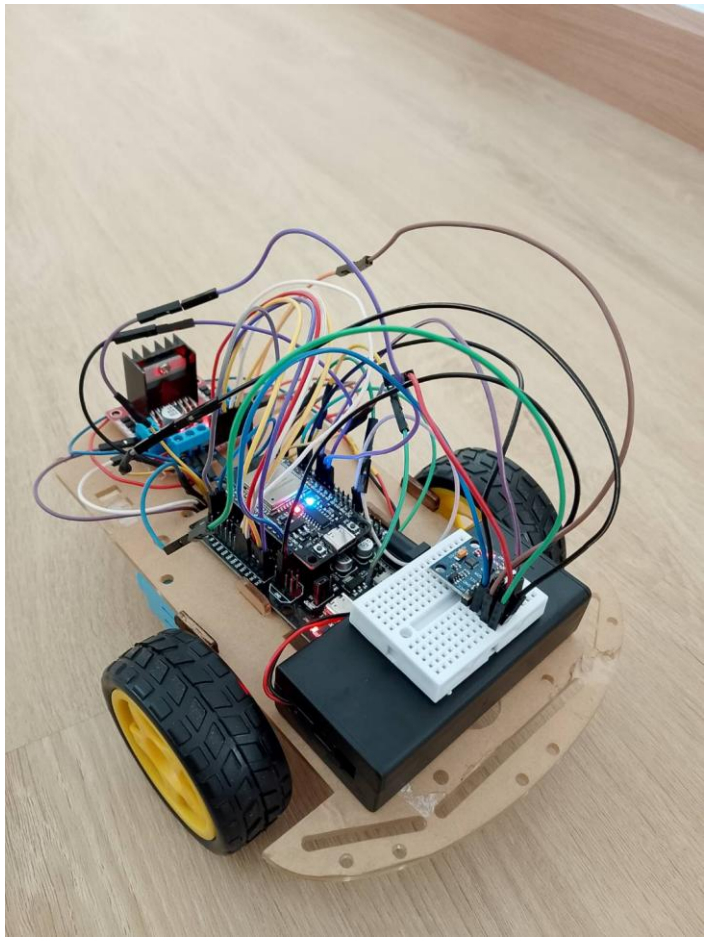
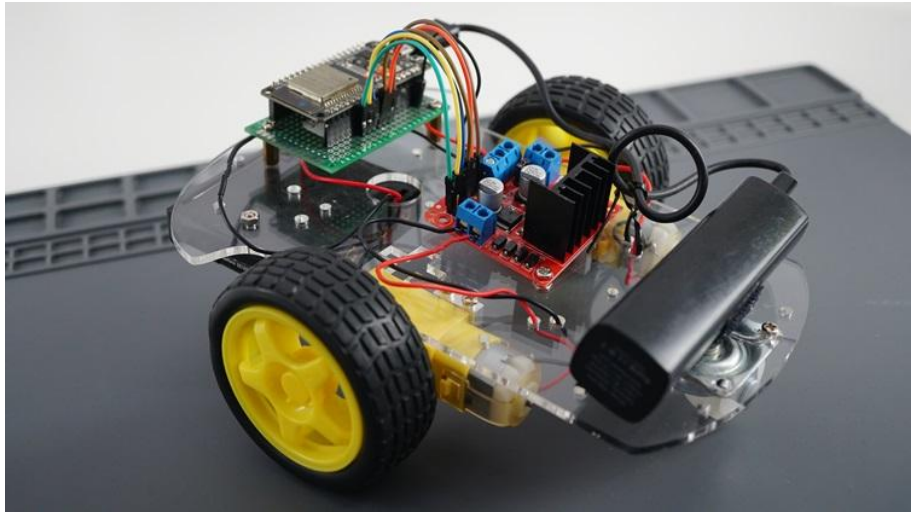




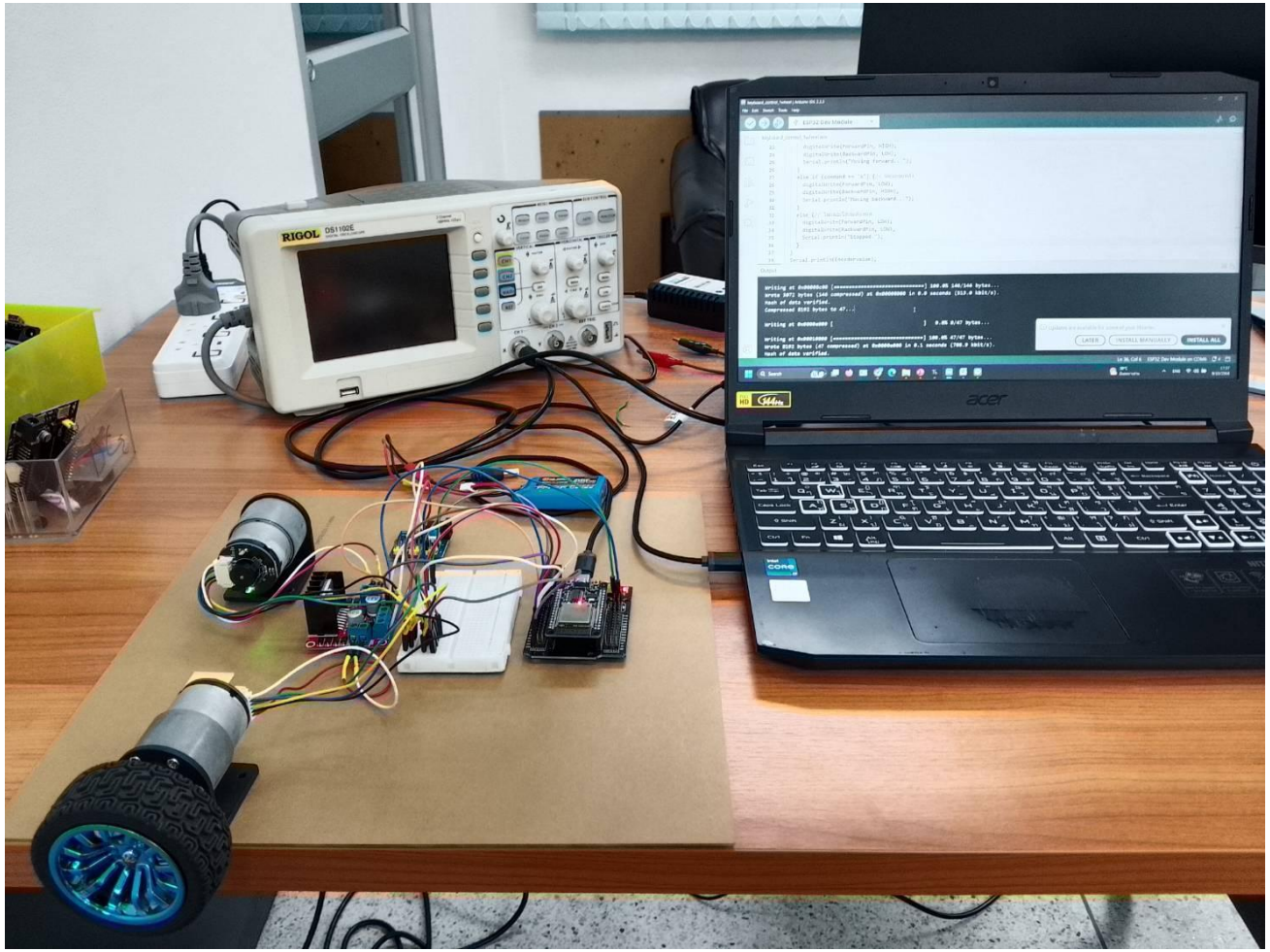
MECHATRONICS
AND AUTOMATION ENGINEERING

Lab work: Basic Robot

Week4: Robot car with keyboard and WIFI control

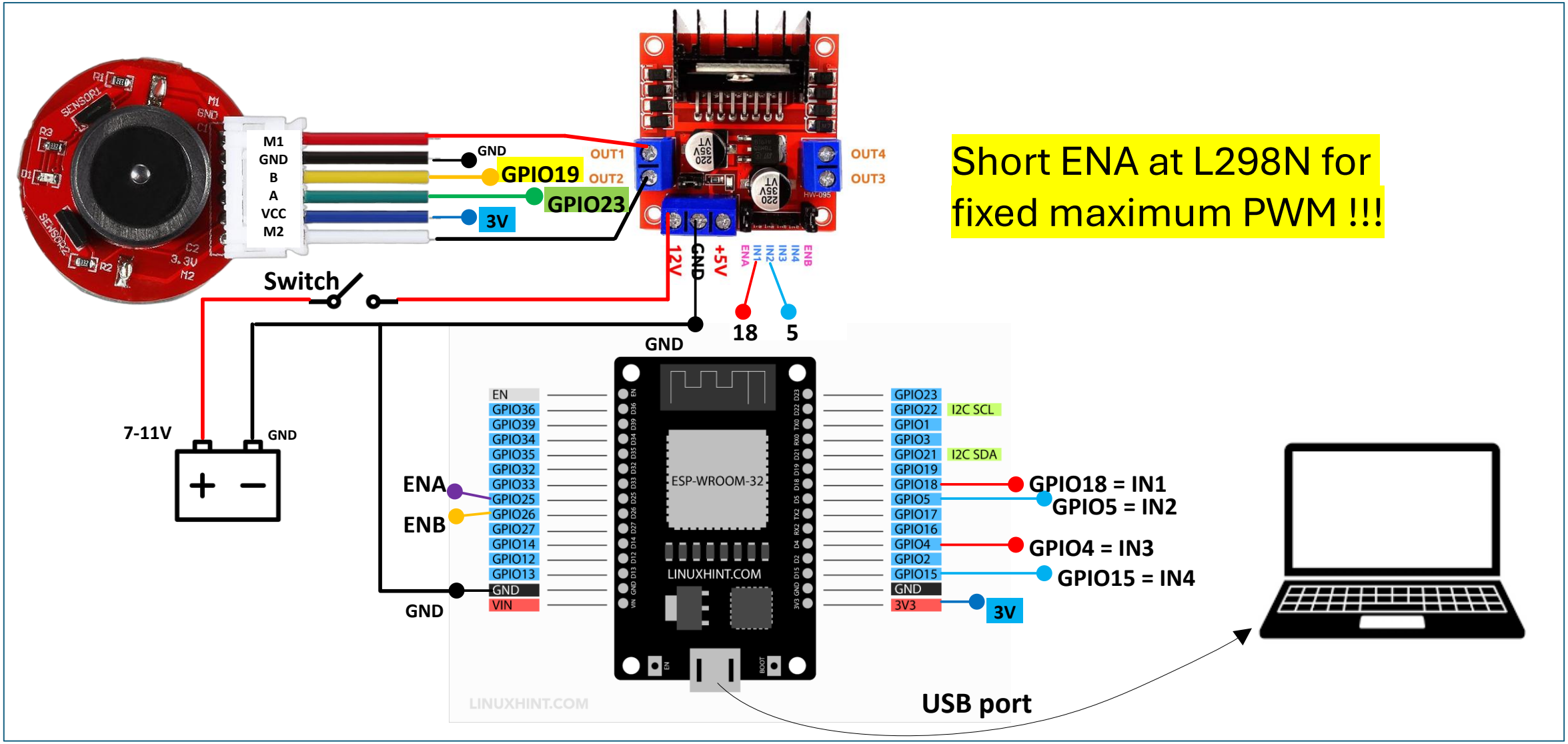
Content

Chapter1: one motor movement by basic keyboard



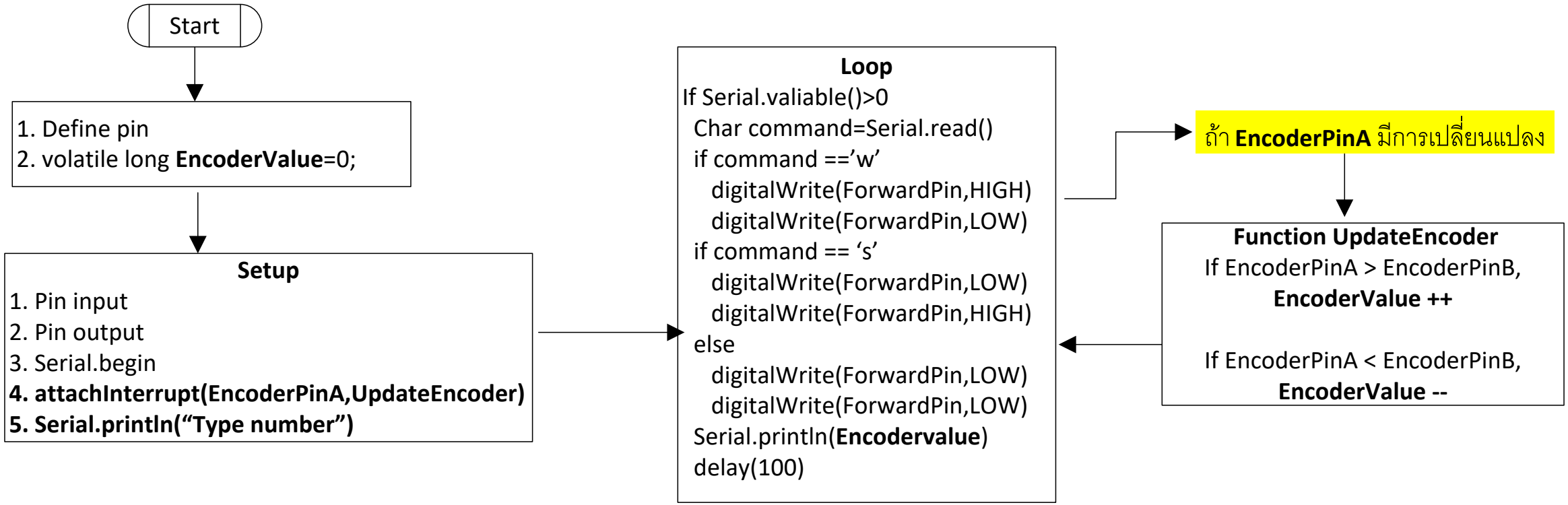
Press ‘w’ motor is forward
Press ‘s’ motor is backward

Ex1: A control movement by basic keyboard



Short ENA at L298N for fixed maximum PWM !!!

Ex1. Control forward, backward with reading the encoder



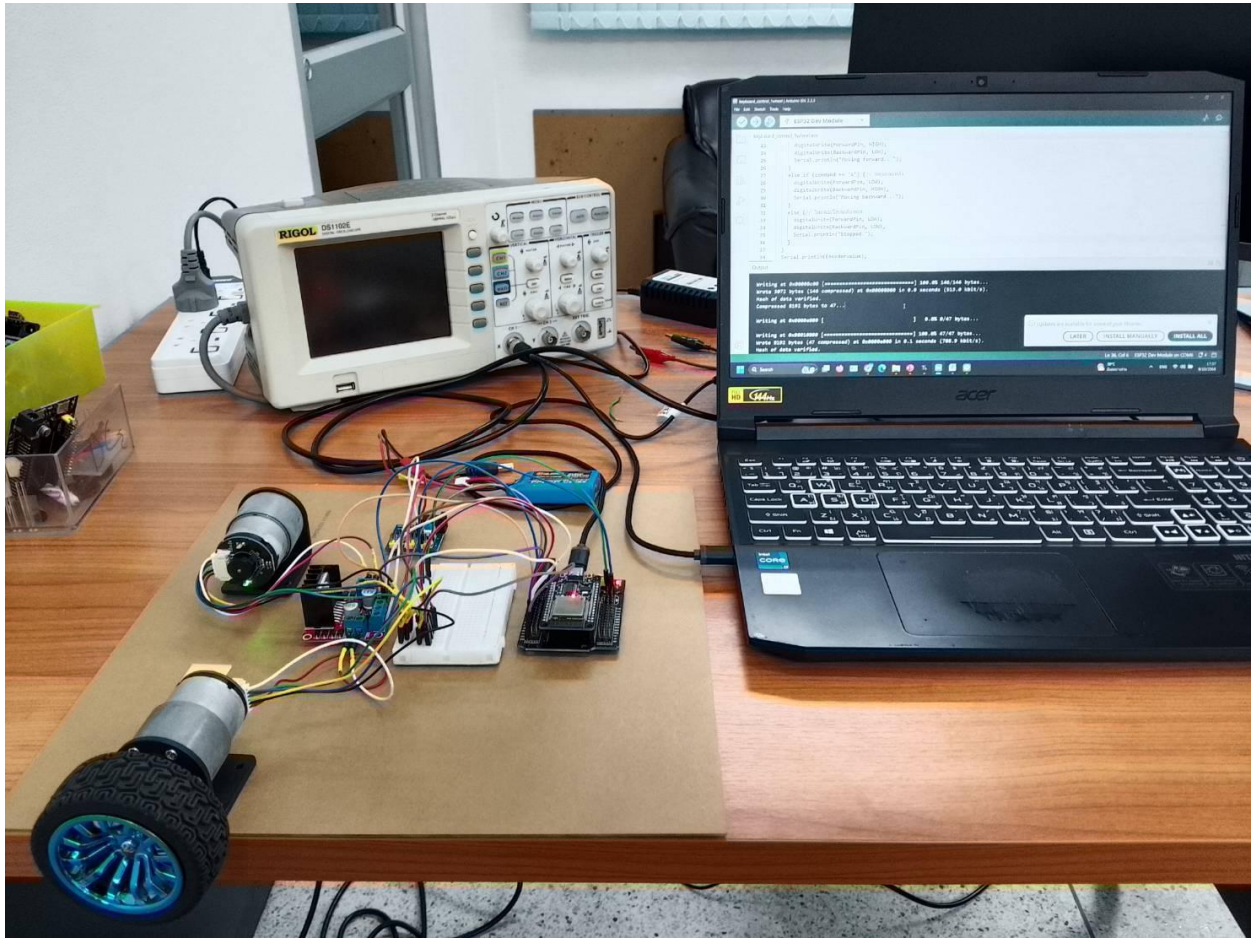
Ex1. Control movement by keyboard

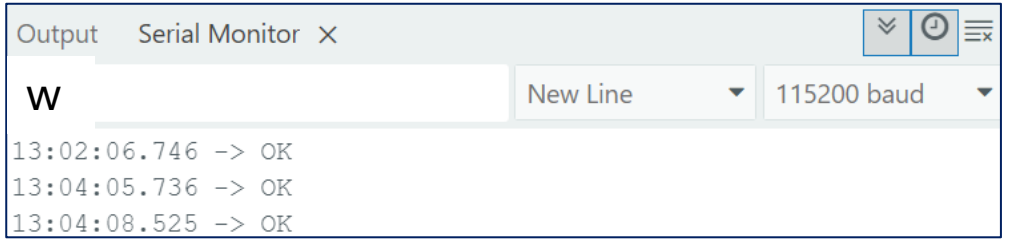
```
#define EncoderA 23
#define EncoderB 19
#define ForwardPin 18
#define BackwardPin 5
volatile long Encodervalue = 0;
void setup() {
  pinMode(EncoderA, INPUT);
  pinMode(EncoderB, INPUT);
  pinMode(ForwardPin, OUTPUT);
  pinMode(BackwardPin, OUTPUT);
  digitalWrite(ForwardPin, LOW);
  digitalWrite(BackwardPin, LOW);
  Serial.begin(115200);
  attachInterrupt(digitalPinToInterrupt(EncoderA), updateEncoder, RISING);
  Serial.println("Press 'w' to move forward, 's' to move backward, any other key to stop.");
}

