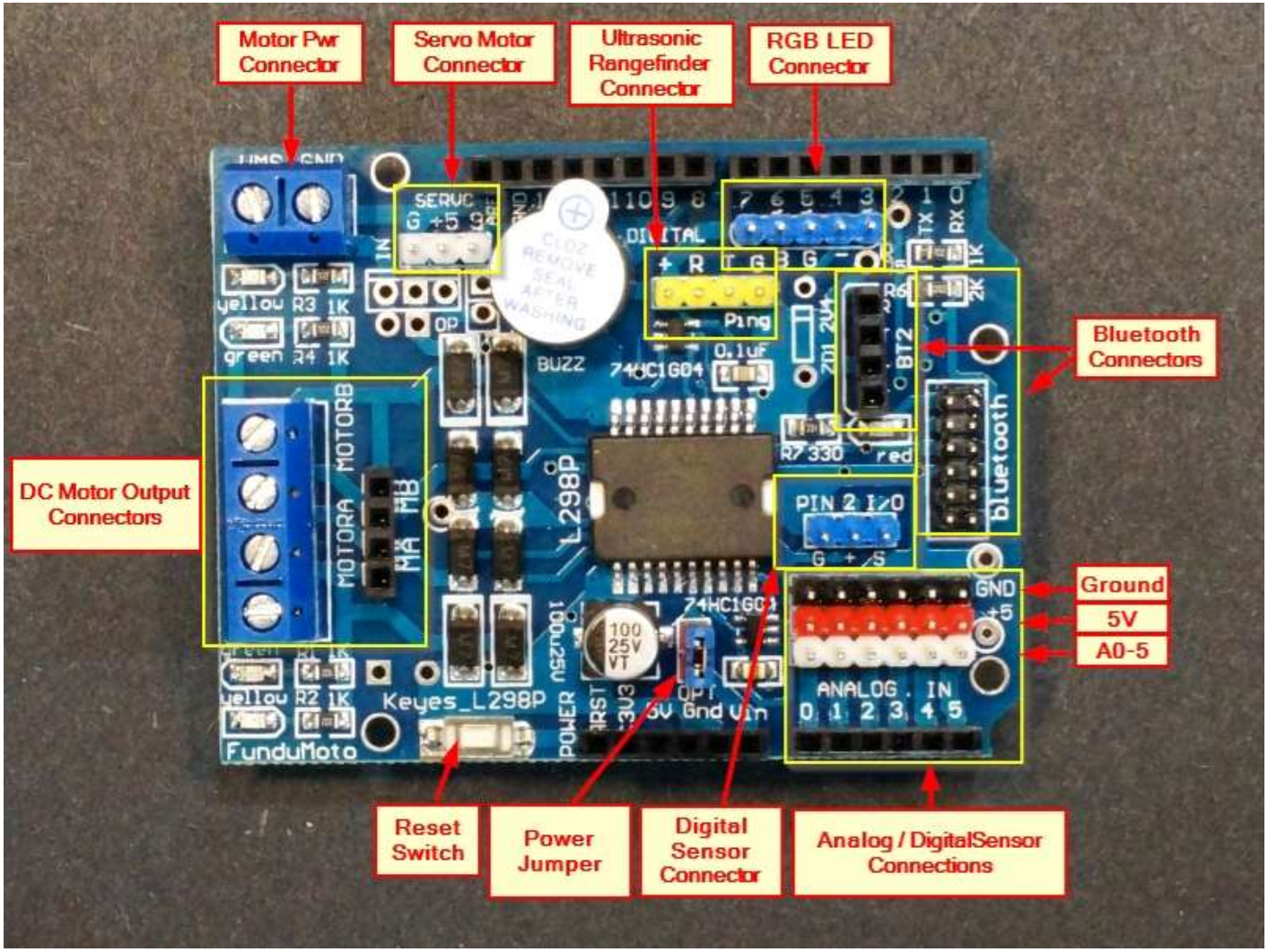
Flash Memory: 256 KB of which 8 KB used by bootloader

SRAM: 8 KB

EEPROM: 4 KB

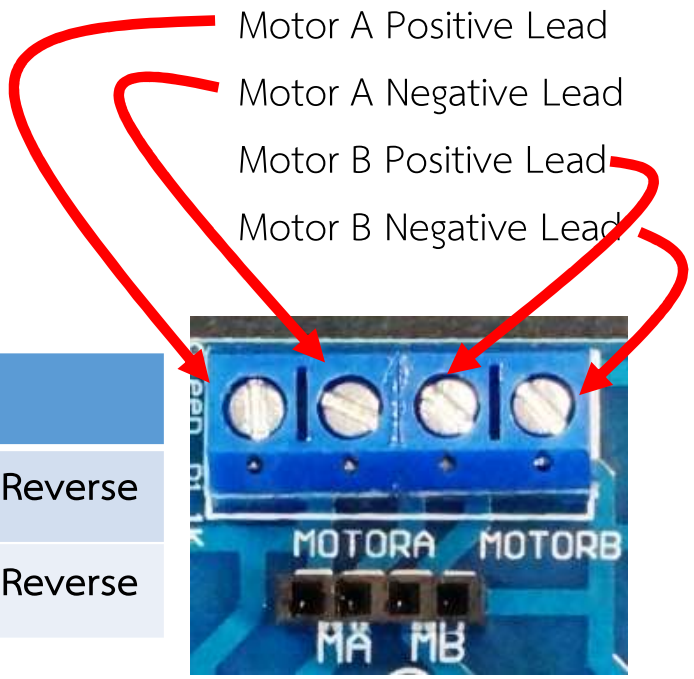
Clock Speed: 16 MHz

LED_BUILTIN: 13


Motor Pwr Connector
Servo Motor Connector
Ultrasonic Rangefinder Connector
RGB LED Connector
Bluetooth Connectors
DC Motor Output Connectors
Ground
5V
A0-5
Reset Switch
Power Jumper
Digital Sensor Connector
Analog / DigitalSensor Connections
SERVO
DIGITAL
Ping
BUZZ
74HC1G04
0.1uF
L298P
MOTORA
MOTORB
PIN 2 I/O
bluetooth
ANALOG . IN
0 1 2 3 4 5
GND
+5
Keyes_L298P
FunduMoto
POWER

VMS = Motor Vcc which must be between 4.8 and 24V.

GND = Motor Ground



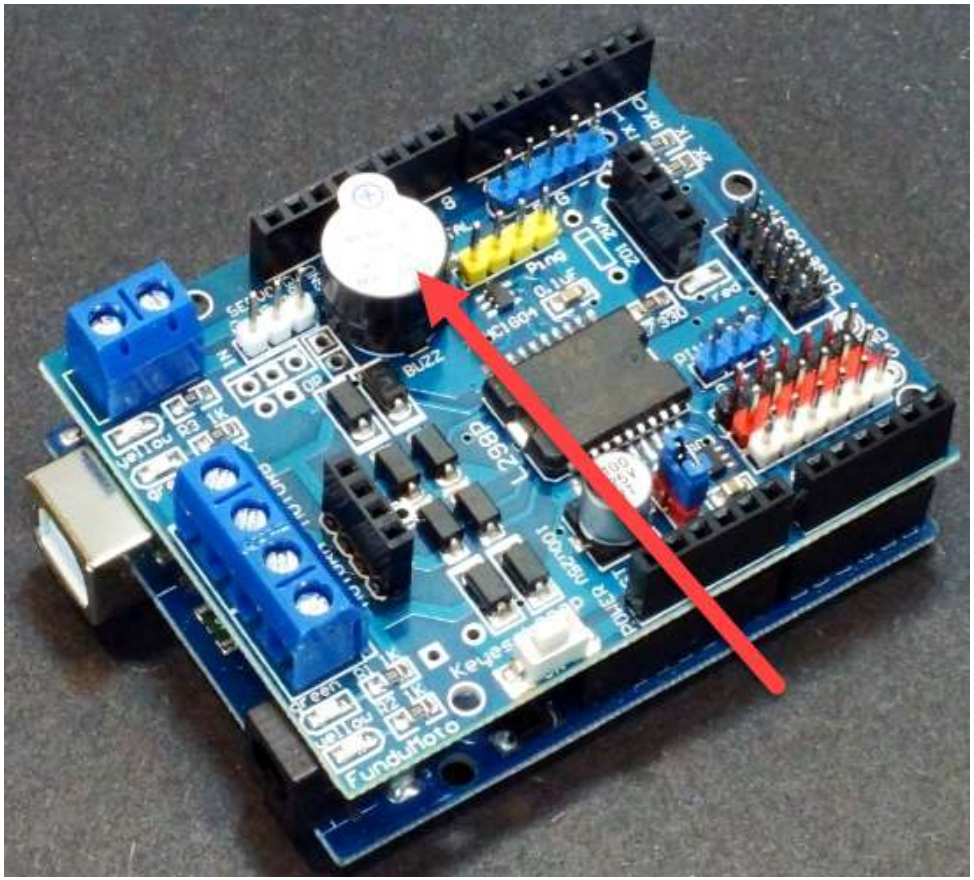
	Speed Pins	Speed Control	Direction Pins	Direction Control	
Motor A	D10	PWM 0-100	D12	HIGH = Forward	LOW = Reverse
Motor B	D11	PWM 0-100	D13	HIGH = Forward	LOW = Reverse