```

```
18  void loop() {
19    // ตรวจสอบว่ามีข้อมูลจาก Serial เข้ามาหรือไม่
20    if (Serial.available() > 0) {
21      char command = Serial.read();  // อ่านอักษรจาก Serial
22      if (command == 'w') {// หมุนเดินหน้า
23        digitalWrite(ForwardPin, HIGH);
24        digitalWrite(BackwardPin, LOW);
25        Serial.println("Moving forward...");
26      }
27      else if (command == 's') {// หมุนถอยหลัง
28        digitalWrite(ForwardPin, LOW);
29        digitalWrite(BackwardPin, HIGH);
30        Serial.println("Moving backward...");
31      }
32      else {// ไม่กดอะไรเลยคือหยุด
33        digitalWrite(ForwardPin, LOW);
34        digitalWrite(BackwardPin, LOW);
35        Serial.println("Stopped.");
36      }
37    }
38    Serial.println(Encodervalue);
39    delay(100);
40  }
```

```
42  void updateEncoder() {
43    if (digitalRead(EncoderA) > digitalRead(EncoderB))
44      Encodervalue++;
45    else
46      Encodervalue--;
47  }
```

Summary

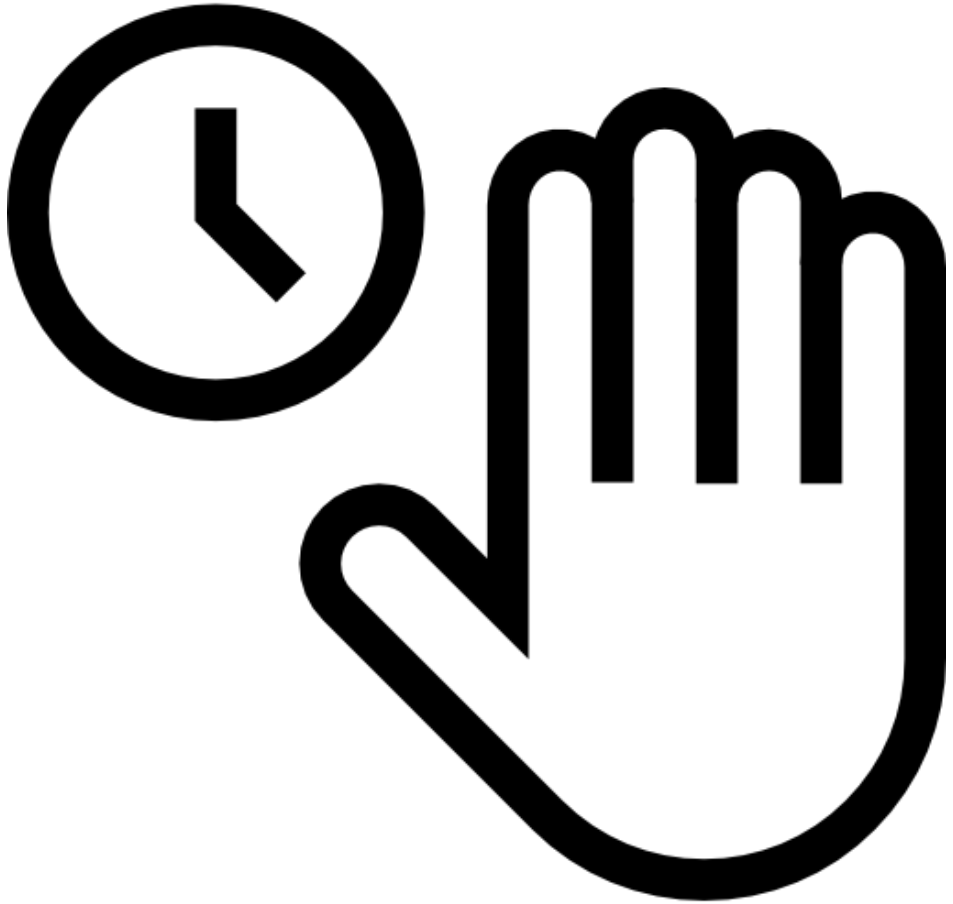




Press ‘w’ motor is forward
Press ‘s’ motor is backward

ปัญหาคือถ้าต่อกับ **servo motor** หลายตัวเพื่อทำแขนจับ ต้องพิมพ์ **class, id, degree** ด้วยเช่น **servo** ตัวที่ 1 หมุน 45 องศา**“v 1 45”**
servo ตัวที่ 2 หมุน 60 องศา **“v 2 60”**
จะเขียนรับ **input** จาก **serial port** ยากขึ้นมาก

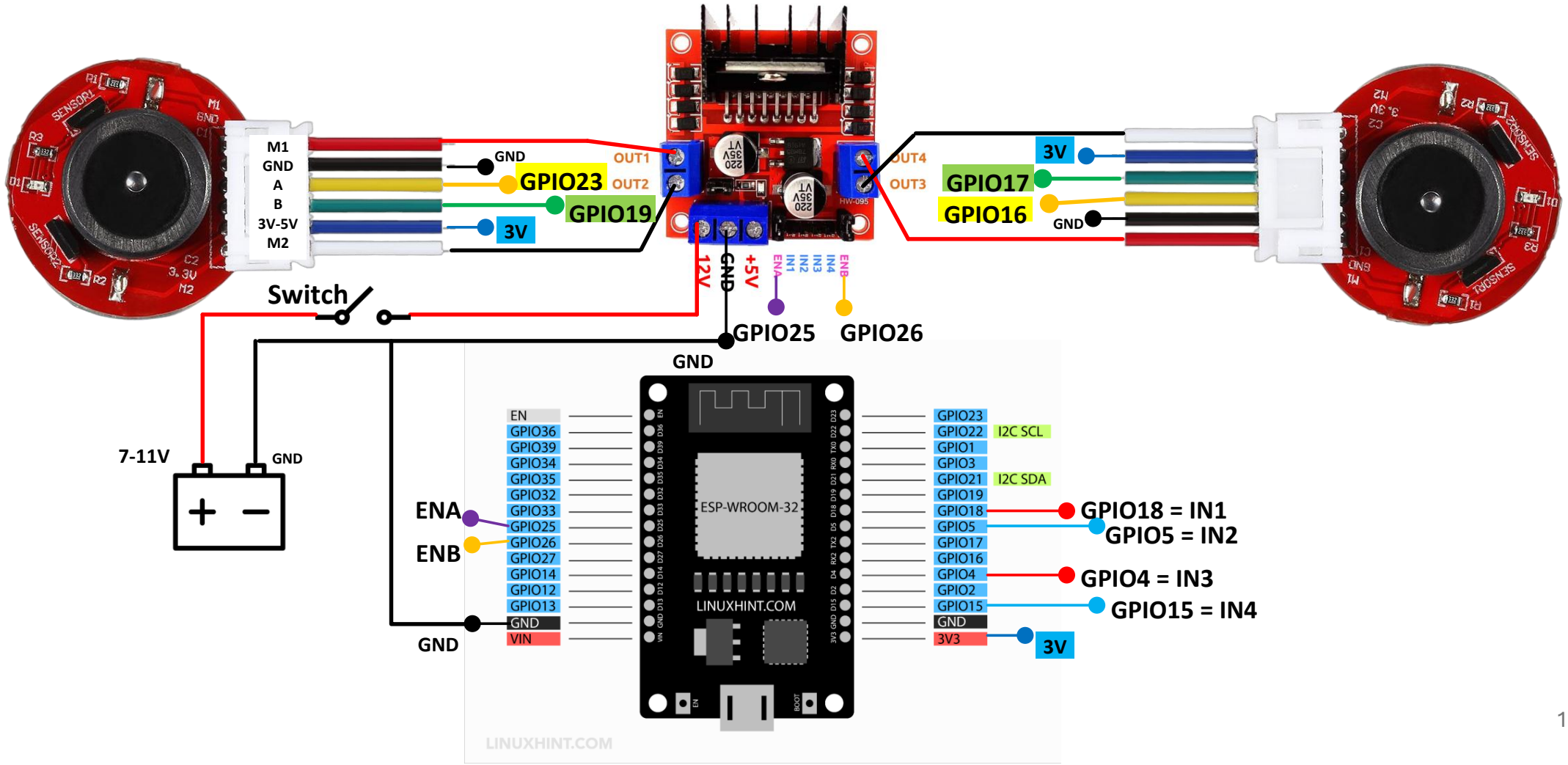
หยุดตรวจรอให้คะแนนและรอเพื่อนๆ !!!



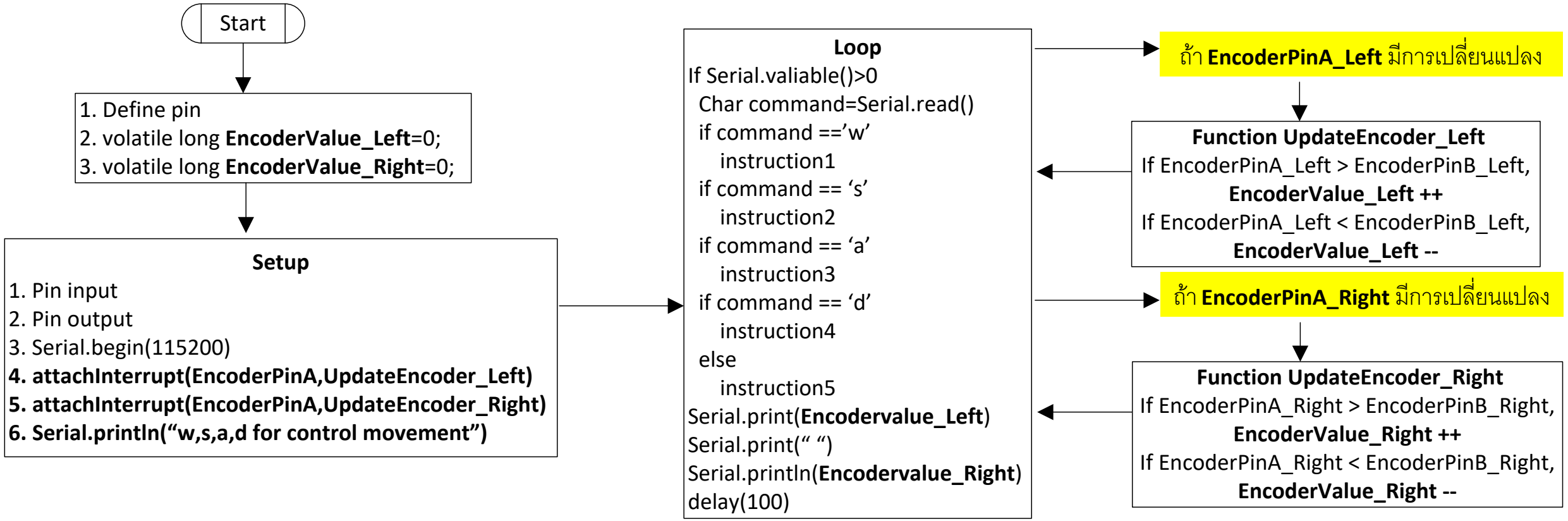
Content

Chapter2. Two motor movement by basic keyboard

1. Connect the right DC motor between the ESP32 and the L298N driver.
2. Connect two PWM pins for the motors: GPIO25 = ENA and GPIO26 = ENB.



Flowchart: Control Two wheels and PWM by keyboard



Arduino code

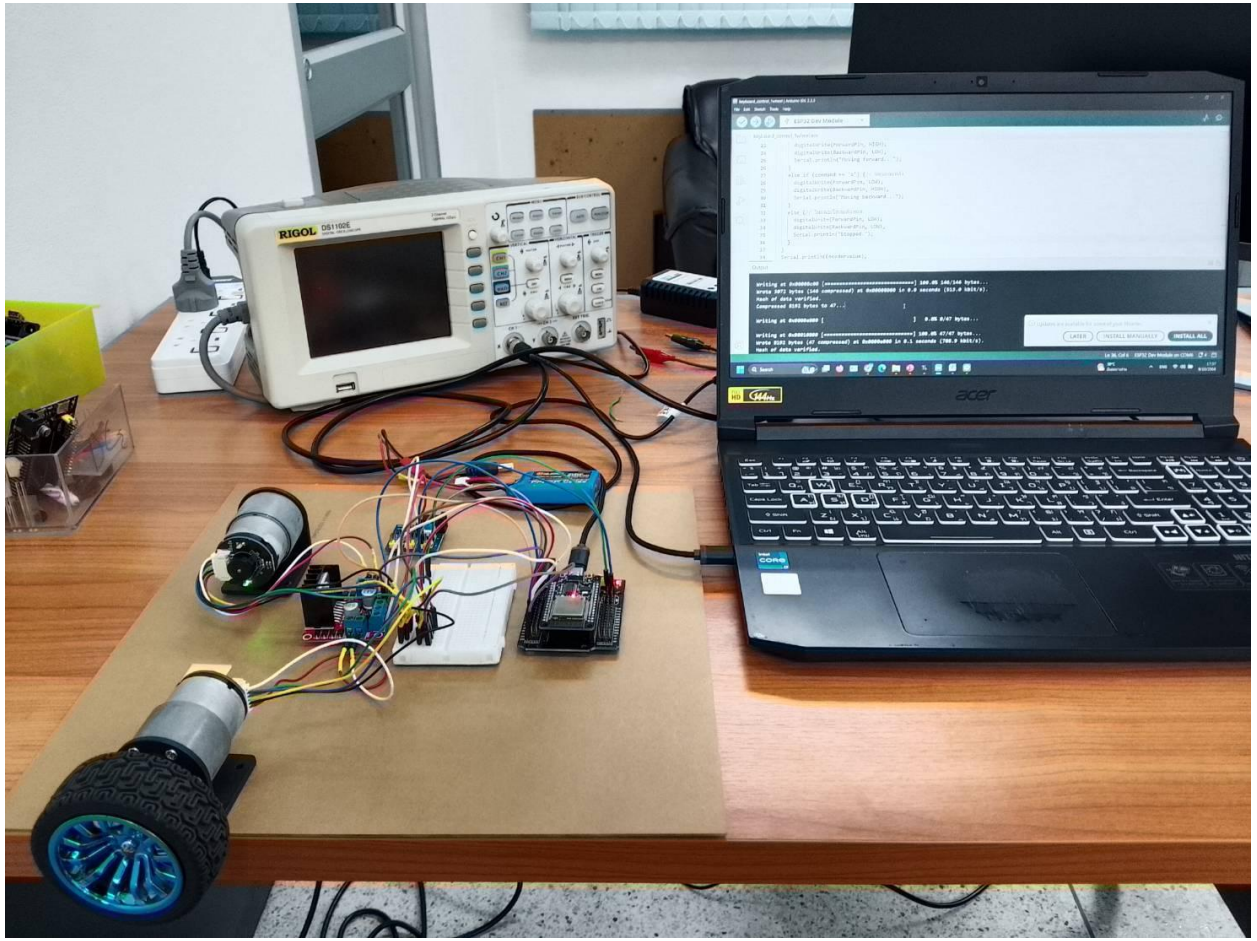
```
#define EncoderA_Left 23
#define EncoderB_Left 19
#define Forward_Left 18 // In1
#define Backward_Left 5 // In2
#define Enable_Left 25

#define EncoderA_Right 17
#define EncoderB_Right 16
#define Forward_Right 4 // In3
#define Backward_Right 15 // In4
#define Enable_Right 26

// Setting PWM properties
const int freq = 30000;
const int resolution = 8;
const int pwmchanne1 = 0;
const int pwmchanne2 = 1;
const int duty_Left = 150;
const int duty_Right = 255;

volatile long Encodervalue_Left = 0;
volatile long Encodervalue_Right = 0;

```

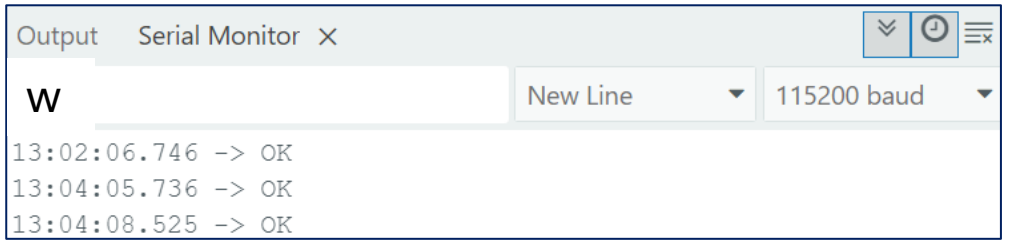
```
void IRAM_ATTR updateEncoder_Left() {
  if (digitalRead(EncoderA_Left) > digitalRead(EncoderB_Left))
    Encodervalue_Left++;
  else
    Encodervalue_Left--;
}
void IRAM_ATTR updateEncoder_Right() {
  if (digitalRead(EncoderA_Right) > digitalRead(EncoderB_Right))
    Encodervalue_Right++;
  else
    Encodervalue_Right--;
}
```

```
37  void setup() {
38    pinMode(EncoderA_Left, INPUT);
39    pinMode(EncoderB_Left, INPUT);
40    pinMode(Forward_Left, OUTPUT);
41    pinMode(Backward_Left, OUTPUT);
42    ledcAttachChannel(Enable_Left, freq, resolution, pwmchanne1);
43    ledcWriteChannel(pwmchanne1, duty_Left);
44
45    pinMode(EncoderA_Right, INPUT);
46    pinMode(EncoderB_Right, INPUT);
47    pinMode(Forward_Right, OUTPUT);
48    pinMode(Backward_Right, OUTPUT);
49    ledcAttachChannel(Enable_Right, freq, resolution, pwmchanne2);
50    ledcWriteChannel(pwmchanne2, duty_Right);
51
52    Serial.begin(115200);
53    attachInterrupt(digitalPinToInterrupt(EncoderA_Left), updateEncoder_Left, RISING);
54    attachInterrupt(digitalPinToInterrupt(EncoderA_Right), updateEncoder_Right, RISING);
55    Serial.println("w is forward, s is backward, a is turn left, d is turn right");
56  }
57
```

```
58  void loop() {
59    if (Serial.available() > 0) {// ตรวจสอบว่ามีข้อมูลจาก Serial เข้ามาหรือไม่
60      char command = Serial.read();  // อ่านอักษรจาก Serial
61      if (command == 'w') {// หมุนเดินหน้า
62        digitalWrite(Forward_Left, HIGH);
63        digitalWrite(Backward_Left, LOW);
64        digitalWrite(Forward_Right, HIGH);
65        digitalWrite(Backward_Right, LOW);
66        Serial.println("Moving forward...");
67      }
68      else if (command == 's') {// หมุนถอยหลัง
69        digitalWrite(Forward_Left, LOW);
70        digitalWrite(Backward_Left, HIGH);
71        digitalWrite(Forward_Right, LOW);
72        digitalWrite(Backward_Right, HIGH);
73        Serial.println("Moving backward...");
74      }
75      else if (command == 'a') {// turn left
76        digitalWrite(Forward_Left, HIGH);
77        digitalWrite(Backward_Left, LOW);
78        digitalWrite(Forward_Right, LOW);
79        digitalWrite(Backward_Right, HIGH);
80        Serial.println("Turn Left");
81      }
82      else if (command == 'd') { // turn right
83        digitalWrite(Forward_Left, LOW);
84        digitalWrite(Backward_Left, HIGH);
85        digitalWrite(Forward_Right, HIGH);
86        digitalWrite(Backward_Right, LOW);
87        Serial.println("Turn Left");
88      }
89
90      else {// ไม่กดอะไรเลยคือหยุด
91        digitalWrite(Forward_Left, LOW);
92        digitalWrite(Backward_Left, LOW);
93        digitalWrite(Forward_Right, LOW);
94        digitalWrite(Backward_Right, LOW);
95        Serial.println("Stopped.");
96      }
97    }
98    Serial.print(Encodervalue_Left);
99    Serial.print(" ");
100   Serial.println(Encodervalue_Right);
101   delay(100);
102 }
103
```

Summary





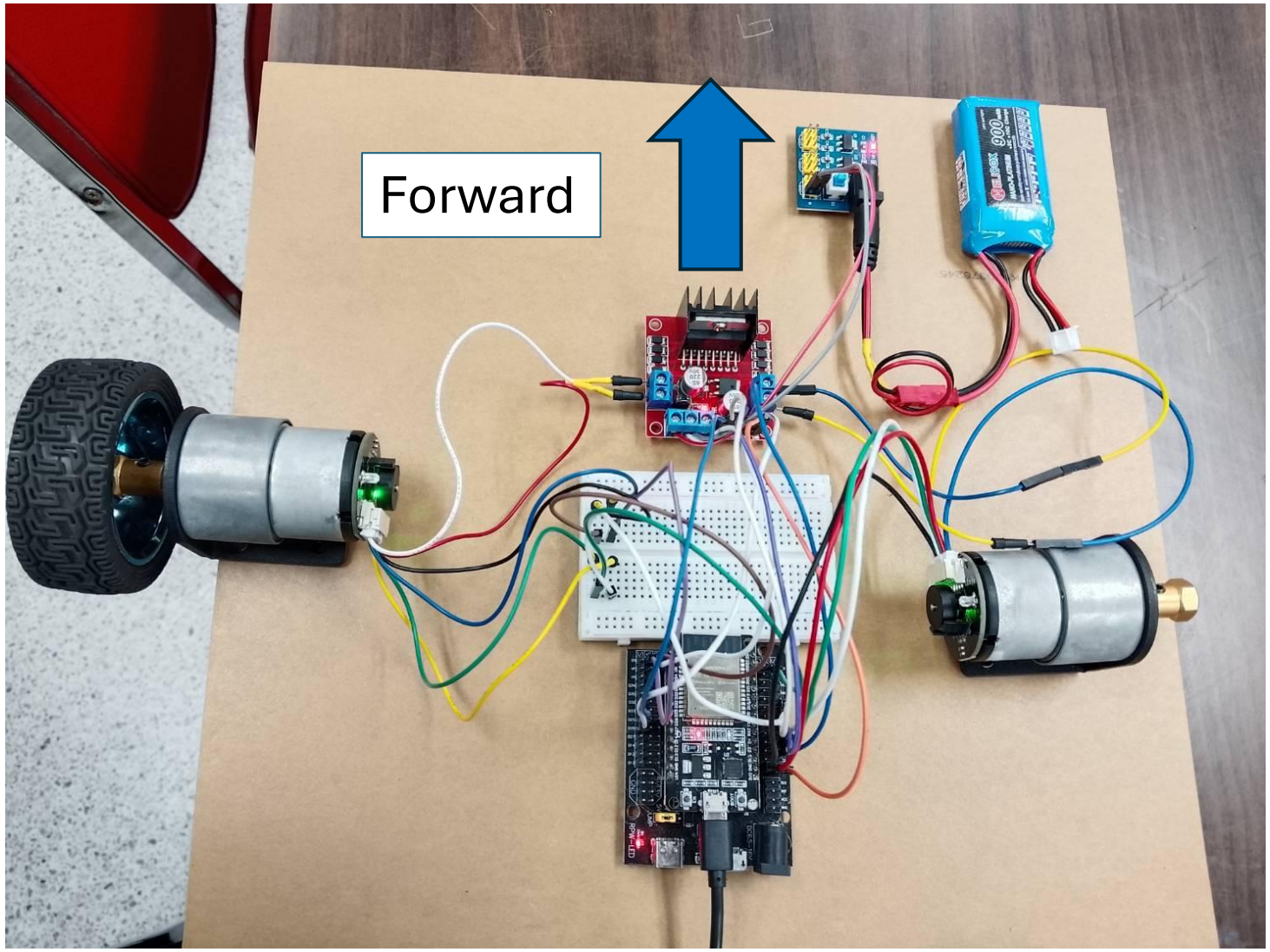
Press ‘w’ motor is forward
Press ‘a’ motor is turn left
Press ‘d’ motor is turn right
Press ‘s’ motor is backward

ปัญหาการเขียน **code** พื้นฐานคือ

1. ไม่สามารถควบคุมความเร็วแต่ละล้อได้
2. ไม่สามารถควบคุม **servo motor** หลายตัวได้

Content

Chapter3: Two motors and control by serial command: “o 255 255”



Original Motor move

Left speed

Right speed

Output Serial Monitor ×

o 255 255

Forward

Output Serial Monitor ×

o 255 -255

15:21:34.623 -> OK

right

Output Serial Monitor ×

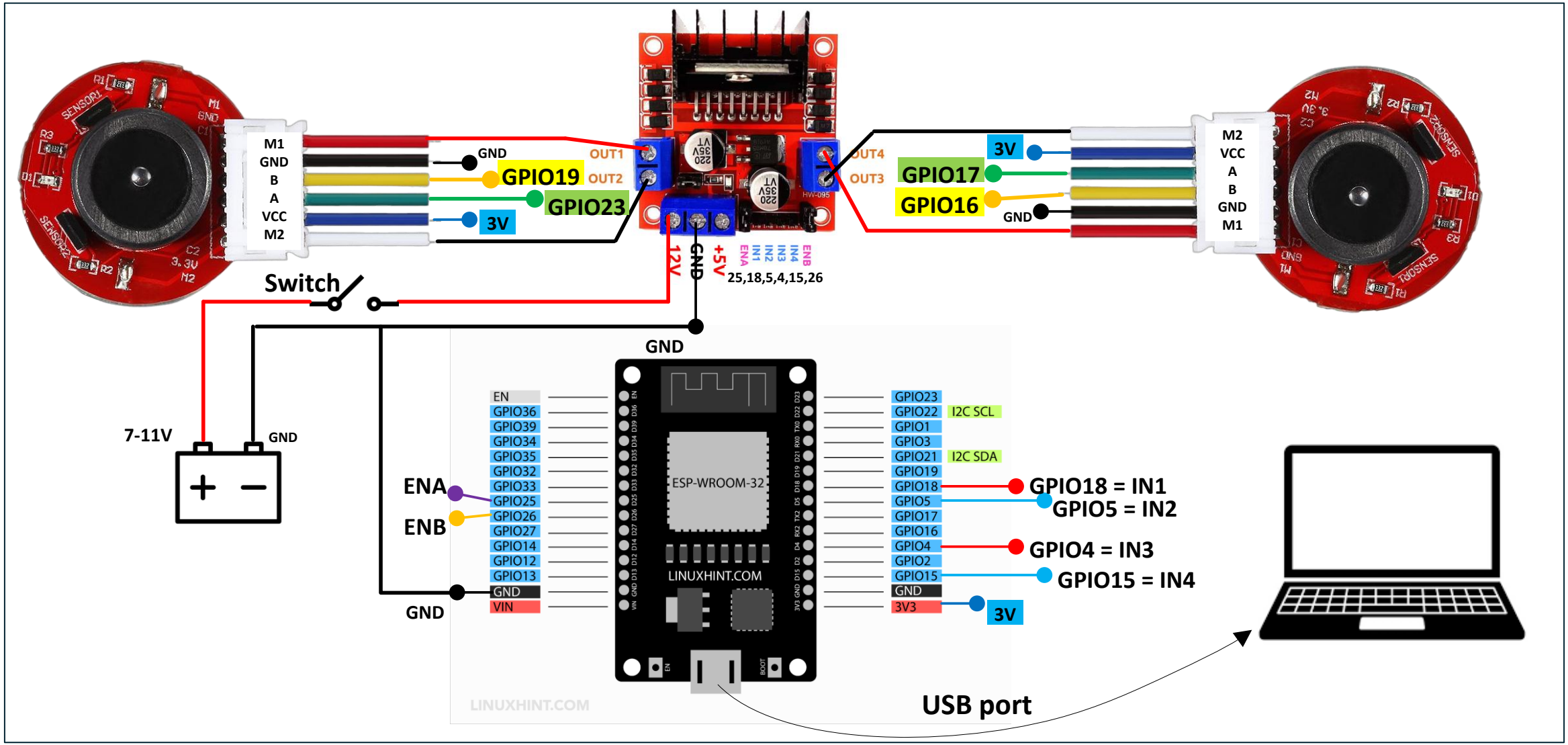
o -255 255

15:21:34.623 -> OK

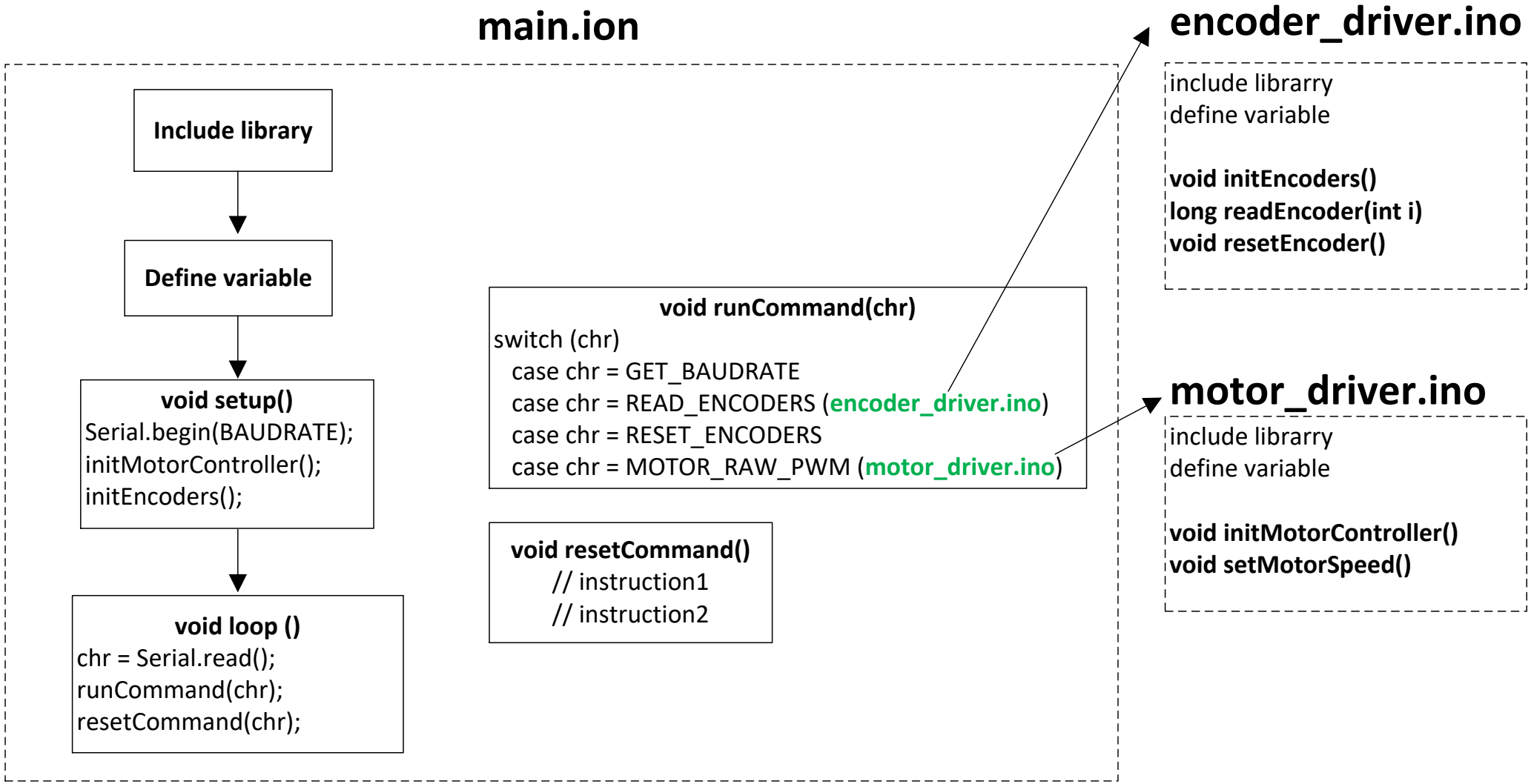
15:21:51.888 -> OK

left

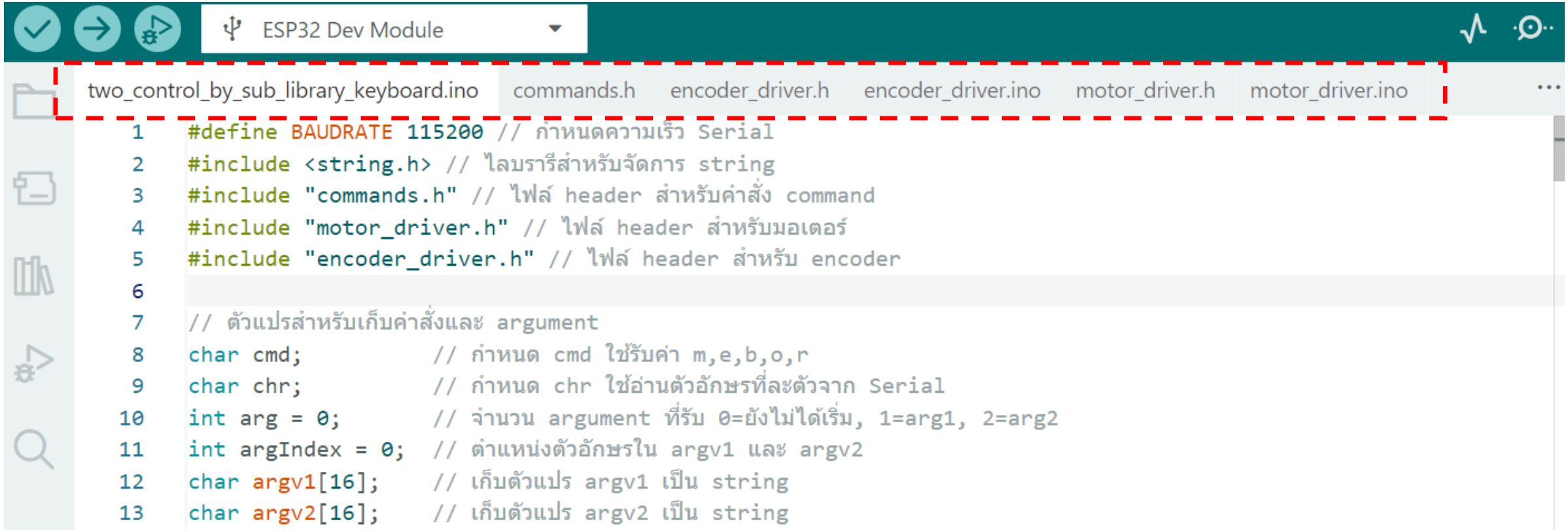
อย่าต่อผิด!!!


M1
GND
B
A
VCC
M2
GND
OUT1
OUT2
GPIO19
GPIO23
3V
OUT4
OUT3
GPIO17
GPIO16
3V
GND
M2
VCC
A
B
GND
M1
12V
GND
+5V
ENA
IN1
IN2
IN3
IN4
ENB
25,18,5,4,15,26
Switch
7-11V
GND
GND
ENA
ENB
GND
EN
GPIO36
GPIO39
GPIO34
GPIO35
GPIO32
GPIO33
GPIO25
GPIO26
GPIO27
GPIO14
GPIO12
GPIO13
GND
VIN
ESP-WROOM-32
LINUXHINT.COM
GPIO23
GPIO22
I2C SCL
GPIO1
GPIO3
GPIO21
I2C SDA
GPIO19
GPIO18
GPIO5
GPIO17
GPIO16
GPIO4
GPIO2
GPIO15
GND
3V3
GPIO18 = IN1
GPIO5 = IN2
GPIO4 = IN3
GPIO15 = IN4
3V
USB port
LINUXHINT.COM

Flowchart


main.ion
Include library
Define variable
void setup()
Serial.begin(BAUDRATE);
initMotorController();
initEncoders();
void loop ()
chr = Serial.read();
runCommand(chr);
resetCommand(chr);
void runCommand(chr)
switch (chr)
case chr = GET_BAUDRATE
case chr = READ_ENCODERS (encoder_driver.ino)
case chr = RESET_ENCODERS
case chr = MOTOR_RAW_PWM (motor_driver.ino)
void resetCommand()
// instruction1
// instruction2
encoder_driver.ino
include librarry
define variable
void initEncoders()
long readEncoder(int i)
void resetEncoder()
motor_driver.ino
include librarry
define variable
void initMotorController()
void setMotorSpeed()

A folder consists of the following files: main.ino, commands.h, encoder_driver.h, encoder_driver.ino, motor_driver.h, and motor_driver.ino.