01_Buzzer | Arduino 1.8.13

File Edit Sketch Tools Help

01_Buzzer

```
int const BUZZER = 4;

void setup() {
  // put your setup code here, to run once:
  pinMode(BUZZER, OUTPUT);

  // beep beep
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);
  delay(50);
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

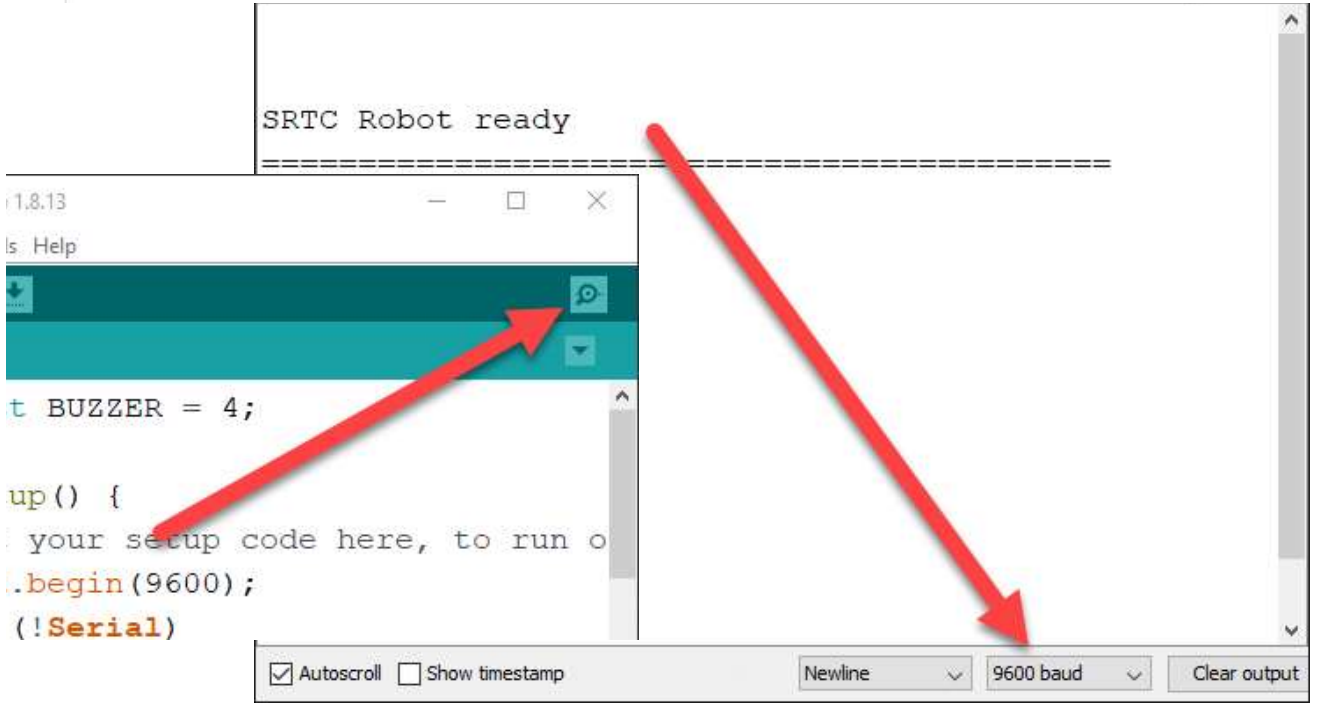


02_Serial | Arduino 1.8.13

File Edit Sketch Tools Help

02_Serial

```ino
int const BUZZER = 4;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial)
  {
    ;
  }

  // digital pin mode
  pinMode(BUZZER, OUTPUT);

  // beep beep
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);
  delay(50);
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);

  // Print to serial
  Serial.println();
  Serial.println();
  Serial.println("SRTC Robot ready");
  Serial.println("==========================================");
}

void loop() {
  // put your main code here, to run repeatedly:

}
```




AO
DO
GND
VCC


开关指示
电源指示

03_Read_IR_sensor

```ino
int const BUZZER = 4;

int const IR_SENSOR_C = 1;

float ir_sensor_c_value = 0.0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial)
  {
    ;
  }

  // digital pin mode
  pinMode(BUZZER, OUTPUT);

  // beep beep
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);
  delay(50);
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);

  // Print to serial
  Serial.println();
  Serial.println();
  Serial.println("SRTC Robot ready");
  Serial.println("=========================================");
}

void loop() {
  // put your main code here, to run repeatedly:

  ir_sensor_c_value = analogRead(IR_SENSOR_C);
  Serial.print("IR Sensor value : ");
  Serial.println(ir_sensor_c_value);

  delay(200);
}
```

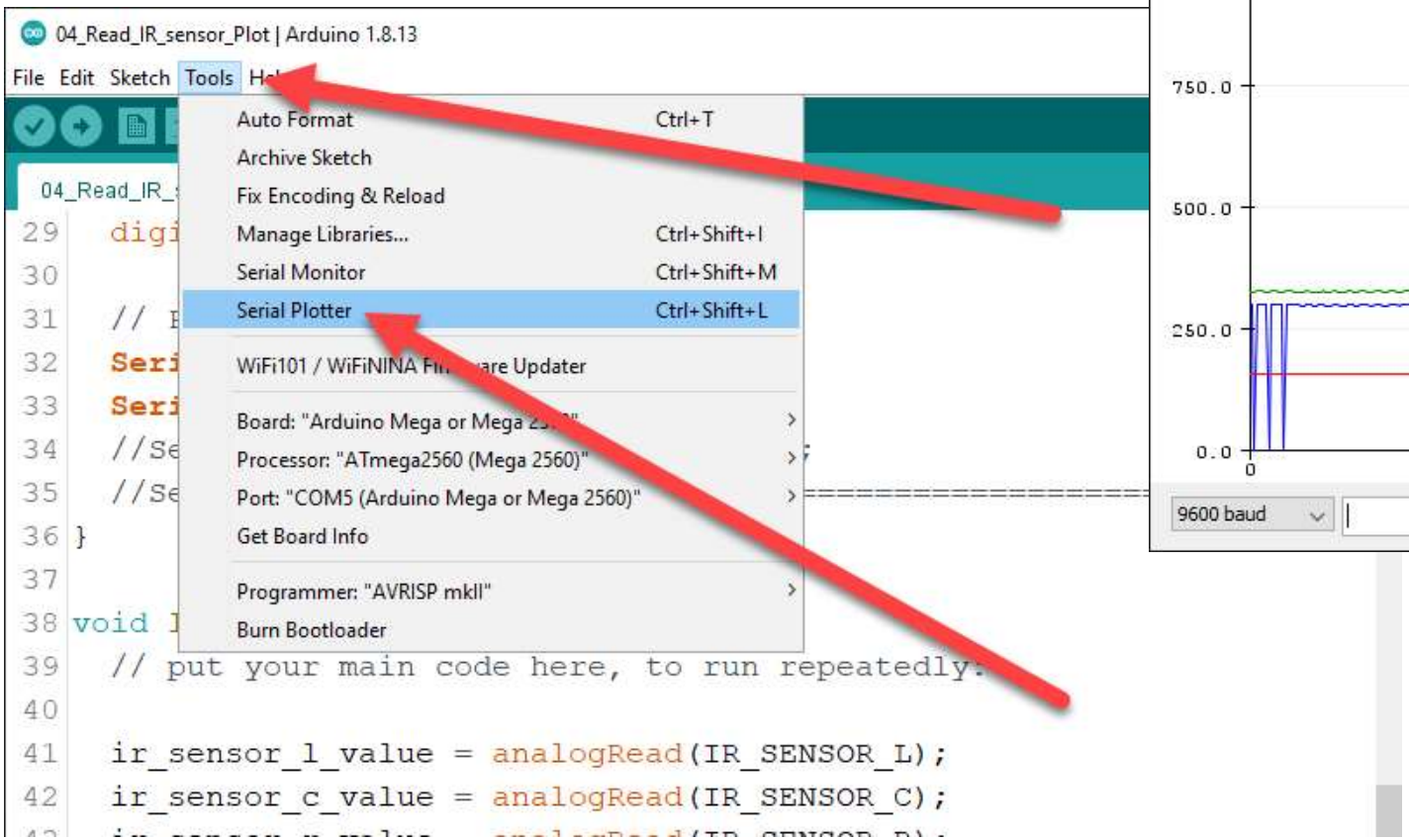
```
IR Sensor value : 147.00
IR Sensor value : 147.00
IR Sensor value : 147.00
IR Sensor value : 147.00
IR Sensor value : 147.00
IR Sensor value : 148.00
IR Sensor value : 148.00
IR Sensor value : 148.00
IR Sensor value : 148.00
```

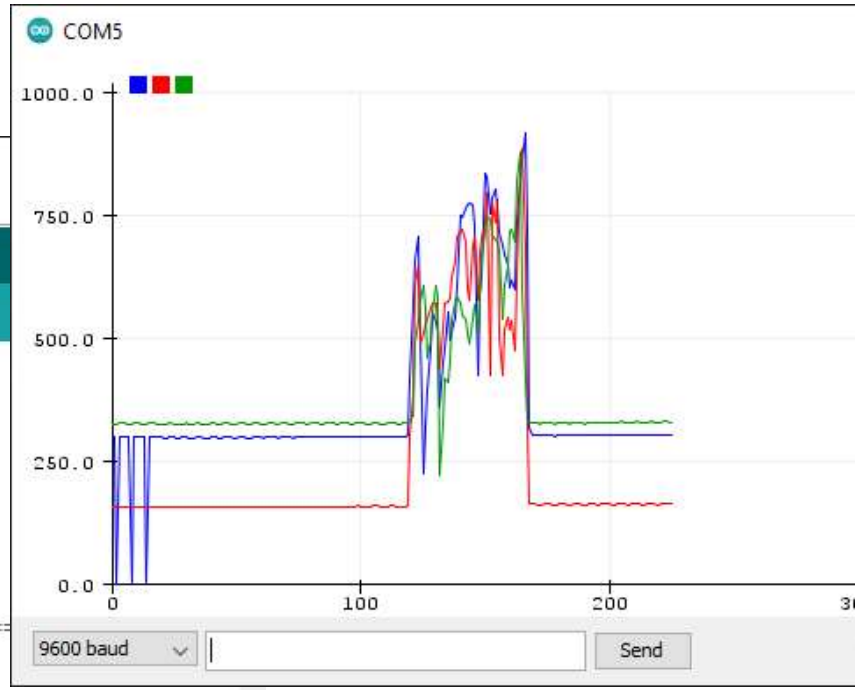
04_Read_IR_sensor_Plot

```ino
int const BUZZER = 4;

int const IR_SENSOR_L = 0;
int const IR_SENSOR_C = 1;
int const IR_SENSOR_R = 2;

float ir_sensor_l_value = 0.0;
float ir_sensor_c_value = 0.0;
float ir_sensor_r_value = 0.0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial)
  {
    ;
  }

  // digital pin mode
  pinMode(BUZZER, OUTPUT);

  // beep beep
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);
  delay(50);
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);

  // Print to serial
  Serial.println();
  Serial.println();
  //Serial.println("SRTC Robot ready");
  //Serial.println("============================
}

void loop() {
  // put your main code here, to run repeatedly:

  ir_sensor_l_value = analogRead(IR_SENSOR_L);
  ir_sensor_c_value = analogRead(IR_SENSOR_C);
  ir_sensor_r_value = analogRead(IR_SENSOR_R);

  //Serial.print("IR Sensor value : ");
  Serial.print(ir_sensor_l_value);
  Serial.print(',');
  Serial.print(ir_sensor_c_value);
  Serial.print(',');
  Serial.println(ir_sensor_r_value);
```