1. Create a main file

two_control_by_sub_library_keyboard.ino

```ino
#define BAUDRATE 115200 // กำหนดความเร็ว Serial
#include <string.h> // ไลบรารีสำหรับจัดการ string
#include "commands.h" // ไฟล์ header สำหรับคำสั่ง command
#include "motor_driver.h" // ไฟล์ header สำหรับมอเตอร์
#include "encoder_driver.h" // ไฟล์ header สำหรับ encoder

// ตัวแปรสำหรับเก็บคำสั่งและ argument
char cmd;           // กำหนด cmd ใช้รับค่า m,e,b,o,r
char chr;           // กำหนด chr ใช้อ่านตัวอักษรที่ละตัวจาก Serial
int arg = 0;        // จำนวน argument ที่รับ 0=ยังไม่ได้เริ่ม, 1=arg1, 2=a
int argIndex = 0;   // ตำแหน่งตัวอักษรใน argv1 และ argv2
char argv1[16];     // เก็บตัวแปร argv1 เป็น string
char argv2[16];     // เก็บตัวแปร argv2 เป็น string
long arg1;          // เก็บตัวแปร arg1 เป็นตัวเลข
long arg2;          // เก็บตัวแปร arg2 เป็นตัวเลข

// ฟังก์ชันรีเซ็ตตัวแปรเพื่อรอรับคำสั่งใหม่
void resetCommand() {
  cmd = 0;                      // รีเซ็ต command
  arg = 0;                       // รีเซ็ตจำนวน arg
  argIndex = 0;                  // รีเซ็ตจำนวน argIndex
  memset(argv1, 0, sizeof(argv1)); // ล้าง argv1
  memset(argv2, 0, sizeof(argv2)); // ล้าง argv2
}
```

```ino
// ฟังก์ชันรัน command
void runCommand() {
  // แปลง argument string เป็นตัวเลข
  arg1 = atol(argv1); // เช่น argv1="255" แล้ว arg1=255
  arg2 = atol(argv2); // เช่น argv2="255" แล้ว arg2=255
  // ตรวจสอบ command
  switch (cmd) {
    case GET_BAUDRATE:
      // ส่งค่า BAUDRATE กลับทาง Serial
      Serial.println(BAUDRATE);
      break;
    case READ_ENCODERS:
      // อ่านค่า encoder ซ้ายและขวาแล้วส่งกลับ
      Serial.print(readEncoder(LEFT));
      Serial.print(" ");
      Serial.println(readEncoder(RIGHT));
      break;
    case RESET_ENCODERS:
      // รีเซ็ตค่า encoder
      resetEncoders();
      Serial.println("OK");
      break;
    case MOTOR_RAW_PWM:
      // ตั้งความเร็วมอเตอร์ตาม arg1 และ arg2
      setMotorSpeeds(arg1, arg2);
      Serial.println("OK");
      break;
    default:
      // command ไม่ถูกต้อง
      Serial.println("Invalid Command");
      break;
  }
}
```

```
60  void setup() { // ฟังก์ชัน setup() รันครั้งเดียวตอนเริ่มต้น
61    Serial.begin(BAUDRATE); // เริ่ม Serial
62    initMotorController();  // เตรียมมอเตอร์
63    initEncoders();         // เตรียม encoder
64  }
65  void loop() { // ฟังก์ชัน loop() รันซ้ำตลอดเวลา
66    while (Serial.available()) { // ตรวจสอบว่ามีข้อมูล Serial เข้ามาหรือไม่
67      chr = Serial.read(); // อ่านตัวอักษรตัวต่อไป
68      if (chr == '\r' || chr == '\n') { // ถ้าเจอ Enter ถือว่าจบคำสั่ง
69        if (arg == 1) argv1[argIndex] = '\0'; // เติม null ให้ argv1
70        if (arg == 2) argv2[argIndex] = '\0'; // เติม null ให้ argv2
71        runCommand();                         // รันคำสั่ง
72        resetCommand();                       // รีเซ็ตเพื่อรับคำสั่งใหม่
73      }
74      else if (chr == ' ') {  // ถ้าเจอ space แยก argument
75        if (arg == 0) arg = 1;                // เริ่ม argument 1
76        else if (arg == 1) {
77          argv1[argIndex] = '\0';             // จบ argument 1
78          arg = 2;                             // เริ่ม argument 2
79          argIndex = 0;                        // รีเซ็ต index
80        }
81      }
82      else { // เก็บตัวอักษร
83        if (arg == 0) cmd = chr;              // ถ้า arg=0 ถือว่าเป็น command
84        else if (arg == 1 && argIndex < 15) argv1[argIndex++] = chr; // เก็บ argv1
85        else if (arg == 2 && argIndex < 15) argv2[argIndex++] = chr; // เก็บ argv2
86      }
87    }
88  }
89
```

Serial command

arg=0	arg=1	arg=2
o	255	255(enter)
cmd=o	argv1="255\0"	argv2="255\0"

Serial command

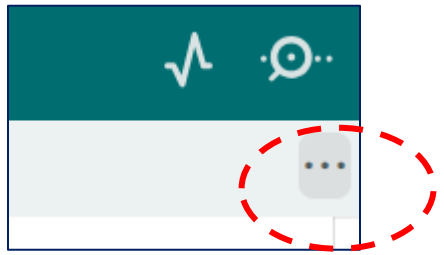
✅ สรุปการทำงานสำหรับ `m 255 255`

ลำดับ	chr	cmd	argv1	argv2	arg	argIndex	การทำงาน
1	m	m	""	""	0	0	เก็บ command
2	' '	m	""	""	1	0	เริ่ม argument 1
3	2	m	"2"	""	1	1	เก็บ argv1
4	5	m	"25"	""	1	2	เก็บ argv1
5	5	m	"255"	""	1	3	เก็บ argv1
6	' '	m	"255\0"	""	2	0	จบ argv1, เริ่ม argv2
7	2	m	"255\0"	"2"	2	1	เก็บ argv2
8	5	m	"255\0"	"25"	2	2	เก็บ argv2
9	5	m	"255\0"	"255"	2	3	เก็บ argv2
10	<Enter>	m	"255\0"	"255\0"	2	3	เติม null, runCommand(), resetCommand()

2. Create a file “commands.h”

commands.h

```
1   #ifndef COMMANDS_H //Header Guard
2   #define COMMANDS_H //Header Guard
3
4   #define GET_BAUDRATE    'b'
5   #define READ_ENCODERS   'e'
6   #define MOTOR_RAW_PWM   'o'
7   #define RESET_ENCODERS 'r'
8   #define LEFT    0
9   #define RIGHT   1
10
11  #endif  // COMMANDS_H
12
```



(new tab)

สร้างไฟล์ใหม่

3. Create files “encoder_driver.h” and “encoder_driver.ino”

encoder_driver.h

```
1   #define LEFT_ENC_PIN_A    23
2   #define LEFT_ENC_PIN_B    19
3   #define RIGHT_ENC_PIN_A   16
4   #define RIGHT_ENC_PIN_B   17
5
6   void initEncoders();
7   long readEncoder(int i);
8   void resetEncoder(int i);
9   void resetEncoders();
10
```

encoder_driver.ino

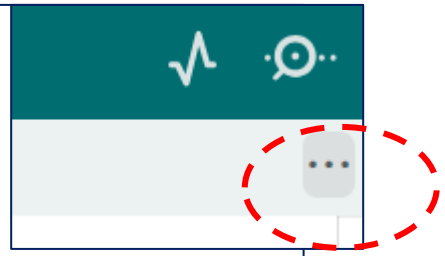
```
1   #include "encoder_driver.h"
2   #include "commands.h"
3   #include <ESP32Encoder.h>
4   ESP32Encoder leftEnc;
5   ESP32Encoder rightEnc;
6
7   void initEncoders() {
8     leftEnc.attachHalfQuad(LEFT_ENC_PIN_A, LEFT_ENC_PIN_B);
9     rightEnc.attachHalfQuad(RIGHT_ENC_PIN_A, RIGHT_ENC_PIN_B);
10    resetEncoders();
11  }
12  long readEncoder(int i) {
13    if (i == LEFT) return leftEnc.getCount();
14    return rightEnc.getCount();
15  }
16  void resetEncoder(int i) {
17    if (i == LEFT) leftEnc.clearCount();
18    else rightEnc.clearCount();
19  }
20  void resetEncoders() {
21    resetEncoder(LEFT);
22    resetEncoder(RIGHT);
23  }
```

Install library!!!

4. Create “motor_driver.h”

motor_driver.h

```
1   #define LEFT_MOTOR_FORWARD   18 // IN1
2   #define LEFT_MOTOR_BACKWARD  5 // IN2
3   #define LEFT_MOTOR_ENABLE    25 // ENA
4   #define RIGHT_MOTOR_FORWARD  4 // IN3
5   #define RIGHT_MOTOR_BACKWARD 15 // IN4
6   #define RIGHT_MOTOR_ENABLE   26 // ENB
7
8   void initMotorController();
9   void setMotorSpeed(int i, int spd);
10  void setMotorSpeeds(int leftSpeed, int rightSpeed);
```



(new tab)

สร้างไฟล์ใหม่

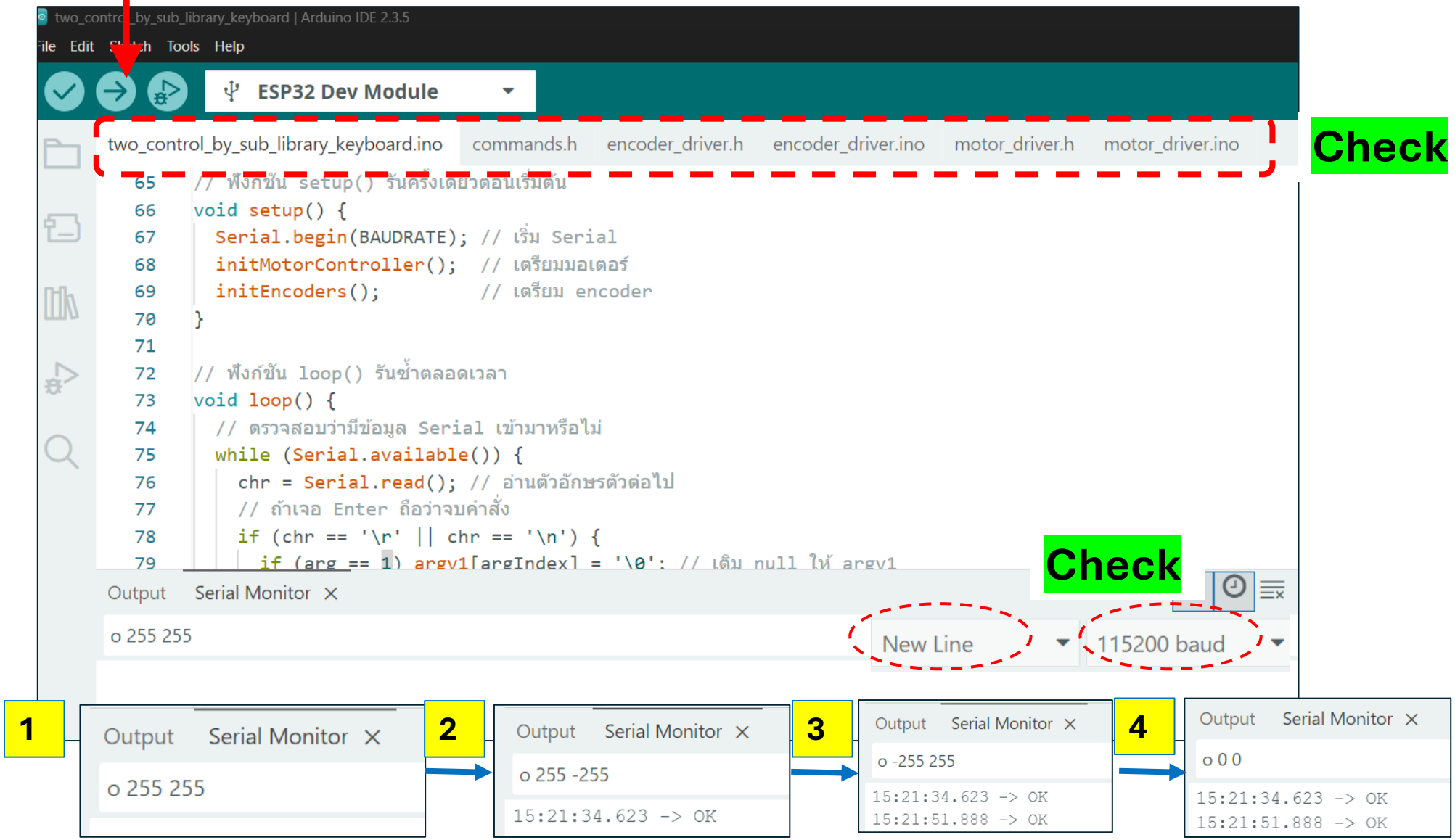
5. Create “motor_driver.ino”

motor_driver.ino

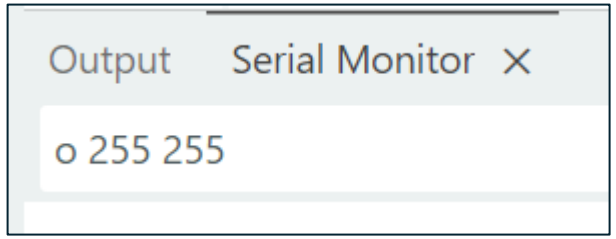
```
#include "motor_driver.h"
#include "commands.h"

#define PWM_FREQ  30000
#define PWM_RES   8
#define MAX_PWM   255

void initMotorController() {
  // Attach pins directly (NEW API)
  ledcAttach(LEFT_MOTOR_FORWARD, PWM_FREQ, PWM_RES);
  ledcAttach(LEFT_MOTOR_BACKWARD, PWM_FREQ, PWM_RES);
  ledcAttach(RIGHT_MOTOR_FORWARD, PWM_FREQ, PWM_RES);
  ledcAttach(RIGHT_MOTOR_BACKWARD, PWM_FREQ, PWM_RES);

  pinMode(LEFT_MOTOR_ENABLE, OUTPUT);
  pinMode(RIGHT_MOTOR_ENABLE, OUTPUT);
  digitalWrite(LEFT_MOTOR_ENABLE, HIGH);
  digitalWrite(RIGHT_MOTOR_ENABLE, HIGH);
}
```

```
void setMotorSpeed(int i, int spd) {
  bool reverse = false;
  if (spd < 0) {
    spd = -spd;
    reverse = true;
  }
  spd = constrain(spd, 0, MAX_PWM);
  if (i == LEFT) {
    if (!reverse) {
      ledcWrite(LEFT_MOTOR_FORWARD, spd);
      ledcWrite(LEFT_MOTOR_BACKWARD, 0);
    } else {
      ledcWrite(LEFT_MOTOR_FORWARD, 0);
      ledcWrite(LEFT_MOTOR_BACKWARD, spd);
    }
  } else {
    if (!reverse) {
      ledcWrite(RIGHT_MOTOR_FORWARD, spd);
      ledcWrite(RIGHT_MOTOR_BACKWARD, 0);
    } else {
      ledcWrite(RIGHT_MOTOR_FORWARD, 0);
      ledcWrite(RIGHT_MOTOR_BACKWARD, spd);
    }
  }
}

void setMotorSpeeds(int leftSpeed, int rightSpeed) {
  setMotorSpeed(LEFT, leftSpeed);
  setMotorSpeed(RIGHT, rightSpeed);
}
```

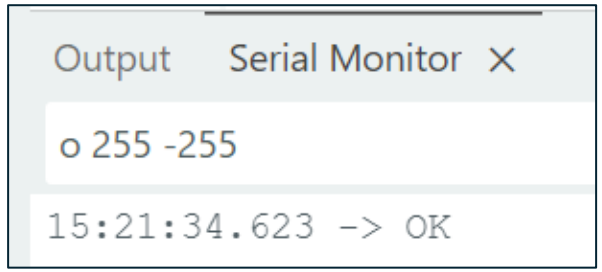
Run program


two_control_by_sub_library_keyboard | Arduino IDE 2.3.5
File Edit Sketch Tools Help
ESP32 Dev Module
two_control_by_sub_library_keyboard.ino
commands.h
encoder_driver.h
encoder_driver.ino
motor_driver.h
motor_driver.ino
Check
// ฟังก์ชัน setup() รันครั้งเดียวตอนเริ่มต้น
void setup() {
Serial.begin(BAUDRATE); // เริ่ม Serial
initMotorController(); // เตรียมมอเตอร์
initEncoders(); // เตรียม encoder
}
// ฟังก์ชัน loop() รันซ้ำตลอดเวลา
void loop() {
// ตรวจสอบว่ามีข้อมูล Serial เข้ามาหรือไม่
while (Serial.available()) {
chr = Serial.read(); // อ่านตัวอักษรตัวต่อไป
// ถ้าเจอ Enter ถือว่าจบคำสั่ง
if (chr == '\r' || chr == '\n') {
Check
Output Serial Monitor
o 255 255
New Line
115200 baud
1
Output Serial Monitor
o 255 255
2
Output Serial Monitor
o 255 -255
15:21:34.623 -> OK
3
Output Serial Monitor
o -255 255
15:21:34.623 -> OK
15:21:51.888 -> OK
4
Output Serial Monitor
o 0 0
15:21:34.623 -> OK
15:21:51.888 -> OK