```
51
52    delay(200);
53  }
```

Done uploading.





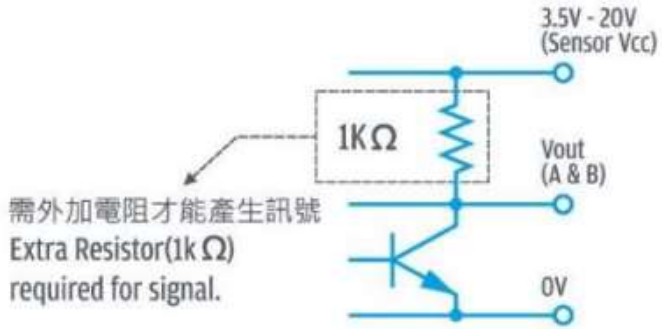
Electrical Characteristics

規格特性 Characteristics	代號 Symbol	測試條件 Test conditions	極小 Min.	基準 Ref.	最大 Max.	單位 Units
輸入電壓 Supply voltage	Vcc	- -	3.5	-	20	V
輸出飽和電壓 Output saturation voltage	Vce (sat)	Vcc = 14V ; IC = 20mA	-	300	700	mV
輸出漏電電流 Output leakage current	Icex	Vcc = 14V ; Vcc = 14V	-	< 0.1	10	μA
輸入電流 Supply current	Ice	Vcc = 20V Output open	-	5	10	mA
輸出上升時間 Output rise time	tr	Vcc = 14V ; R_L = 820 Ω ; C_L = 20pF	-	0.3	1.5	μS
輸出下降時間 Output fall time	tr	Vcc = 14V ; R_L = 820 Ω ; C_L = 20pF	-	0.3	1.5	μS

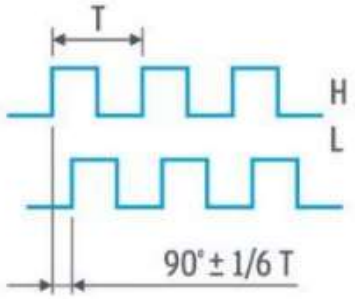
- 使用相對溼度：20%~85%RH
  Operating relative humidity
- 使用度範圍：-10°C~+60°C
  Operating temperature range

Two Channel Encoder Connections	
1. Black	: - Motor
2. Red	: + Motor
3. Brown	: Hall Sensor Vcc
4. Green	: Hall Sensor GND
5. Blue	: Hall Sensor A Vout
6. Purple	: Hall Sensor B Vout

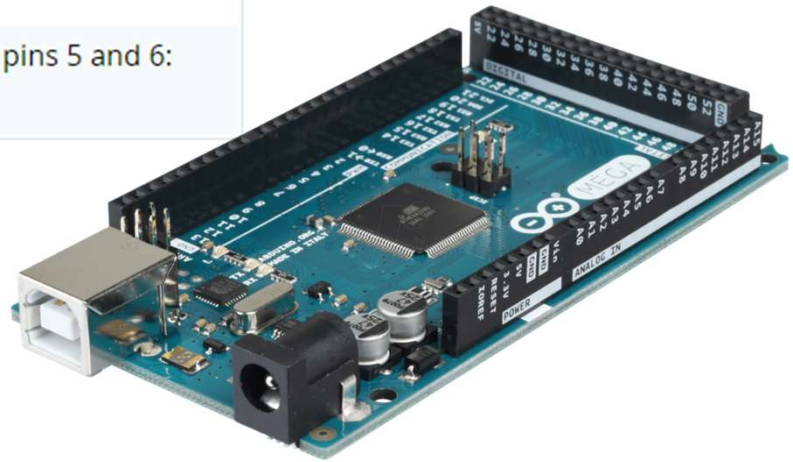
輸出電路
Output circuit

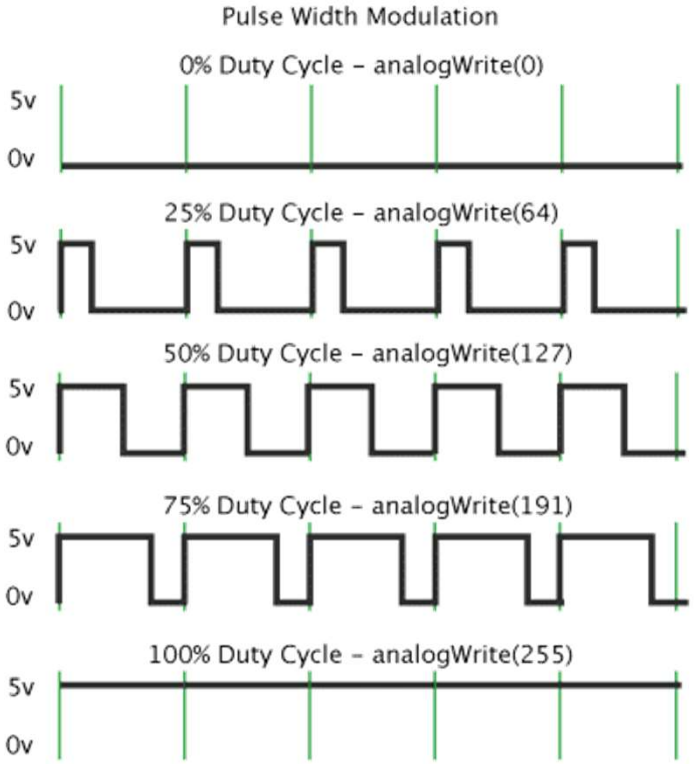


輸出波形
Output wave



BOARD	PWM PINS	PWM FREQUENCY
Uno, Nano, Mini	3, 5, 6, 9, 10, 11	490 Hz (pins 5 and 6: 980 Hz)
Mega	2 - 13, 44 - 46	490 Hz (pins 4 and 13: 980 Hz)
Leonardo, Micro, Yún	3, 5, 6, 9, 10, 11, 13	490 Hz (pins 3 and 11: 980 Hz)
Uno WiFi Rev2, Nano Every	3, 5, 6, 9, 10	976 Hz
MKR boards *	0 - 8, 10, A3, A4	732 Hz
MKR1000 WiFi *	0 - 8, 10, 11, A3, A4	732 Hz
Zero *	3 - 13, A0, A1	732 Hz
Nano 33 IoT *	2, 3, 5, 6, 9 - 12, A2, A3, A5	732 Hz
Nano 33 BLE/BLE Sense	1 - 13, A0 - A7	500 Hz
Due **	2-13	1000 Hz
101	3, 5, 6, 9	pins 3 and 9: 490 Hz, pins 5 and 6: 980 Hz





$$V_{DC} = \frac{VCC \times Duty\ Cycle}{100}$$

$$V_{DC} = \frac{12 \times 50}{100}$$

$$V_{DC} = 6\ Volts$$

```
1126  TCCR0B = (TCCR0B & 0xF8) | 0x04;
1127  //TCCR2B = (TCCR2B & 0xF8) | 0x06;    // set up timer 2 (pin 9, 10) to 122.5Hz
1128  TCCR4B = (TCCR4B & 0xF8) | 0x04;    // set up timer 4 (pin 6, 7, 8) to 112.5Hz
1129  TCCR5B = (TCCR5B & 0xF8) | 0x04;    // set up timer 5 (pin 44, 45, 46) to 112.5Hz
1130  Timer3.initialize(20000);           // set up timer 3 to 20mSec (50Hz)
1131  Timer3.attachInterrupt(timer3Isr);  // enable timer 3 INT
1132
```

05_Motor_Control

```
int const BUZZER = 4;

int const IR_SENSOR_L = 0;
int const IR_SENSOR_C = 1;
int const IR_SENSOR_R = 2;

// EN -> speed, Direction control -> HIGH =
// motor A
int const MOTOR_A_EN = 10;
int const MOTOR_A_DIR = 12;

// motor B
int const MOTOR_B_EN = 11;
int const MOTOR_B_DIR = 13;

float ir_sensor_l_value = 0.0;
float ir_sensor_c_value = 0.0;
float ir_sensor_r_value = 0.0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial)
  {
    ;
  }

  // digital pin mode
  pinMode(BUZZER, OUTPUT);

  pinMode(MOTOR_A_EN, OUTPUT);
  pinMode(MOTOR_A_DIR, OUTPUT);

  // config digital pin
  digitalWrite(MOTOR_A_DIR, HIGH);
  analogWrite(MOTOR_A_EN, 0);

  // beep beep
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);
  delay(50);
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);

  // Print to serial
  //Serial.println();
  //Serial.println();
  //Serial.println("SRTC Robot ready");
```

```
  //Serial.println("================================
}

void loop() {
  // put your main code here, to run repeatedly:

  ir_sensor_l_value = analogRead(IR_SENSOR_L);
  ir_sensor_c_value = analogRead(IR_SENSOR_C);
  ir_sensor_r_value = analogRead(IR_SENSOR_R);

  //Serial.print("IR Sensor value : ");
  //Serial.print(ir_sensor_l_value);
  //Serial.print(',');
  //Serial.print(ir_sensor_c_value);
  //Serial.print(',');
  //Serial.println(ir_sensor_r_value);

  // motor A Forward
  digitalWrite(MOTOR_A_DIR, HIGH);
  analogWrite(MOTOR_A_EN, 64);
  delay(5000);

  // motor A stop
  analogWrite(MOTOR_A_EN, 0);
  delay(2000);
}
```

07_Drive_Two_Motors

```
int const BUZZER = 4;

int const IR_SENSOR_L = 0;
int const IR_SENSOR_C = 1;
int const IR_SENSOR_R = 2;

// EN -> speed, Direction control -> HIGH =
// motor A
int const MOTOR_A_EN = 10;
int const MOTOR_A_DIR = 12;

// motor B
int const MOTOR_B_EN = 11;
int const MOTOR_B_DIR = 13;

float ir_sensor_l_value = 0.0;
float ir_sensor_c_value = 0.0;
float ir_sensor_r_value = 0.0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial)
  {
    ;
  }

  // digital pin mode
  pinMode(BUZZER, OUTPUT);

  pinMode(MOTOR_A_EN, OUTPUT);
  pinMode(MOTOR_A_DIR, OUTPUT);
  pinMode(MOTOR_B_EN, OUTPUT);
  pinMode(MOTOR_B_DIR, OUTPUT);

  // config digital pin
  digitalWrite(MOTOR_A_DIR, HIGH);
  analogWrite(MOTOR_A_EN, 0);
  digitalWrite(MOTOR_B_DIR, HIGH);
  analogWrite(MOTOR_B_EN, 0);

  // beep beep
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);
  delay(50);
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);

```

```
  // Print to serial
  //Serial.println();
  //Serial.println();
  //Serial.println("SRTC Robot ready");
  //Serial.println("==============================
}

void loop() {
  // put your main code here, to run repeatedly:

  ir_sensor_l_value = analogRead(IR_SENSOR_L);
  ir_sensor_c_value = analogRead(IR_SENSOR_C);
  ir_sensor_r_value = analogRead(IR_SENSOR_R);

  //Serial.print("IR Sensor value : ");
  //Serial.print(ir_sensor_l_value);
  //Serial.print(',');
  //Serial.print(ir_sensor_c_value);
  //Serial.print(',');
  //Serial.println(ir_sensor_r_value);

  // motor A Forward
  //digitalWrite(MOTOR_A_DIR, HIGH);
  //analogWrite(MOTOR_A_EN, 64);
  //delay(5000);

  // motor A stop
  //analogWrite(MOTOR_A_EN, 0);
  //delay(2000);

  // motor A Reverse
  //digitalWrite(MOTOR_A_DIR, LOW);
  //analogWrite(MOTOR_A_EN, 64);
  //delay(5000);

  // motor A stop
  //analogWrite(MOTOR_A_EN, 0);
  //delay(2000);

  //delay(200);

  // Drive two motors
  digitalWrite(MOTOR_A_DIR, HIGH);
  digitalWrite(MOTOR_B_DIR, HIGH);
  analogWrite(MOTOR_A_EN, 64);
  analogWrite(MOTOR_B_EN, 64);
  delay(5000);

  // Stop two motors
  analogWrite(MOTOR_A_EN, 0);
  analogWrite(MOTOR_B_EN, 0);
  delay(2000);
}
```

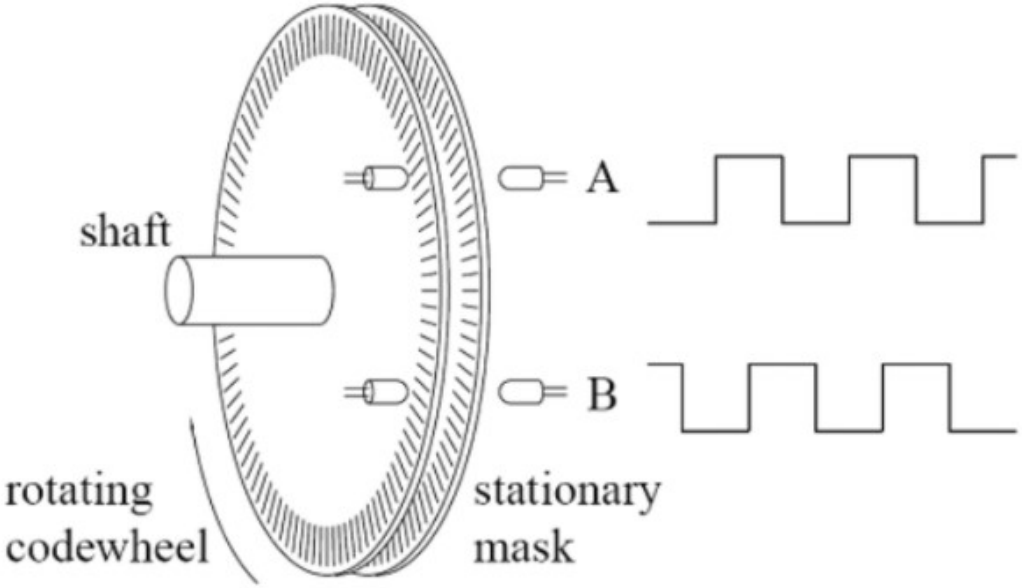
10_Car_Turn_Left_Half_Drive

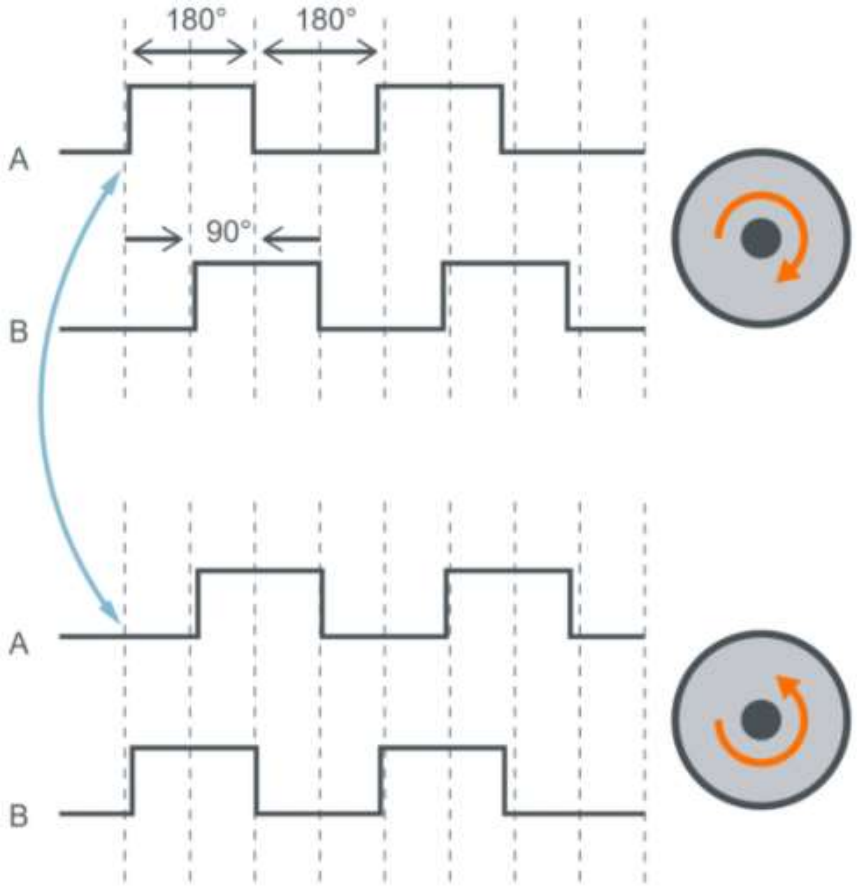
```
int const BUZZER = 4;

int const IR_SENSOR_L = 0;
int const IR_SENSOR_C = 1;
int const IR_SENSOR_R = 2;

// EN -> speed, Direction control -> HIGH = Forward,
// motor A
int const MOTOR_A_EN = 10;
int const MOTOR_A_DIR = 12;

// motor B
int const MOTOR_B_EN = 11;
int const MOTOR_B_DIR = 13;

float ir_sensor_l_value = 0.0;
float ir_sensor_c_value = 0.0;
float ir_sensor_r_value = 0.0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial)
  {
    ;
  }

  // digital pin mode
  pinMode(BUZZER, OUTPUT);

  pinMode(MOTOR_A_EN, OUTPUT);
  pinMode(MOTOR_A_DIR, OUTPUT);
  pinMode(MOTOR_B_EN, OUTPUT);
  pinMode(MOTOR_B_DIR, OUTPUT);

  // config digital pin
  digitalWrite(MOTOR_A_DIR, HIGH);
  analogWrite(MOTOR_A_EN, 0);
  digitalWrite(MOTOR_B_DIR, HIGH);
  analogWrite(MOTOR_B_EN, 0);