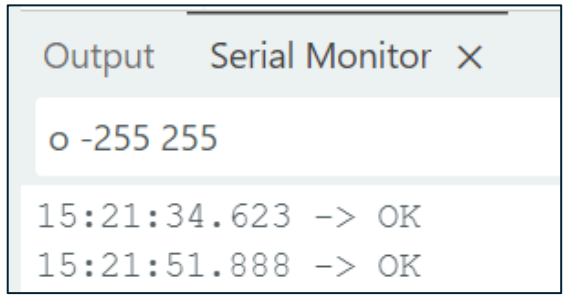
How to control



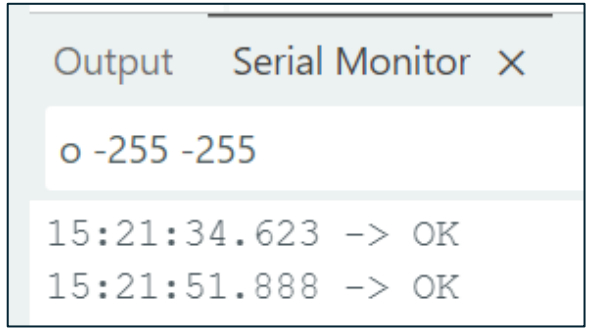
Forward



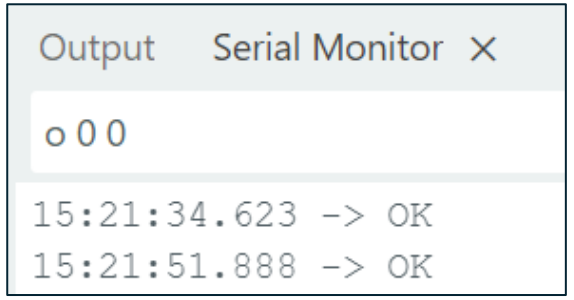
Turn Right



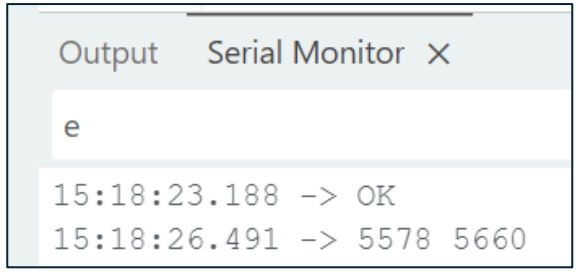
Turn Left



Backward

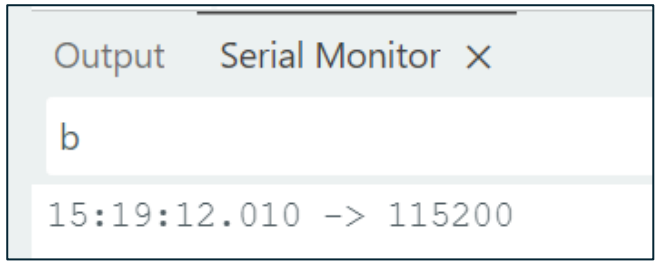


Stop

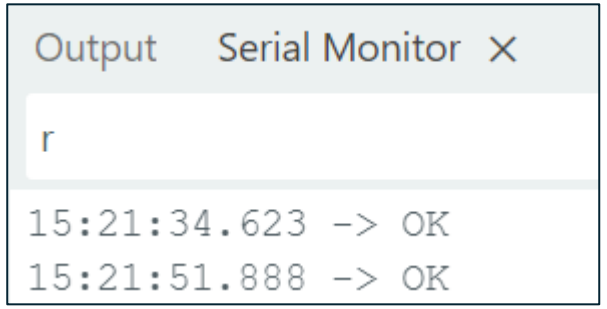


Read Encoder

ถ้าสั่งเดินหน้าแล้วอ่านค่า เรียกอาจารย์

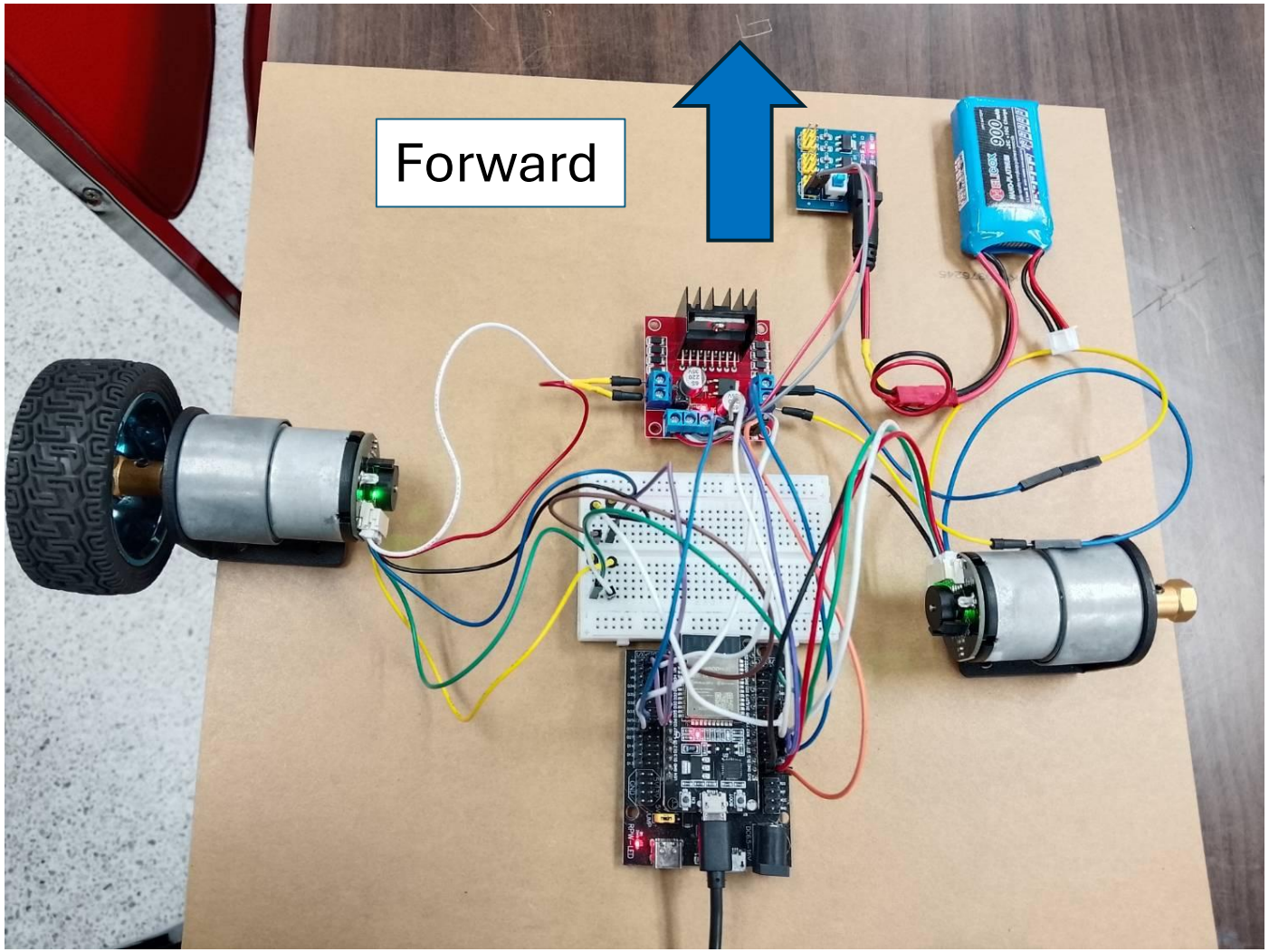


Read board rate

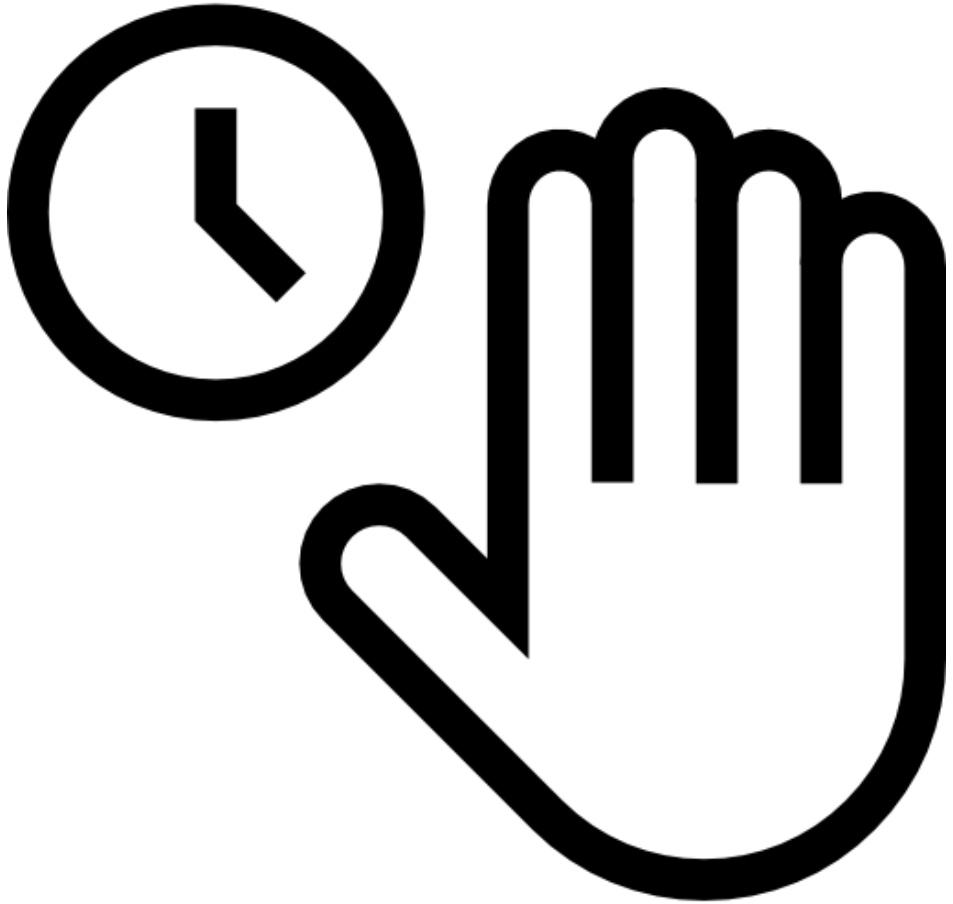


Reset Encoder

Homework1: Have students write a summary and the problems they encountered?

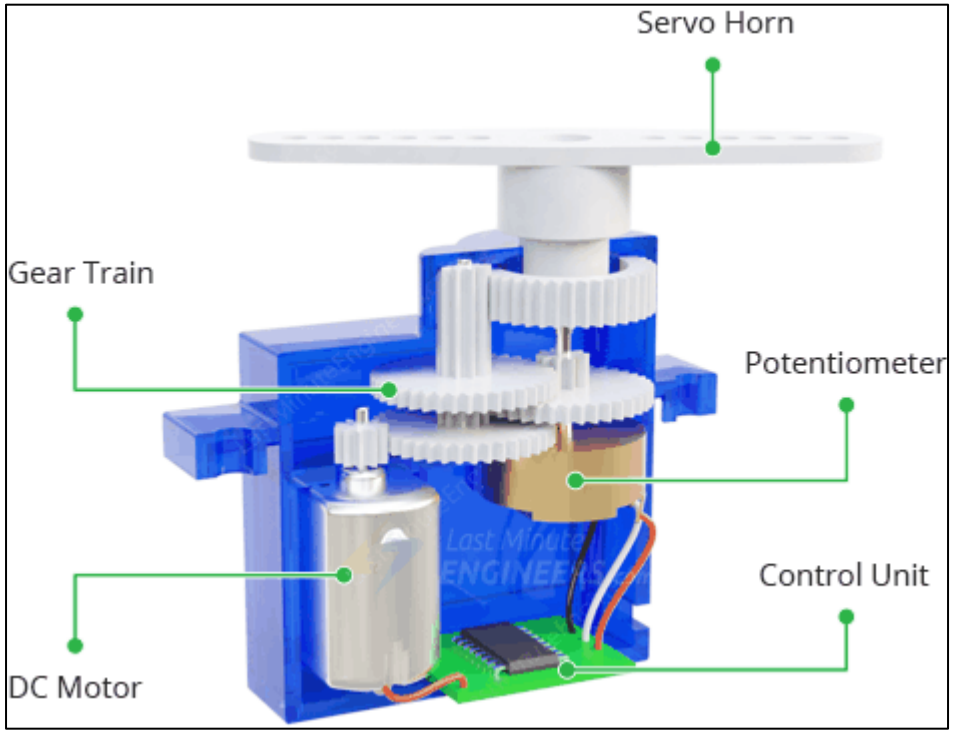


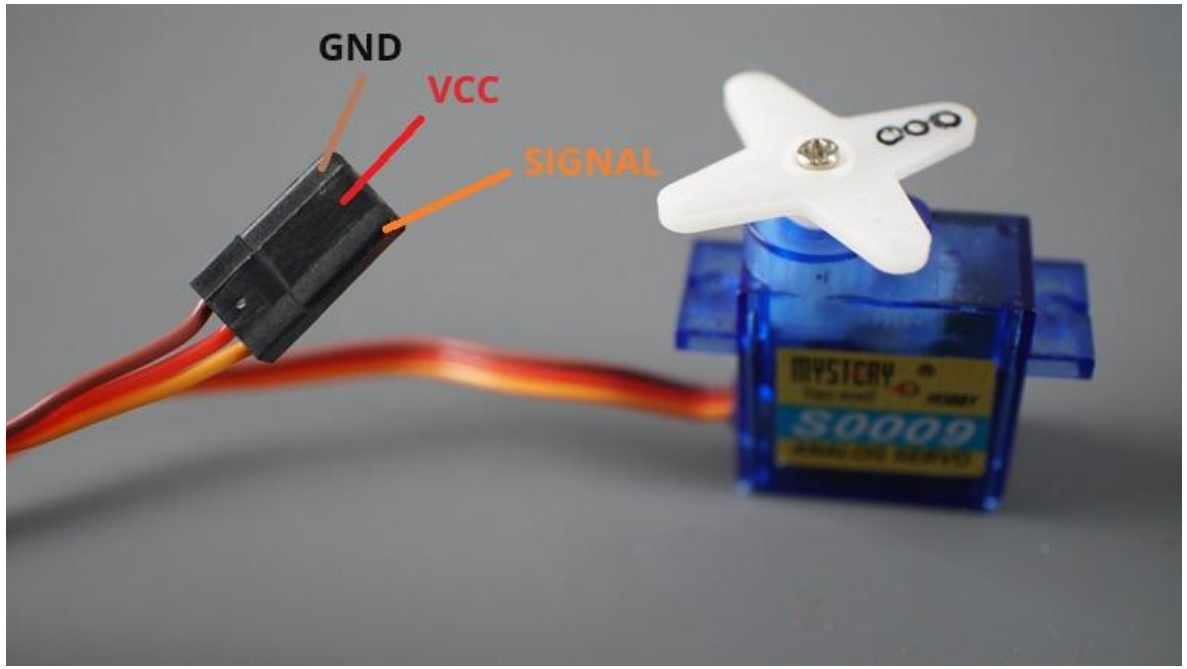
หยุดตรวจรอให้คะแนนและรอเพื่อนๆ !!!



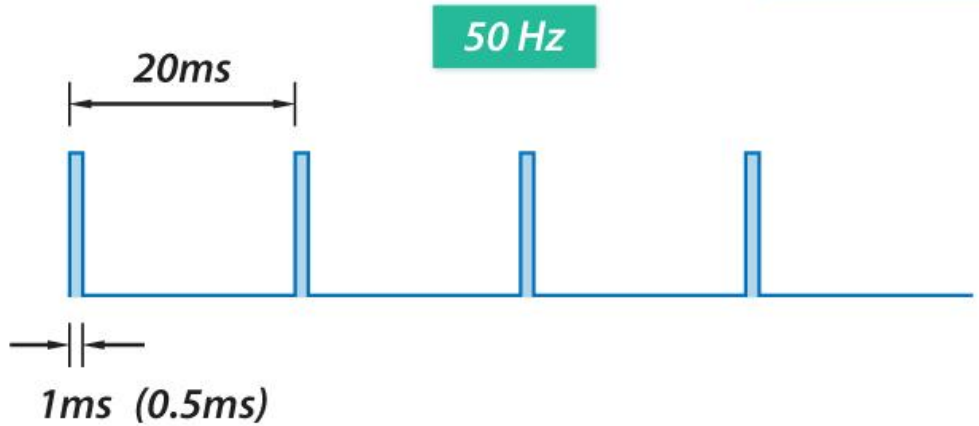
Content

3.1 Basic Servo motor

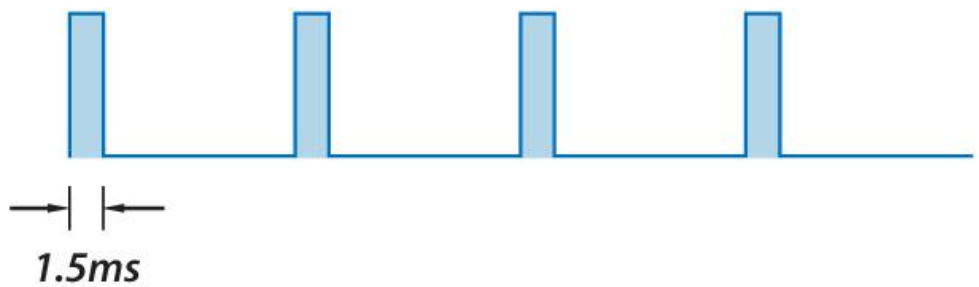




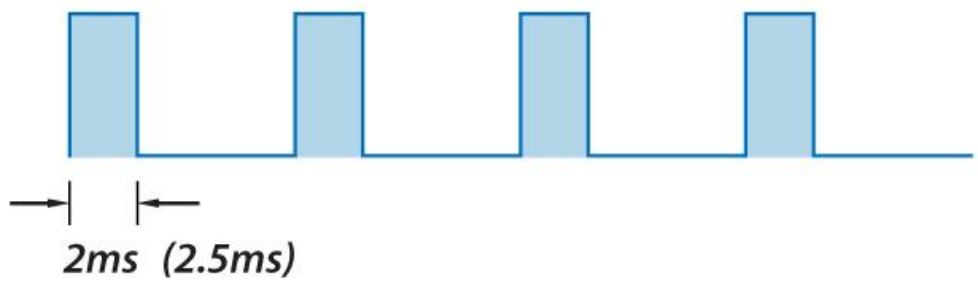
SERVO MOTOR CONTROL



0 Degrees

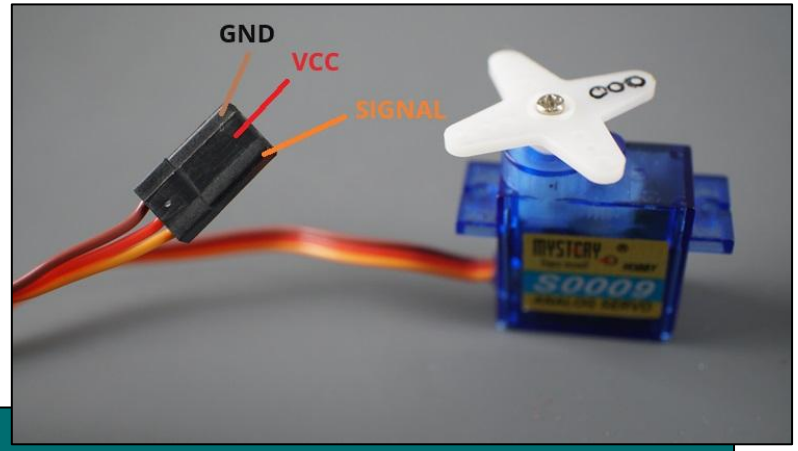


90 Degrees



180 Degrees

Run Servo motor movement!!!



ESP32 Dev Module

LIBRARY MANAGER

esp32servo

Type: All

Topic: All

ESP32Servo by Kevin Harrington,John K. Bennett

Allows ESP32 boards to control servo, tone and analogWrite motors using Arduino...