  // beep beep
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);
  delay(50);
  digitalWrite(BUZZER, HIGH);
  delay(70);
  digitalWrite(BUZZER, LOW);

```

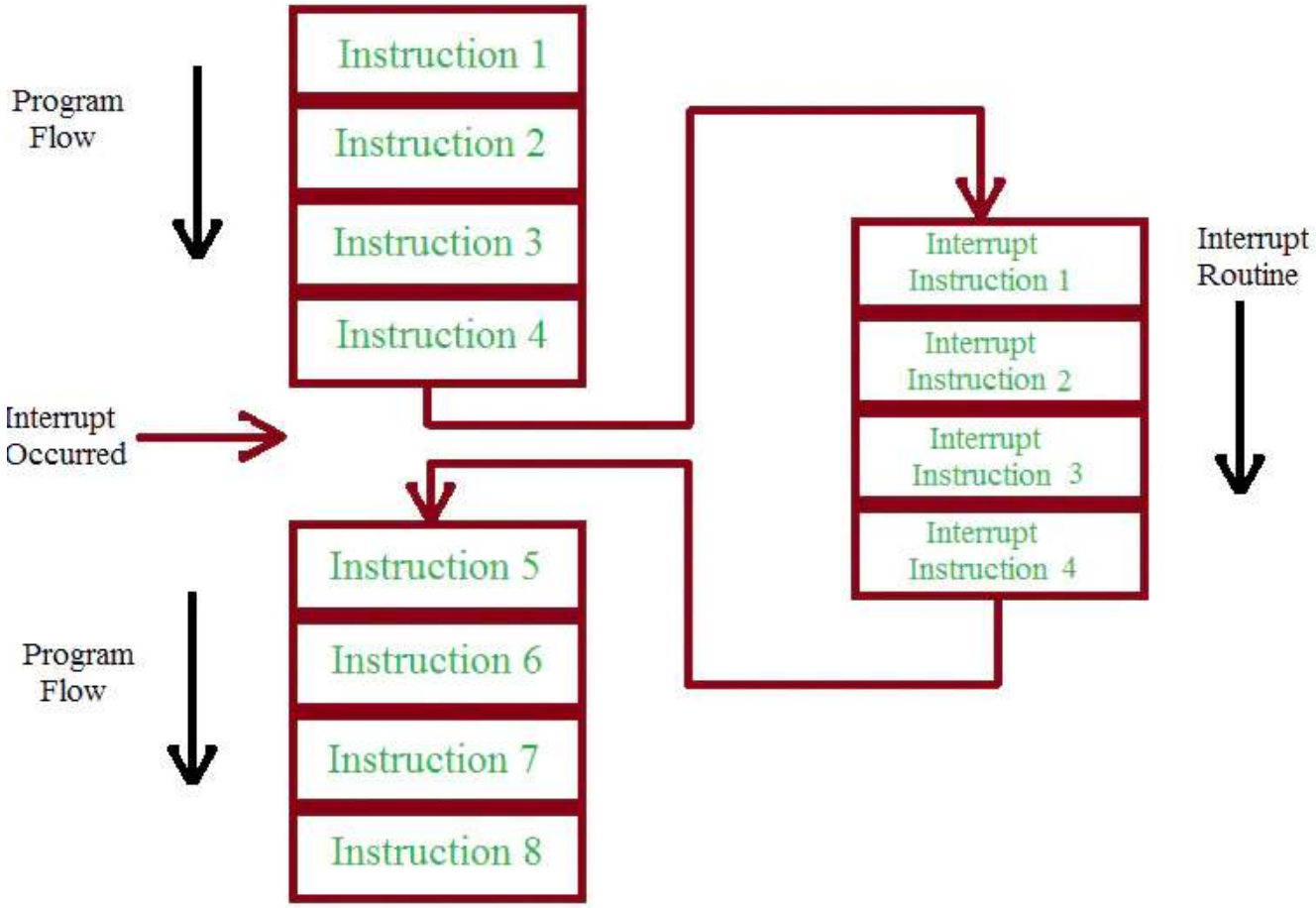
```
  // Print to serial
  //Serial.println();
  //Serial.println();
  //Serial.println("SRTC Robot ready");
  //Serial.println("============================
}

void loop() {
  // put your main code here, to run repeatedly:

  ir_sensor_l_value = analogRead(IR_SENSOR_L);
  ir_sensor_c_value = analogRead(IR_SENSOR_C);
  ir_sensor_r_value = analogRead(IR_SENSOR_R);

  //Serial.print("IR Sensor value : ");
  //Serial.print(ir_sensor_l_value);
  //Serial.print(',');
  //Serial.print(ir_sensor_c_value);
  //Serial.print(',');
  //Serial.println(ir_sensor_r_value);

  // motor A Forward
  //digitalWrite(MOTOR_A_DIR, HIGH);
  //analogWrite(MOTOR_A_EN, 64);
  //delay(5000);

  // motor A stop
  //analogWrite(MOTOR_A_EN, 0);
  //delay(2000);

  // motor A Reverse
  //digitalWrite(MOTOR_A_DIR, LOW);
  //analogWrite(MOTOR_A_EN, 64);
  //delay(5000);

  // motor A stop
  //analogWrite(MOTOR_A_EN, 0);
  //delay(2000);

  //delay(200);

  // Drive two motors
  //digitalWrite(MOTOR_A_DIR, HIGH);
  //digitalWrite(MOTOR_B_DIR, HIGH);
  //analogWrite(MOTOR_A_EN, 64);
  //analogWrite(MOTOR_B_EN, 64);
  //delay(5000);

  // Car Forward
  //digitalWrite(MOTOR_A_DIR, HIGH);
```

```
  //digitalWrite(MOTOR_B_DIR, LOW);
  //analogWrite(MOTOR_A_EN, 64);
  //analogWrite(MOTOR_B_EN, 64);
  //delay(5000);

  // Car Reverse
  //digitalWrite(MOTOR_A_DIR, LOW);
  //digitalWrite(MOTOR_B_DIR, HIGH);
  //analogWrite(MOTOR_A_EN, 64);
  //analogWrite(MOTOR_B_EN, 64);
  //delay(5000);

  // Car Turn Left Half Drive
  digitalWrite(MOTOR_B_DIR, LOW);
  analogWrite(MOTOR_A_EN, 0);
  analogWrite(MOTOR_B_EN, 64);
  delay(5000);

  // Stop two motors
  analogWrite(MOTOR_A_EN, 0);
  analogWrite(MOTOR_B_EN, 0);
  delay(2000);
}
```

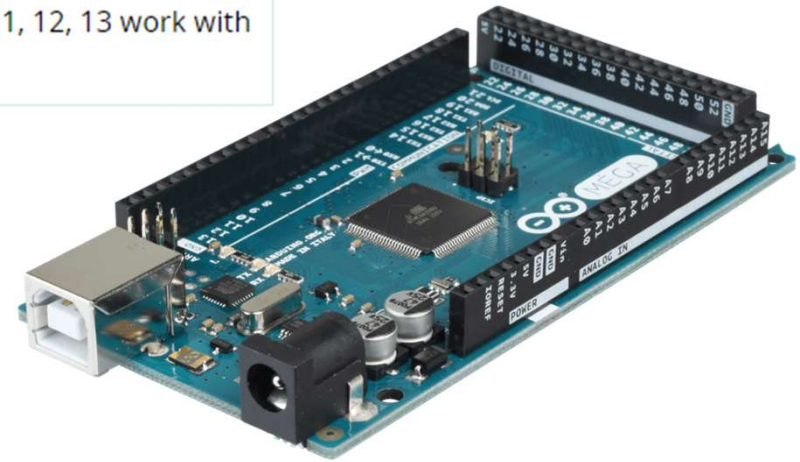
```
  // Car Turn Left Half Drive
  //digitalWrite(MOTOR_B_DIR, LOW);
  //analogWrite(MOTOR_A_EN, 0);
  //analogWrite(MOTOR_B_EN, 64);
  //delay(5000);

  // Car Turn Left Full Drive
  digitalWrite(MOTOR_A_DIR, LOW);
  digitalWrite(MOTOR_B_DIR, LOW);
  analogWrite(MOTOR_A_EN, 64);
  analogWrite(MOTOR_B_EN, 64);
  delay(5000);

  // Stop two motors
  analogWrite(MOTOR_A_EN, 0);
  analogWrite(MOTOR_B_EN, 0);
  delay(2000);
}
```


A
B
shaft
rotating
codewheel
stationary
mask


180°
180°
A
90°
B
A
B


Program
Flow
Instruction 1
Instruction 2
Instruction 3
Instruction 4
Interrupt
Occurred
Interrupt
Instruction 1
Interrupt
Instruction 2
Interrupt
Instruction 3
Interrupt
Instruction 4
Interrupt
Routine
Program
Flow
Instruction 5
Instruction 6
Instruction 7
Instruction 8

BOARD	DIGITAL PINS USABLE FOR INTERRUPTS
Uno, Nano, Mini, other 328-based	2, 3
Uno WiFi Rev.2, Nano Every	all digital pins
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7
Zero	all digital pins, except 4
MKR Family boards	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Nano 33 IoT	2, 3, 9, 10, 11, 13, 15, A5, A7
Nano 33 BLE, Nano 33 BLE Sense	all pins
Due	all digital pins
101	all digital pins (Only pins 2, 5, 7, 8, 10, 11, 12, 13 work with **CHANGE**)



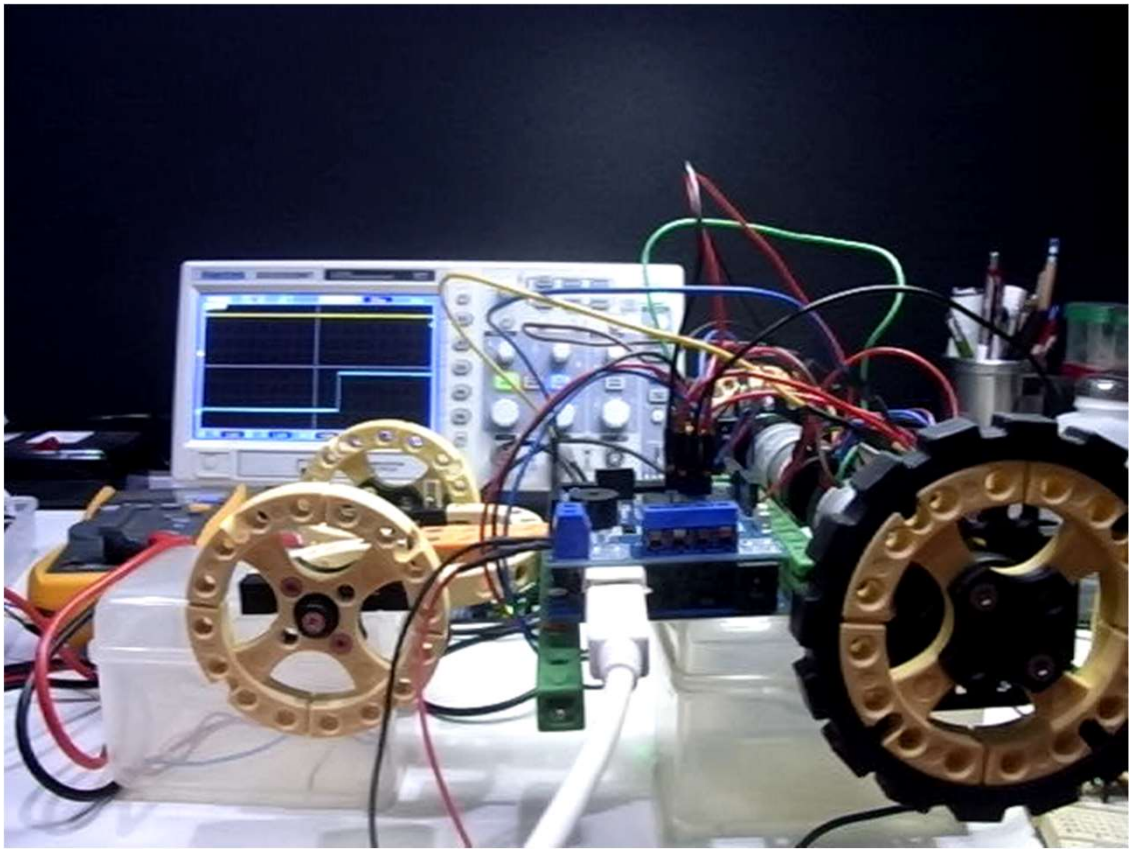
14_Motor_Encoder

```
int const BUZZER = 4;

int const IR_SENSOR_L = 0;
int const IR_SENSOR_C = 1;
int const IR_SENSOR_R = 2;

// EN -> speed, Direction control -> I
// motor A
int const MOTOR_A_EN = 10;
int const MOTOR_A_DIR = 12;
int const MOTOR_A_SENSOR_A = 18;
int const MOTOR_A_SENSOR_B = 19;

// motor B
int const MOTOR_B_EN = 11;
int const MOTOR_B_DIR = 13;

float ir_sensor_l_value = 0.0;
float ir_sensor_c_value = 0.0;
float ir_sensor_r_value = 0.0;

int i;
double motor_A_pos = 0;

void motor_A_encoder_A(void);
void motor_A_encoder_B(void);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial)
  {
    ;
  }

  // digital pin mode
  pinMode(BUZZER, OUTPUT);

  pinMode(MOTOR_A_EN, OUTPUT);
  pinMode(MOTOR_A_DIR, OUTPUT);

  pinMode(MOTOR_A_SENSOR_A, INPUT_PULLUP);
  pinMode(MOTOR_A_SENSOR_B, INPUT_PULLUP);

  // config digital pin
  digitalWrite(MOTOR_A_DIR, HIGH);
  analogWrite(MOTOR_A_EN, 0);

  digitalWrite(MOTOR_A_SENSOR_A, HIGH);
  digitalWrite(MOTOR_A_SENSOR_B, HIGH);
```

```
51
52    // Interrupt
53    attachInterrupt(digitalPinToInterrupt(MOTOR_A_SENSOR_A), motor_A_encoder_A, CHANGE);
54    attachInterrupt(digitalPinToInterrupt(MOTOR_A_SENSOR_B), motor_A_encoder_B, CHANGE);
55
56    // beep beep
57    digitalWrite(BUZZER, HIGH);
58    delay(70);
59    digitalWrite(BUZZER, LOW);
60    delay(50);
61    digitalWrite(BUZZER, HIGH);
62    delay(70);
63    digitalWrite(BUZZER, LOW);
64
65    // Print to serial
66    //Serial.println();
67    //Serial.println();
68    //Serial.println("SRTC Robot ready");
69    //Serial.println("===========================================");
70  }
71
72  void loop() {
73    // put your main code here, to run repeatedly:
74
75    motor_A_pos = 0;
```

```
  digitalWrite(MOTOR_A_DIR, HIGH);
  analogWrite(MOTOR_A_EN, 64);
  while(motor_A_pos >= -800) delay(1);
  Serial.println(motor_A_pos);
  analogWrite(MOTOR_A_EN, 00);
  digitalWrite(BUZZER, HIGH);
  delay(100);
  digitalWrite(BUZZER, LOW);
  delay(8000);
}

void motor_A_encoder_A(void)
{
  if (digitalRead(MOTOR_A_SENSOR_A) == HIGH)
  {
    if (digitalRead(MOTOR_A_SENSOR_B) == LOW) motor_A_pos++; // CW
    else motor_A_pos--; // CCW
  }
  else
  {
    if (digitalRead(MOTOR_A_SENSOR_B) == HIGH) motor_A_pos++; // CW
    else motor_A_pos--; // CCW
  }
}

```

```
void motor_A_encoder_B(void)
{
  if (digitalRead(MOTOR_A_SENSOR_B) == HIGH)
  {
    if (digitalRead(MOTOR_A_SENSOR_A) == HIGH) motor_A_pos++; // CW
    else motor_A_pos--; // CCW
  }
  else
  {
    if (digitalRead(MOTOR_A_SENSOR_A) == LOW) motor_A_pos++; // CW
    else motor_A_pos--; // CCW
  }
}
```



15_Drive_Two_Motor_Encoder

```
int const BUZZER = 4;

int const IR_SENSOR_L = 0;
int const IR_SENSOR_C = 1;
int const IR_SENSOR_R = 2;