More info

3.0.9 INSTALL

sketch_dec13b.ino

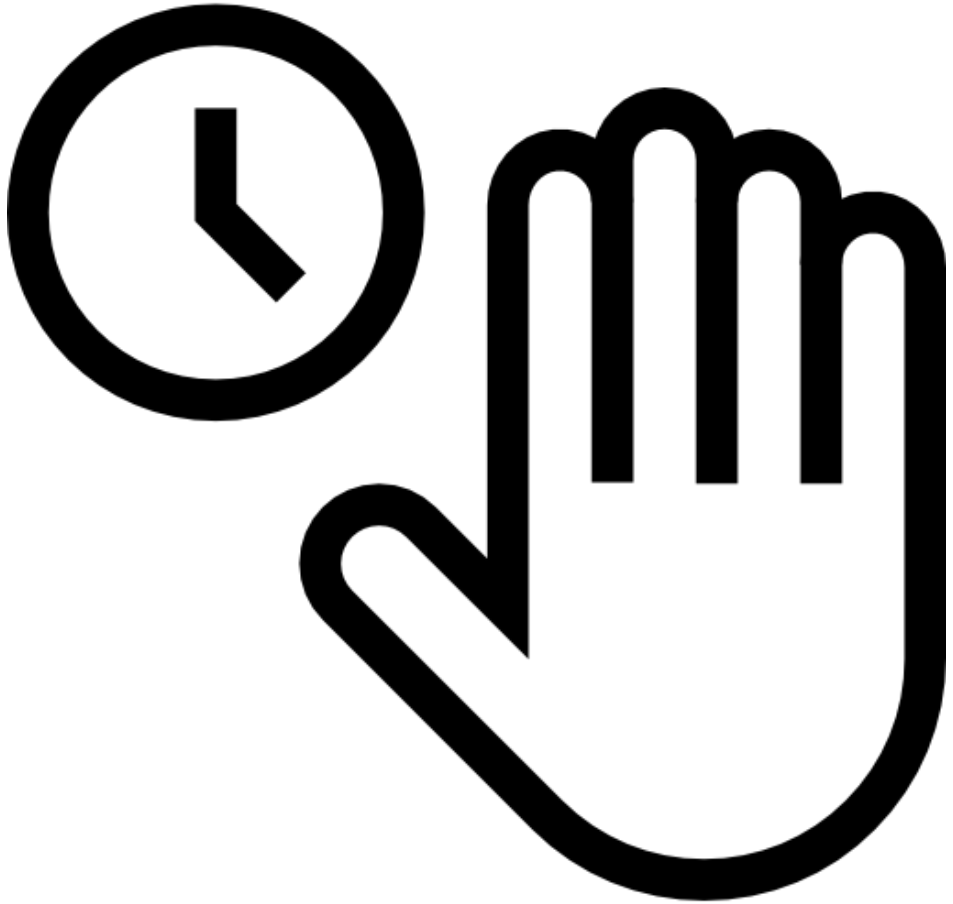
```
1  #include <ESP32Servo.h>
2  Servo MicroServo;
3  int ServoPin = 12;
4  void setup() {
5    MicroServo.setPeriodHertz(50);      // ความถี่ servo มาตรฐาน
6    MicroServo.attach(ServoPin, 500, 2400);
7  }
8  void loop() {
9    MicroServo.write(0);
10   delay(1000);
11   MicroServo.write(45);
12   delay(1000);
13   MicroServo.write(90);
14   delay(1000);
15 }
16
```

ต่อ pin12

ความถี่ 50 Hz

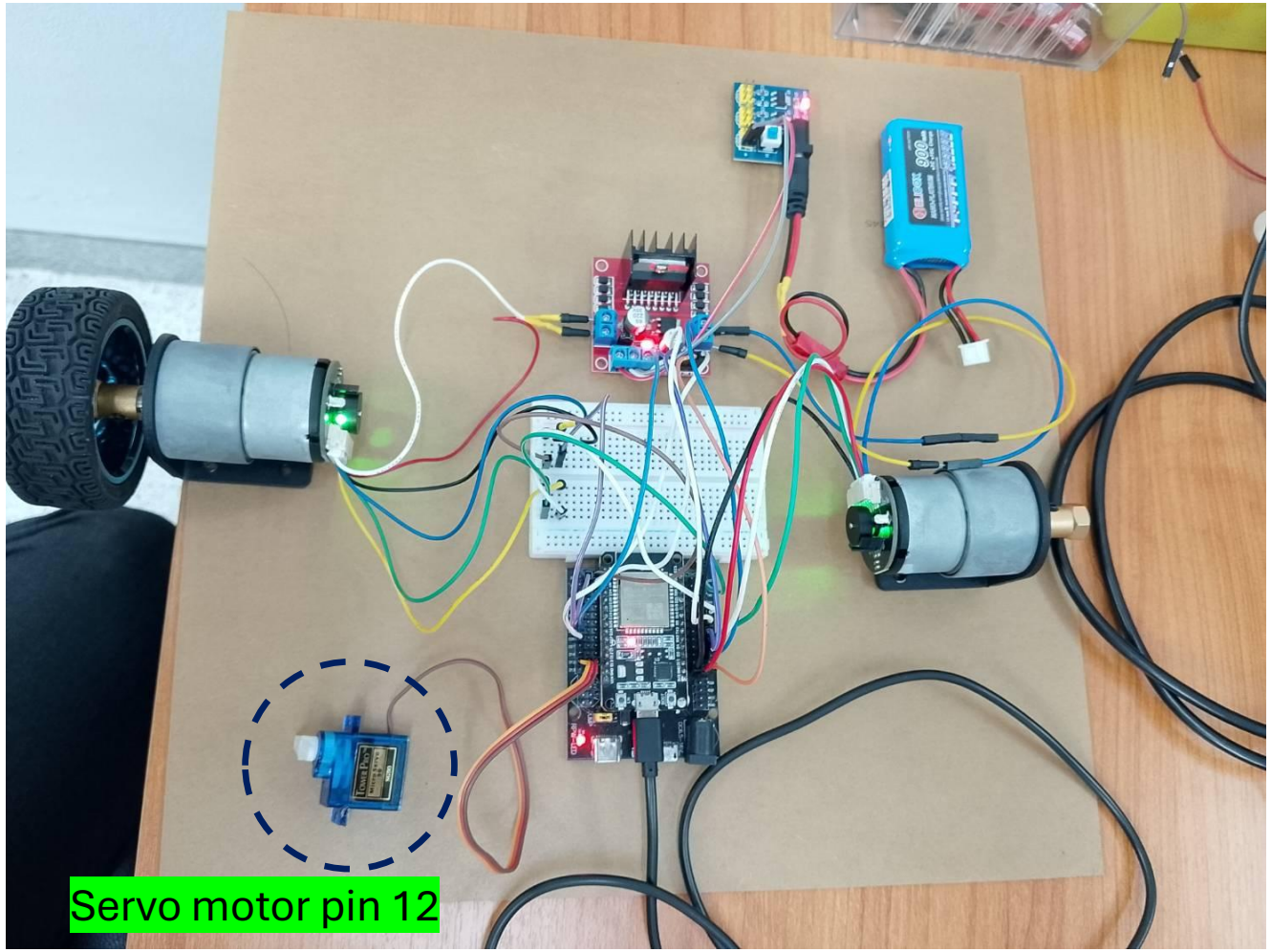
ถ้ารันแล้วมอเตอร์ไม่หมุนรีบถอด servo motor ออกทันทีแล้วเรียกอาจารย์ !!!

หยุดตรวจรอให้คะแนนและรอเพื่อนๆ !!!

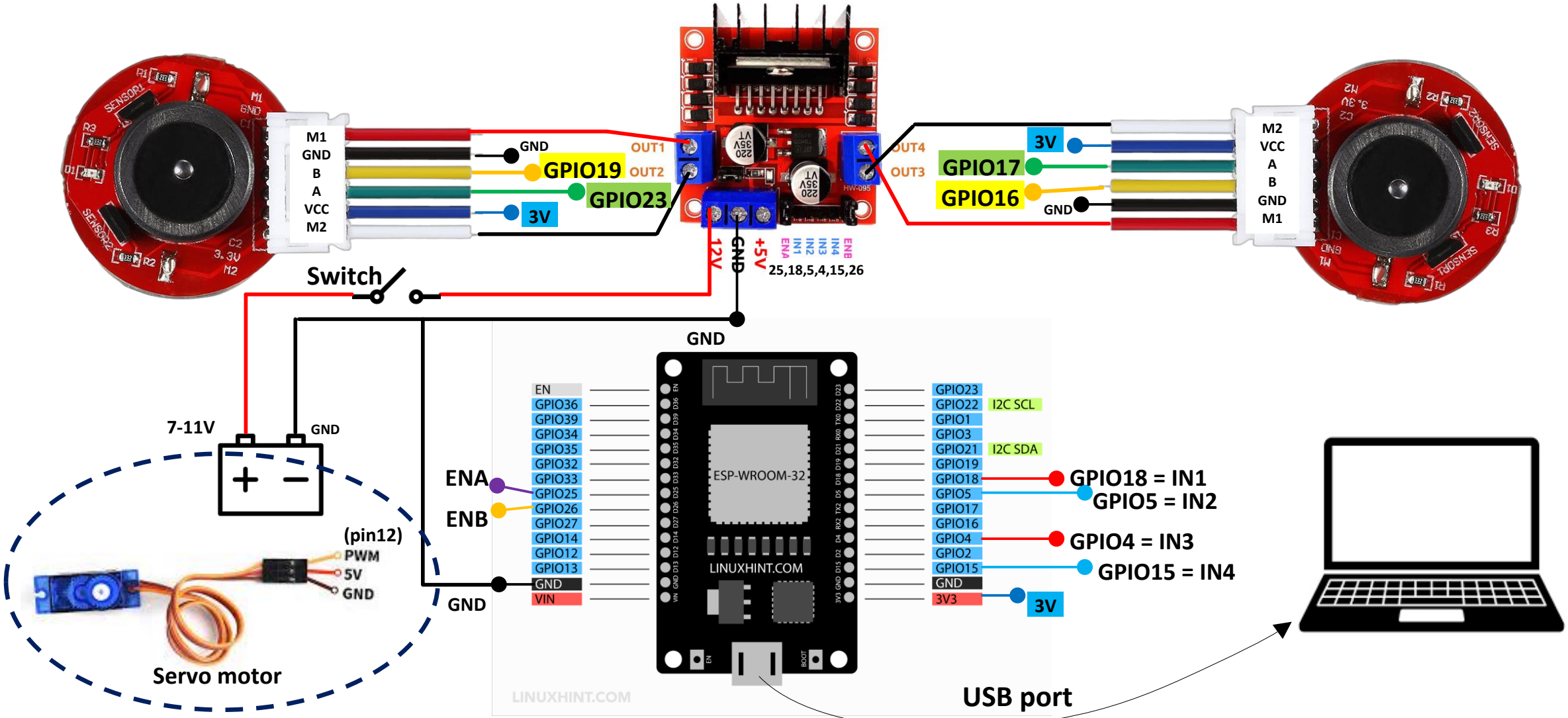


Content

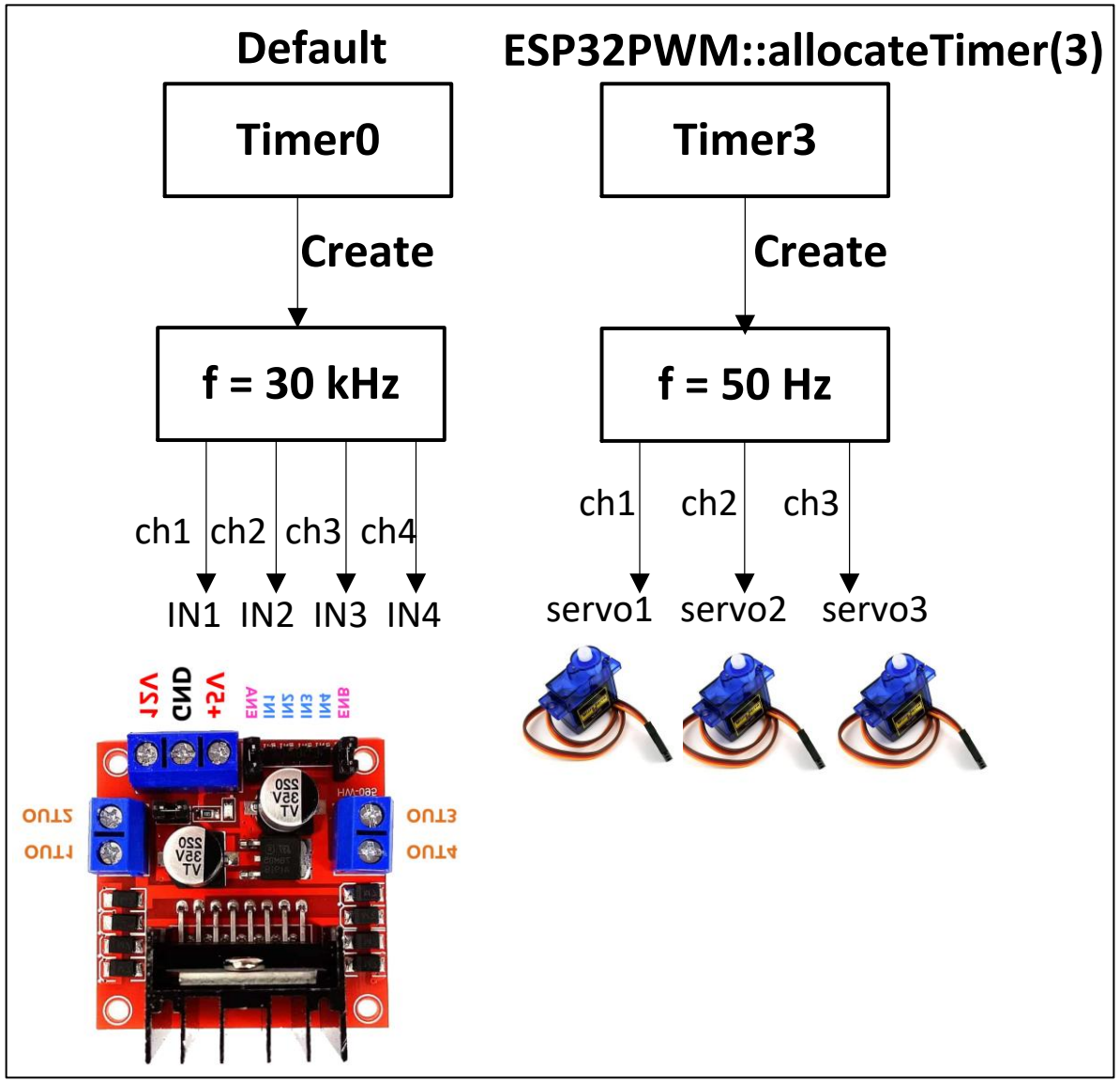
Chapter4: Two motors and a servo motor by “o 255 255” and “s 45”



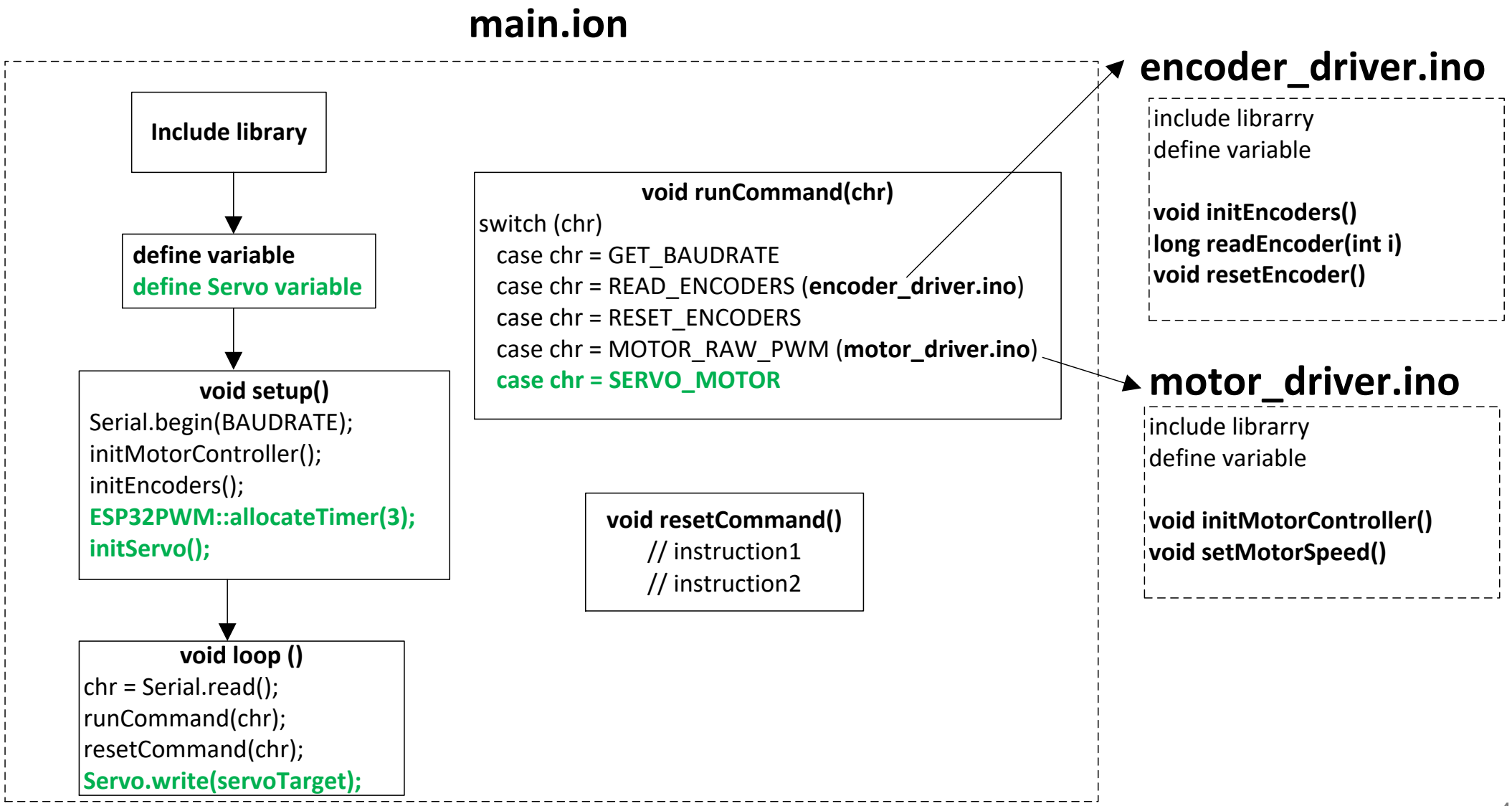
Chapter4: Two motors and a servo motor



ในกรณีมี PWM ที่มากกว่า 1 ค่าขึ้นไป ควรสร้าง allocateTime() เพื่อไม่ให้ความถี่ PWM ทับซ้อนกัน !!!