// EN -> speed, Direction control ->
// motor A
int const MOTOR_A_EN = 10;
int const MOTOR_A_DIR = 12;
int const MOTOR_A_SENSOR_A = 18;
int const MOTOR_A_SENSOR_B = 19;

// motor B
int const MOTOR_B_EN = 11;
int const MOTOR_B_DIR = 13;
int const MOTOR_B_SENSOR_A = 20;
int const MOTOR_B_SENSOR_B = 21;

float ir_sensor_l_value = 0.0;
float ir_sensor_c_value = 0.0;
float ir_sensor_r_value = 0.0;

int i;
double motor_A_pos = 0;
double motor_B_pos = 0;

void motor_A_encoder_A(void);
void motor_A_encoder_B(void);
void motor_B_encoder_A(void);
void motor_B_encoder_B(void);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial)
  {
    ;
  }

  // digital pin mode
  pinMode(BUZZER, OUTPUT);

  pinMode(MOTOR_A_EN, OUTPUT);
  pinMode(MOTOR_A_DIR, OUTPUT);

  pinMode(MOTOR_A_SENSOR_A, INPUT_PULLUP);
  pinMode(MOTOR_A_SENSOR_B, INPUT_PULLUP);
  pinMode(MOTOR_B_SENSOR_A, INPUT_PULLUP);
  pinMode(MOTOR_B_SENSOR_B, INPUT_PULLUP);
```

```
51
52  // config digital pin
53  digitalWrite(MOTOR_A_DIR, HIGH);
54  analogWrite(MOTOR_A_EN, 0);
55  digitalWrite(MOTOR_B_DIR, HIGH);
56  analogWrite(MOTOR_B_EN, 0);
57
58  digitalWrite(MOTOR_A_SENSOR_A, HIGH);
59  digitalWrite(MOTOR_A_SENSOR_B, HIGH);
60  digitalWrite(MOTOR_B_SENSOR_A, HIGH);
61  digitalWrite(MOTOR_B_SENSOR_B, HIGH);
62
63  // Interrupt
64  attachInterrupt(digitalPinToInterrupt(MOTOR_A_SENSOR_A), motor_A_encoder_A, CHANGE);
65  attachInterrupt(digitalPinToInterrupt(MOTOR_A_SENSOR_B), motor_A_encoder_B, CHANGE);
66  attachInterrupt(digitalPinToInterrupt(MOTOR_B_SENSOR_A), motor_B_encoder_A, CHANGE);
67  attachInterrupt(digitalPinToInterrupt(MOTOR_B_SENSOR_B), motor_B_encoder_B, CHANGE);
68
69  // beep beep
70  digitalWrite(BUZZER, HIGH);
71  delay(70);
72  digitalWrite(BUZZER, LOW);
73  delay(50);
74  digitalWrite(BUZZER, HIGH);
75  delay(70);
```

```
  digitalWrite(BUZZER, LOW);

  // Print to serial
  //Serial.println();
  //Serial.println();
  //Serial.println("SRTC Robot ready");
  //Serial.println("=====================================
}

void loop() {
  // put your main code here, to run repeatedly:

  motor_A_pos = 0;
  motor_B_pos = 0;
  digitalWrite(MOTOR_A_DIR, HIGH);
  digitalWrite(MOTOR_B_DIR, LOW);
  analogWrite(MOTOR_A_EN, 64);
  analogWrite(MOTOR_B_EN, 50);
  while( (motor_A_pos >= -800) && (motor_B_pos <= 800) )
  {
    Serial.print("Motor A Pos = ");
    Serial.print(motor_A_pos);
    Serial.print(", Motor B Pos = ");
    Serial.println(motor_B_pos);
  }
```

```
  analogWrite(MOTOR_A_EN, 00);
  analogWrite(MOTOR_B_EN, 00);
  Serial.print("Motor A Pos = ");
  Serial.print(motor_A_pos);
  Serial.print(", Motor B Pos = ");
  Serial.println(motor_B_pos);
  digitalWrite(BUZZER, HIGH);
  delay(100);
  digitalWrite(BUZZER, LOW);
  delay(8000);
}

void motor_A_encoder_A(void)
{
  if (digitalRead(MOTOR_A_SENSOR_A) == HIGH)
  {
    if (digitalRead(MOTOR_A_SENSOR_B) == LOW) motor_A_pos++; // CW
    else motor_A_pos--; // CCW
  }
  else
  {
    if (digitalRead(MOTOR_A_SENSOR_B) == HIGH) motor_A_pos++; // CW
    else motor_A_pos--; // CCW
  }
}
```

```
126
127 void motor_A_encoder_B(void)
128 {
129   if (digitalRead(MOTOR_A_SENSOR_B) == HIGH)
130   {
131     if (digitalRead(MOTOR_A_SENSOR_A) == HIGH) motor_A_pos++; // CW
132     else motor_A_pos--; // CCW
133   }
134   else
135   {
136     if (digitalRead(MOTOR_A_SENSOR_A) == LOW) motor_A_pos++; // CW
137     else motor_A_pos--; // CCW
138   }
139 }
140
141 void motor_B_encoder_A(void)
142 {
143   if (digitalRead(MOTOR_B_SENSOR_A) == HIGH)
144   {
145     if (digitalRead(MOTOR_B_SENSOR_B) == LOW) motor_B_pos++; // CW
146     else motor_B_pos--; // CCW
147   }
148   else
149   {
150     if (digitalRead(MOTOR_B_SENSOR_B) == HIGH) motor_B_pos++; // CW
```

```
150     if (digitalRead(MOTOR_B_SENSOR_B) == HIGH) motor_B_pos++; // CW
151     else motor_B_pos--; // CCW
152   }
153 }
154
155 void motor_B_encoder_B(void)
156 {
157   if (digitalRead(MOTOR_B_SENSOR_B) == HIGH)
158   {
159     if (digitalRead(MOTOR_B_SENSOR_A) == HIGH) motor_B_pos++; // CW
160     else motor_B_pos--; // CCW
161   }
162   else
163   {
164     if (digitalRead(MOTOR_B_SENSOR_A) == LOW) motor_B_pos++; // CW
165     else motor_B_pos--; // CCW
166   }
167 }
```

16_Free_Run

```
85 void loop() {
86   // put your main code here, to run repeatedly:
87
88   // fw
89   motor_A_pos = 0;
90   motor_B_pos = 0;
91   digitalWrite(MOTOR_A_DIR, HIGH);
92   digitalWrite(MOTOR_B_DIR, LOW);
93   analogWrite(MOTOR_A_EN, 64);
94   analogWrite(MOTOR_B_EN, 64);
95   while(motor_A_pos >= -10000)
96   {
97     Serial.print("Motor A Pos = ");
98     Serial.print(motor_A_pos);
99     Serial.print(", Motor B Pos = ");
100    Serial.println(motor_B_pos);
101  }
102  analogWrite(MOTOR_A_EN, 00);
103  analogWrite(MOTOR_B_EN, 00);
104  delay(500);
105
106  // 1
107  motor_A_pos = 0;
108  motor_B_pos = 0;
109  digitalWrite(MOTOR_A_DIR, LOW);
110  digitalWrite(MOTOR_B_DIR, LOW);
111  analogWrite(MOTOR_A_EN, 64);
112  analogWrite(MOTOR_B_EN, 64);
113  while(motor_A_pos <= 1000)
114  {
115    Serial.print("Motor A Pos = ");
116    Serial.print(motor_A_pos);
117    Serial.print(", Motor B Pos = ");
118    Serial.println(motor_B_pos);
119  }
120  analogWrite(MOTOR_A_EN, 00);
121  analogWrite(MOTOR_B_EN, 00);
122  delay(500);
123
124  digitalWrite(BUZZER, HIGH);
125  delay(100);
126  digitalWrite(BUZZER, LOW);
127  while(true)
128  {
129    //
130  }
131 }
```