Flowchart: Two motors and a servo motor



วิธีการเขียนโค้ด: นำโค้ดเดิมทั้งหมดจาก chapter3 มา save as และตั้งชื่อเป็นไฟล์ใหม่ จากนั้นเขียนโค้ดเพิ่มเติมดังนี้

ไฟล์หลัก

twomotor_servo_full_v1.ino | commands.h | encoder_driver.h | encoder_driver.ino | motor_driver.h | motor_driver.ino

```
1   #define BAUDRATE 115200 // กำหนดความเร็ว Serial
2   #include <string.h> // ไลบรารีสำหรับจัดการ string
3   #include "commands.h" // ไฟล์ header สำหรับคำสั่ง command
4   #include "motor_driver.h" // ไฟล์ header สำหรับมอเตอร์
5   #include "encoder_driver.h" // ไฟล์ header สำหรับ encoder
6
7   /// Servo motor
8   #include <ESP32Servo.h>
9   #define SERVO_PIN 12
10  #define SERVO_MOTOR 's'
11  Servo myservo;
12  volatile int servoTarget = 90;
13
```

เพิ่มเติม 1

ไฟล์หลัก

twomotor_servo_full_v1.ino | commands.h | encoder_driver.h | encoder_driver.ino | motor_driver.h | motor_driver.ino

```
55    case MOTOR_RAW_PWM:
56      // ตั้งความเร็วมอเตอร์ตาม arg1 และ arg2
57      setMotorSpeeds(arg1, arg2);
58      Serial.println("OK");
59      break;
60    case SERVO_MOTOR: // Servo motor
61      if (arg1 >= 0 && arg1 <= 180) {
62        servoTarget = arg1;
63        Serial.println("OK");
64      }
65      break;
66    default:
67      // command ไม่ถูกต้อง
68      Serial.println("Invalid Command");
69      break;
70  }
```

เพิ่มเติม 2

ไฟล์หลัก

twomotor_servo_full_v1.ino | commands.h | encoder_driver.h | encoder_driver.ino | motor_driver.h | motor_driver.ino

```
72
73  void setup() { // ฟังก์ชัน setup() รันครั้งเดียวตอนเริ่มต้น
74    Serial.begin(BAUDRATE); // เริ่ม Serial
75    initMotorController();  // เตรียมมอเตอร์
76    initEncoders();         // เตรียม encoder
77
78    // Servo motor setup
79    ESP32PWM::allocateTimer(3); //คำสั่งนี้เป็นการบอก Library ESP32Servo ว่า "ให้ไปใช้ Timer ตัวที่ 3 เท่านั้น"
80    myservo.setPeriodHertz(50);
81    myservo.attach(SERVO_PIN, 500, 2400);
82    myservo.write(servoTarget);
83
84  }
85
```

เพิ่มเติม 3

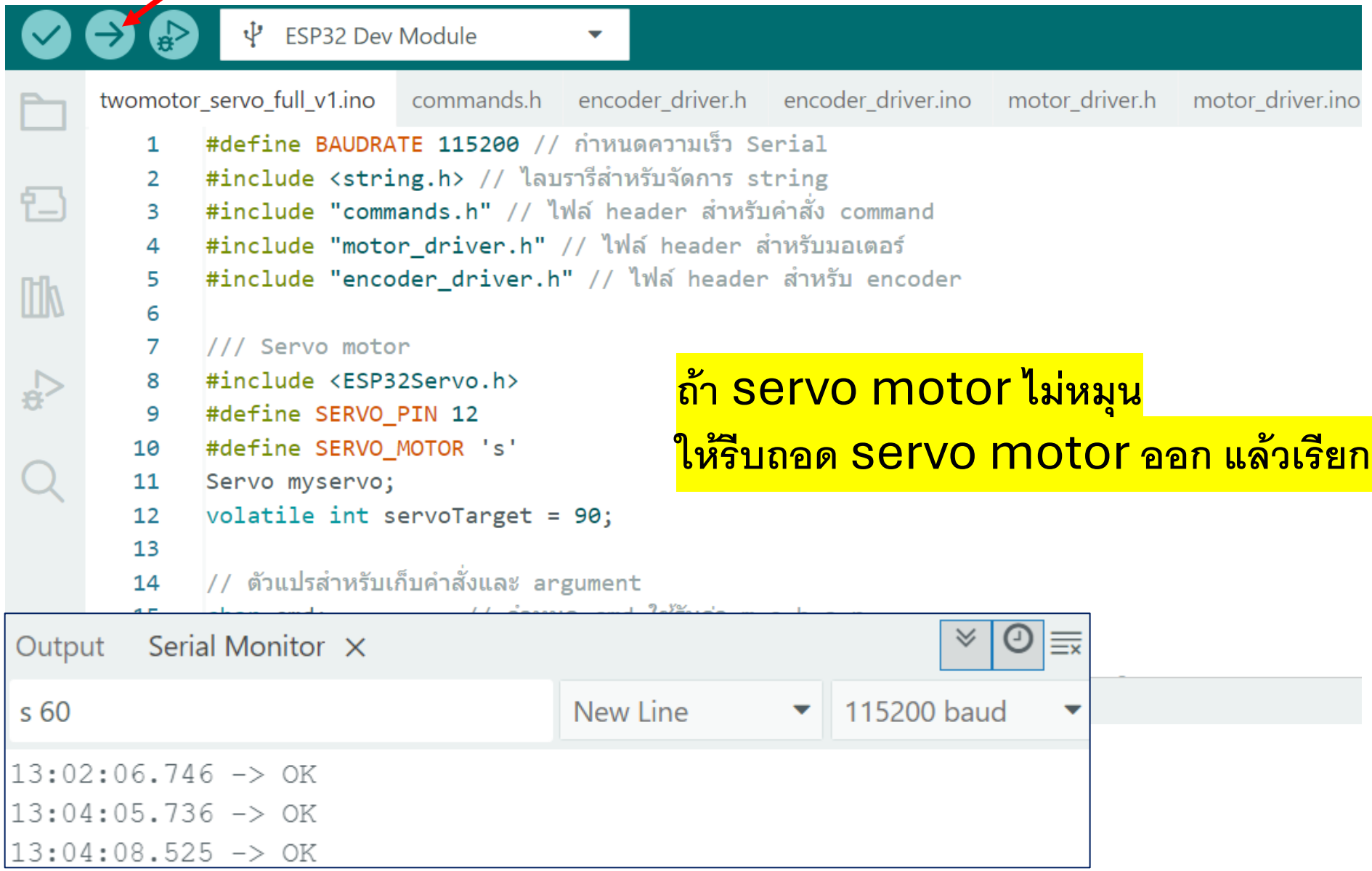
ไฟล์หลัก

twomotor_servo_full_v1.ino | commands.h | encoder_driver.h | encoder_driver.ino | motor_driver.h | motor_driver.ino

```
95      else if (chr == ' ') {  // ถ้าเจอ space แยก argument
96        if (arg == 0) arg = 1;                  // เริ่ม argument 1
97        else if (arg == 1) {
98          argv1[argIndex] = '\0';               // จบ argument 1
99          arg = 2;                               // เริ่ม argument 2
100         argIndex = 0;                          // รีเซ็ต index
101       }
102     }
103     else { // เก็บตัวอักษร
104       if (arg == 0) cmd = chr;                  // ถ้า arg=0 ถือว่าเป็น command
105       else if (arg == 1 && argIndex < 15) argv1[argIndex++] = chr; // เก็บ argv1
106       else if (arg == 2 && argIndex < 15) argv2[argIndex++] = chr; // เก็บ argv2
107     }
108   }
109   myservo.write(servoTarget);// servo work
110 }
111
```

เพิ่มเติม 4

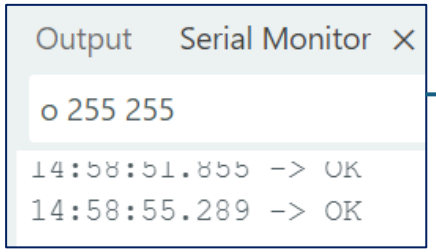
Run program



ถ้า servo motor ไม่หมุน
ให้รีบถอด servo motor ออก แล้วเรียกอาจารย์ !!!

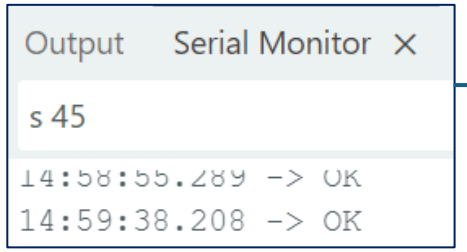
Please type these commands, OK?

1



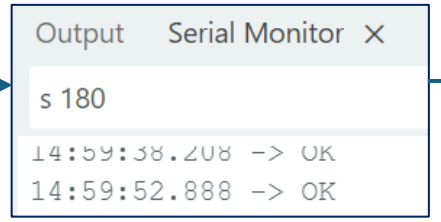
Car Forward

2



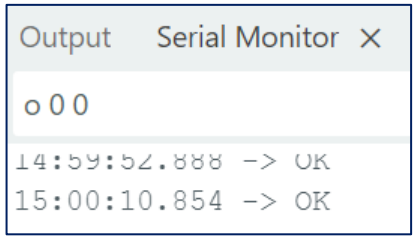
Rotate 45

3



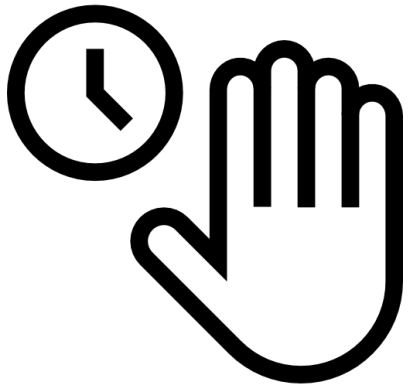
Rotate 180

4



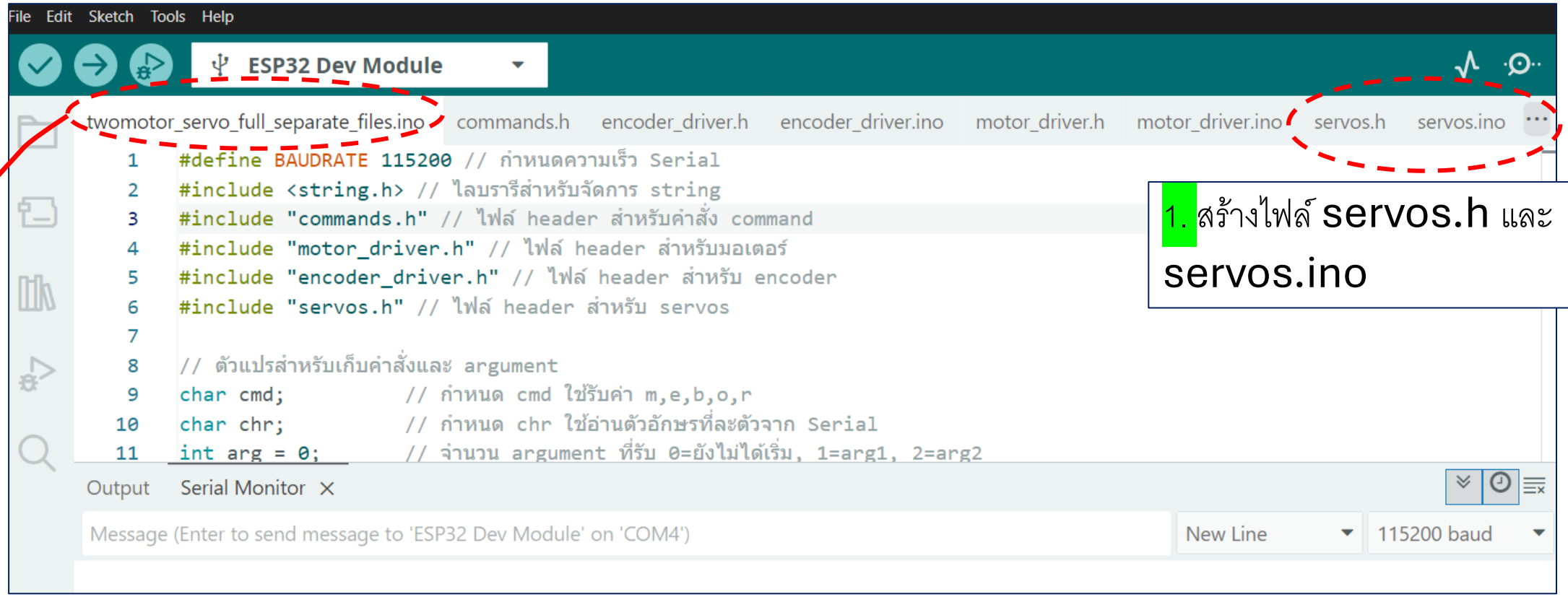
Car stop

ถ้าผ่านทุกเงื่อนไข หยุดตรวจรอให้คะแนนและรอเพื่อน ๆ !!!



Content

Chapter5: From chapter4, split it into files “servos.h” and “servos.ino”



1. สร้างไฟล์ servos.h และ servos.ino

2. ลบทุกอย่างที่เกี่ยวข้องกับ servo motor ใน main ไฟล์ เพราะจะย้ายทุกอย่างที่เกี่ยวข้องไปไฟล์ servos.h และ servos.ino

commands.h

```
#ifndef COMMANDS_H //Header Guard
#define COMMANDS_H //Header Guard

#define GET_BAUDRATE    'b'
#define READ_ENCODERS   'e'
#define MOTOR_RAW_PWM   'o'
#define RESET_ENCODERS  'r'
#define SERVO_MOTOR 's'
#define LEFT    0
#define RIGHT   1

#endif  // COMMANDS_H
```

Add

servos.h

Add

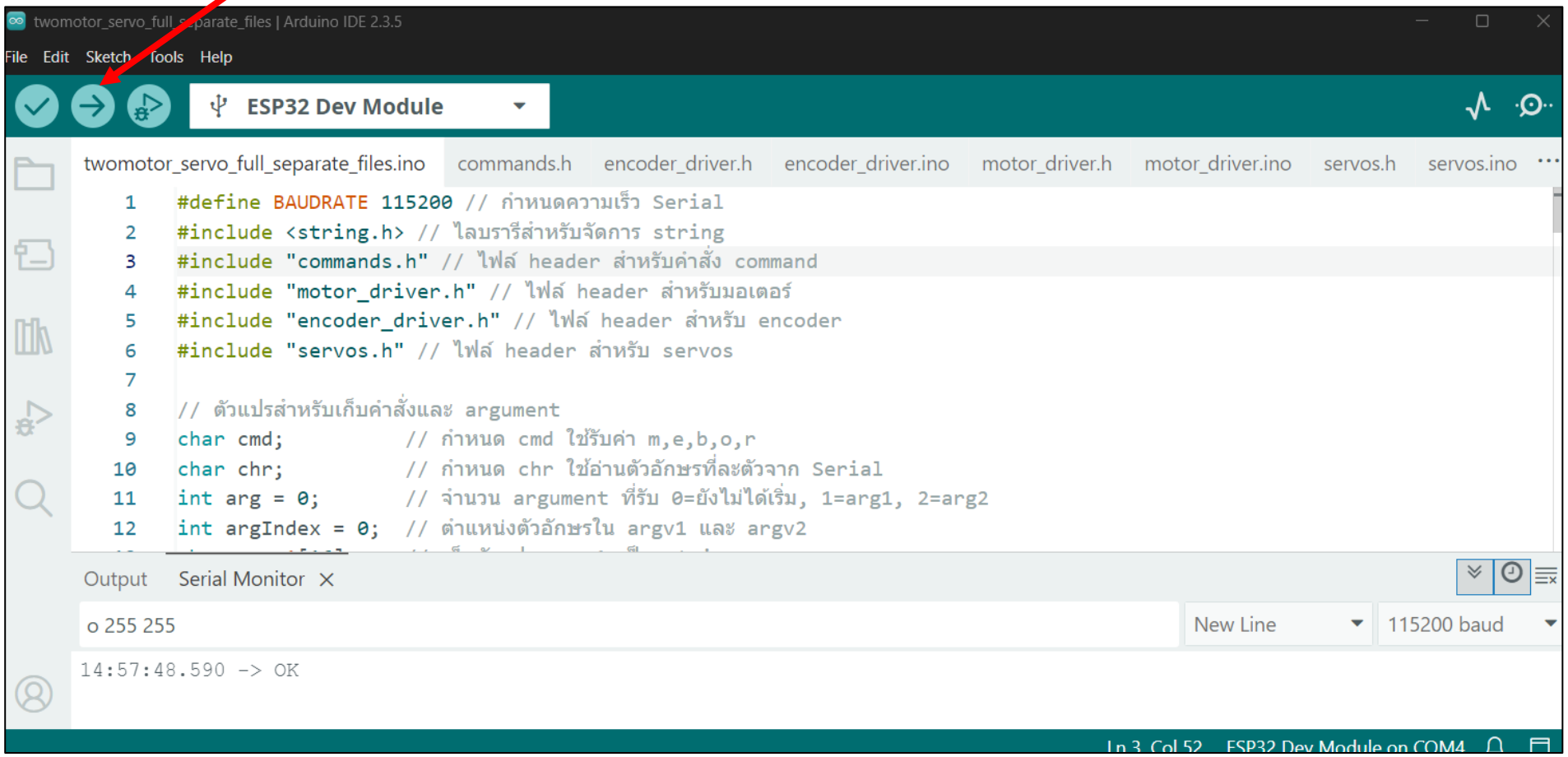
```
#define SERVO_PIN 12
 // ประกาศว่า servoTarget อยู่ไฟล์อื่น
extern volatile int servoTarget;

void initServos();
void Servo_do(float degree);
```

servos.ino

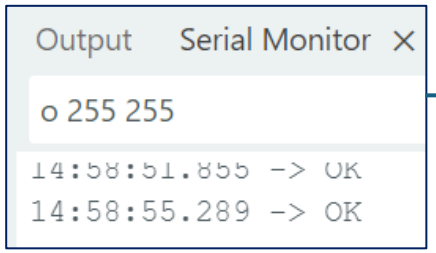
Add

```
#include "servos.h"
#include <ESP32Servo.h>
volatile int servoTarget = 90;
Servo myservo;

void initServos() {
  ESP32PWM::allocateTimer(3);
  myservo.setPeriodHertz(50);
  myservo.attach(SERVO_PIN, 500, 2400);
  myservo.write(servoTarget);
}

void Servo_do(float degree) {
  myservo.write(degree);
}
```

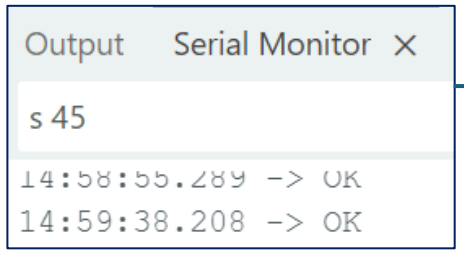
Run program



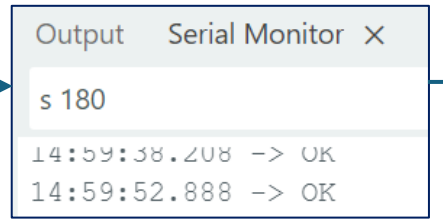
1



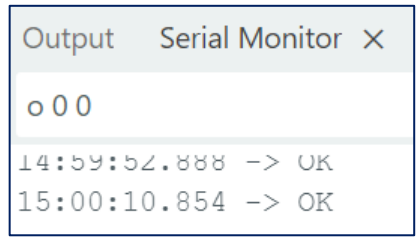
2



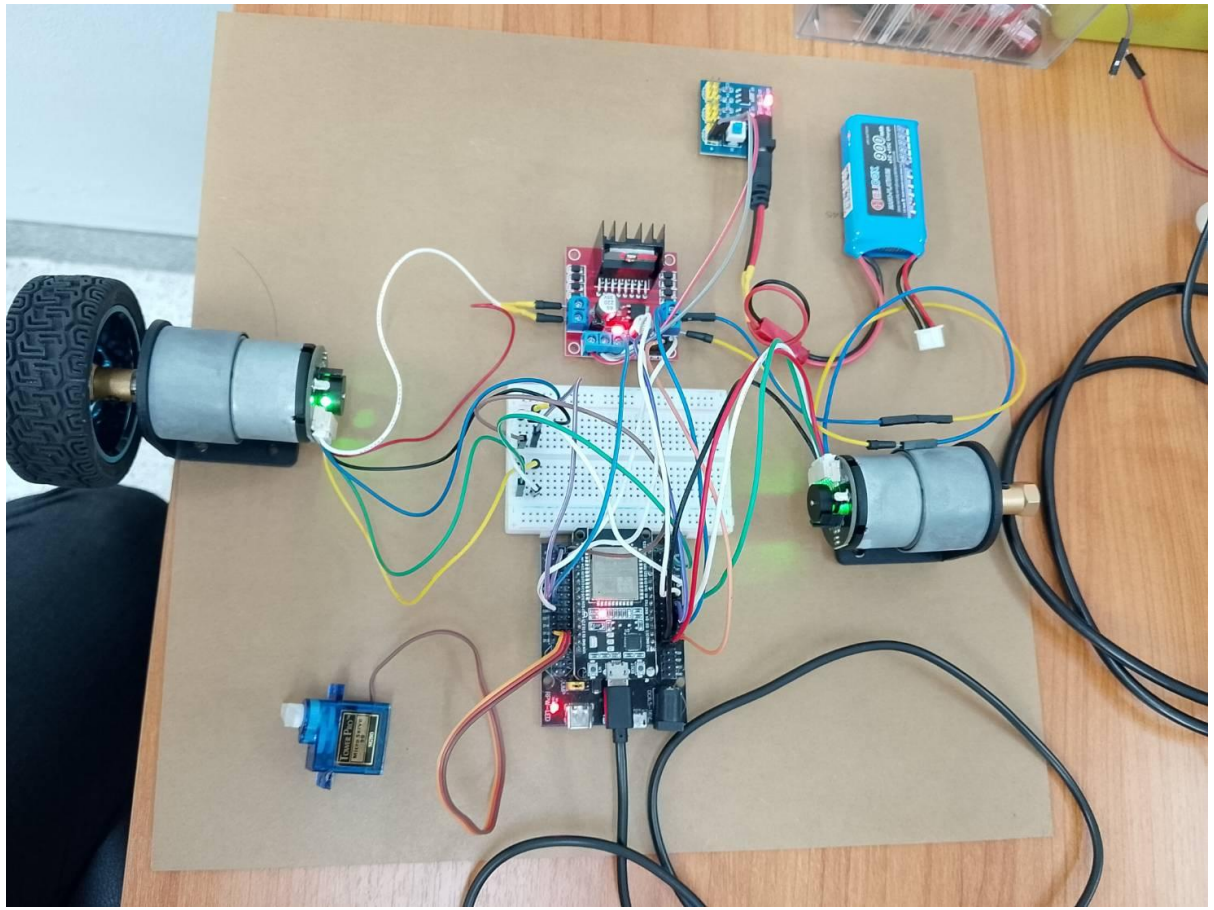
3



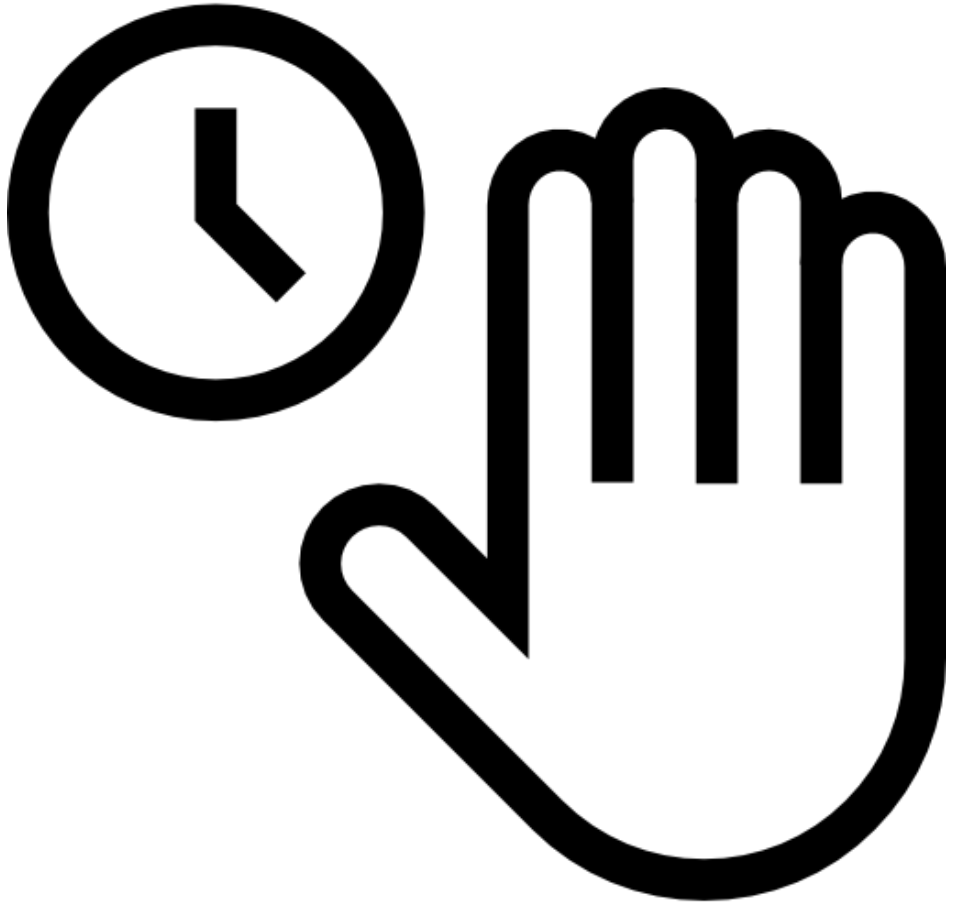
4



Homework 2: please show your control by keyboard and show the flowchart block diagram as you understand it.

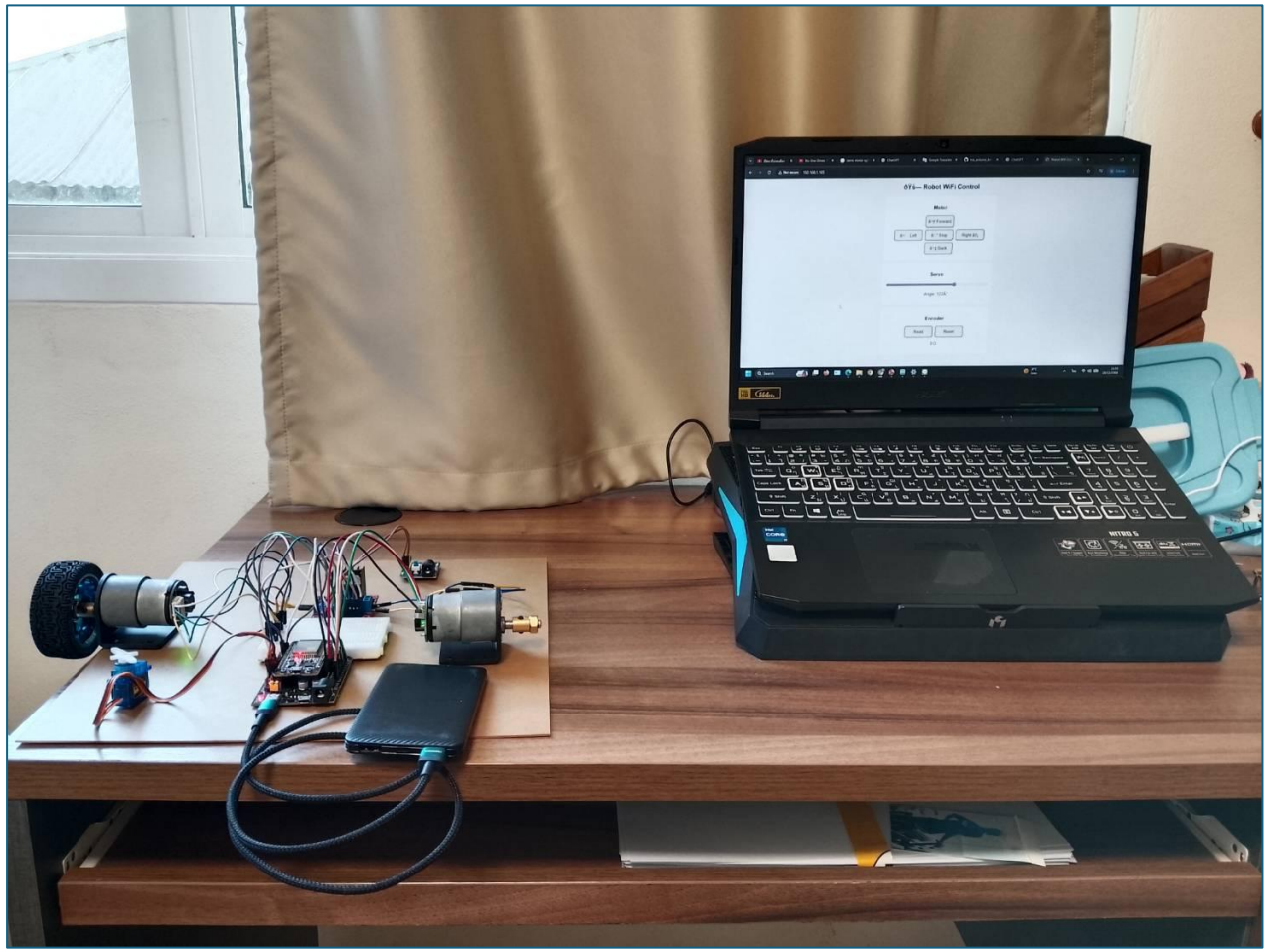


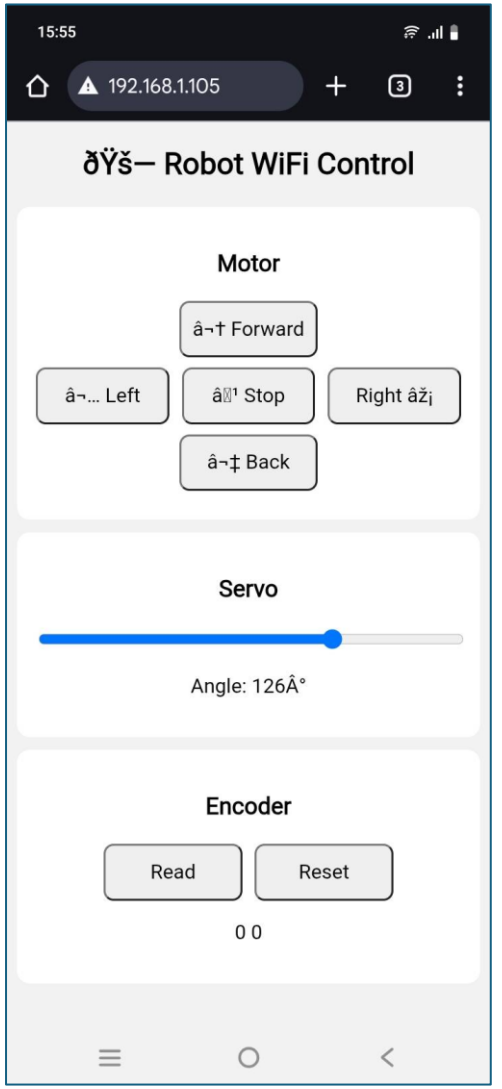
หยุดตรวจรอให้คะแนนและรอเพื่อนๆ !!!



Content

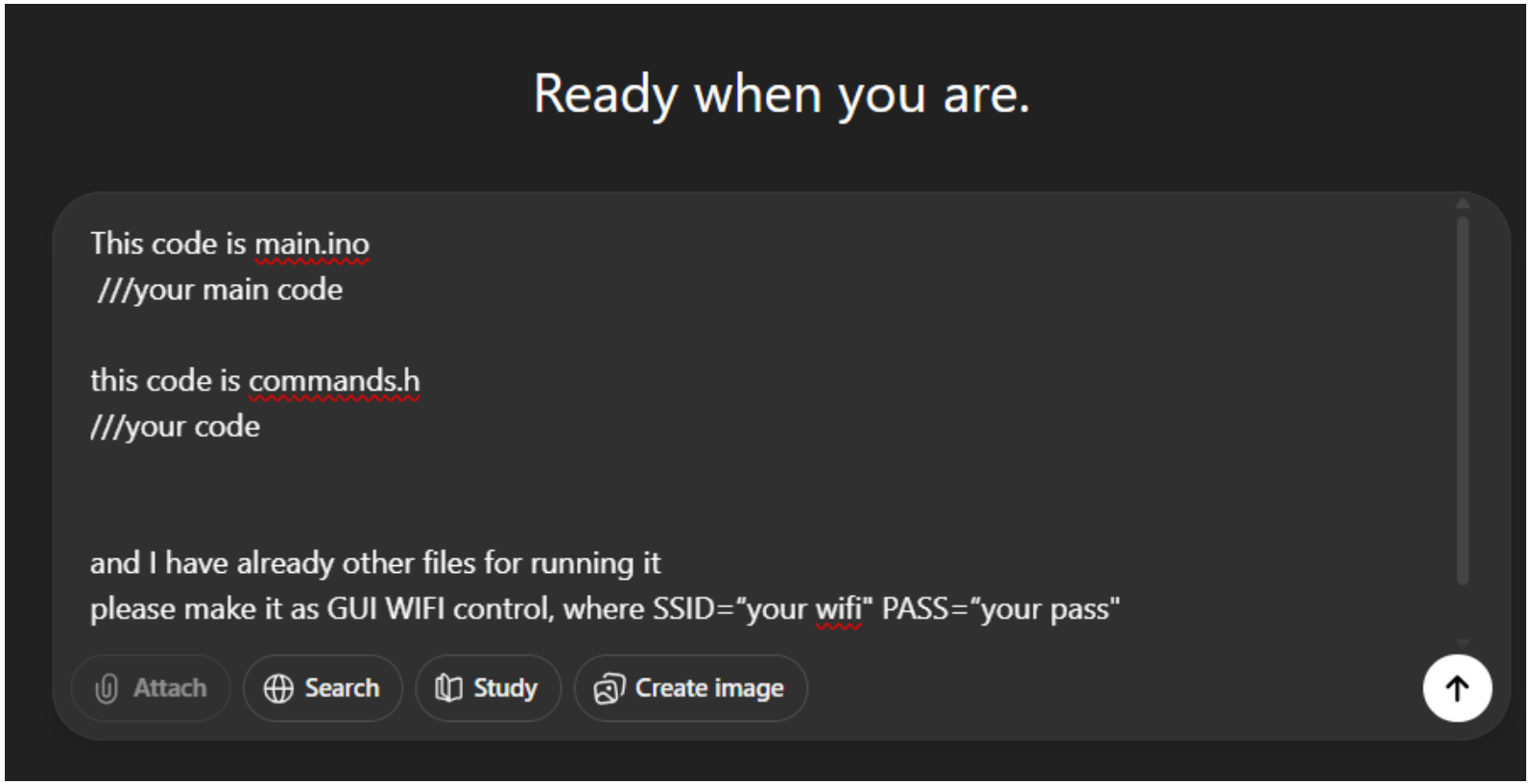
Chapter6: From chapter5, it is developed as GUI WIFI control





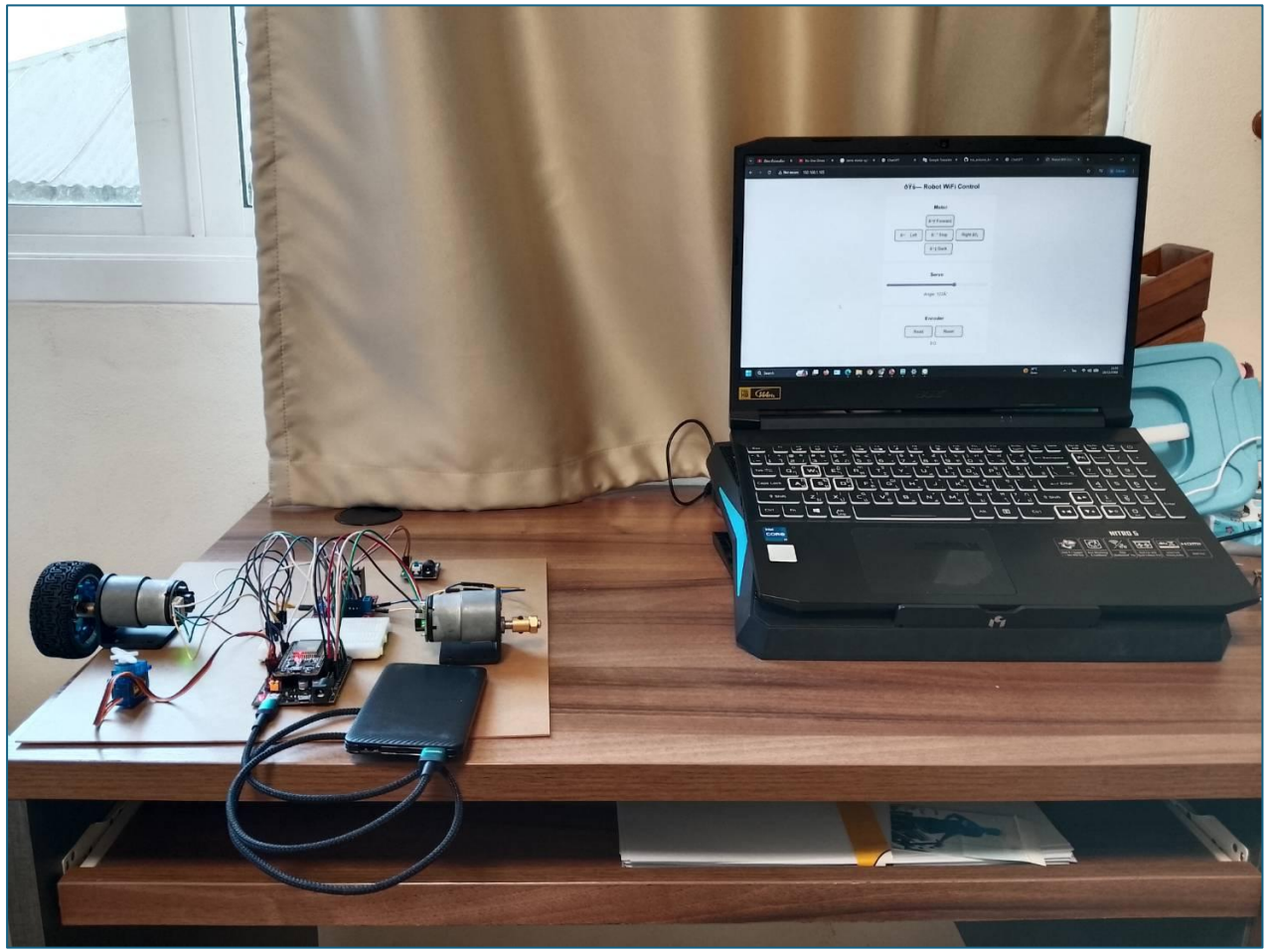
1. Open “ChatGPT” or others

2. Typing message and add your code as following



3. it will generate a new file for main.ion

Homework 3: please show your control by wifi and show the flowchart block diagram as you understand it.



หยุดตรวจรอให้คะแนนและรอเพื่อนๆ !!!