

หุ่นยนต์ชุดนี้ เหมาะสำหรับนำไปศึกษาและเรียนรู้ทั้ง แม็คคานิกส์ อิเล็กทรอนิกส์ และการเขียนโปรแกรมคอมพิวเตอร์

สิ่งที่ผู้เรียนจะได้รับ

1. ได้เรียนรู้เกี่ยวกับการสั่งงานหุ่นยนต์ผ่านแอปพลิเคชันบนโทรศัพท์มือถือ
2. ได้เรียนรู้เกี่ยวกับการเขียนโปรแกรม ARDUINO
3. ได้เรียนรู้เกี่ยวกับการเขียนแอปพลิเคชันที่ใช้กับโทรศัพท์มือถือระบบ ANDROID
4. ได้เรียนรู้และพัฒนาการสั่งงานผ่านระบบ Internet of Things (IoT)

ข้อมูลทางค่านเทคนิค

- ไซแหล่งจ่ายไฟขนาด 6 โวลต์ดีซี
- กินกระแสสูงสุดประมาณ 500 มิลลิแอมป์
- สามารถใช้แอปพลิเคชันบนโทรศัพท์มือถือควบคุมหุ่นยนต์ได้ ผ่านระบบไวไฟ
- มี LED จำนวน 3 ดวง สำหรับไฟแสดงผล
- แอปพลิเคชันที่จัดทำไว้ ใช้ได้บนโทรศัพท์มือถือระบบ ANDROID เท่านั้น
- ขนาดแผ่นวงจรพิมพ์ของวงจรควบคุม : 2.54 นิ้ว x 3.09 นิ้ว

(1) วงจรควบคุมหุ่นยนต์

1.การทำงานของวงจร

หัวใจของวงจรนี้อยู่ที่ IC1 ซึ่งเป็นโมดูล NODE MCU V.2 (รูปที่ 1) ในสภาวะปกติที่ยังไม่มีการสั่งงานผ่านทางแอปพลิเคชันโทรศัพท์มือถือ IC1 จะยังไม่มีการสั่งให้หุ่นยนต์ทำงาน แต่เมื่อไรก็ตามที่ทำการสั่งงานผ่านทางแอปพลิเคชัน เช่น เมื่อสั่งเดินหนา IC1 จะทำการสั่งให้มอเตอร์วิ่งไปข้างหน้า โดยส่งแรงดันออกทางขา D5 และ D7 ส่วนขา D6 และ D8 จะไม่ส่งแรงดันออกมา มอเตอร์จึงหมุนไปข้างหน้า แต่ถ้าทำการสั่งให้ถอยหลัง IC1 จะทำการสั่งให้มอเตอร์วิ่งไปด้านหลัง เป็นต้น

2.การประกอบวงจร

ในการลงอุปกรณ์นั้นจะแบ่งแผ่นวงจรพิมพ์ออกเป็น 2 แผ่น ได้แก่ แผ่น FK1130 จะเป็นแผ่นวงจรควบคุมและ BR002 จะเป็นแผ่นติดตั้งมอเตอร์และล้อ รวมทั้งรังจาดันด้วย รูปการลงอุปกรณ์ของแผ่น FK1130 แสดงไว้ในรูปที่ 2 ในการประกอบวงจร ควรจะเริ่มจากอุปกรณ์ที่มีความสูงที่น้อยที่สุดก่อน เพื่อความสวยงามและการประกอบที่ง่าย โดยให้เริ่มจากไดโอดตามด้วย ตัวต้านทานและไอคอนสูงไปเรื่อยๆ สำหรับอุปกรณ์ที่มีขั้วต่างๆ เช่น ไดโอด,คาปาซิเตอร์แบบอิเล็กโทรไลต์และทรานซิสเตอร์ เป็นต้น ควรใช้ความระมัดระวังในการประกอบวงจร ก่อนการใส่อุปกรณ์เหล่านี้จะต้อง ให้ขั้วที่แผ่นวงจรพิมพ์กับตัวอุปกรณ์ให้ตรงกัน เพราะถาหากใส่กลับขั้วแล้ว อาจจะทำให้อุปกรณ์หรือวงจรเสียหายได้ วิธีการดูขั้วและการใส่อุปกรณ์นั้นได้แสดงไว้ในรูปที่ 3 แลว สำหรับขา IDE นั้นให้ทำการกดขาสี่ทองคานสั้นลงในเสมอกับตัวพลาสติกสีดำก่อน จึงนำไปใส่ลงบนแผ่นวงจรพิมพ์ได้ ซึ่งแสดงอยู่ในรูปที่ 4 ในการบัดกรีให้ใช้หัวแร้งขนาดไม่เกิน 40 วัตต์ และใช้ตะกั่วบัดกรีที่มีอัตราส่วนของดีบุกและตะกั่วอยู่ระหว่าง 60/40 รวมทั้งจะต้องมีน้ำยาประสานอยู่ภายในตะกั่วด้วย หลังจากที่ได้ใส่อุปกรณ์และบัดกรีเรียบร้อยแล้ว ให้ทำการตรวจสอบความถูกต้องอีกครั้งหนึ่ง เพื่อให้เกิดความมั่นใจแก่ตัวเราเอง แต่ถ้าเกิดใส่อุปกรณ์ผิดตำแหน่ง ควรใช้ที่ดูดตะกั่วหรือลวดจับตะกั่ว เพื่อป้องกันความเสียหายที่อาจจะเกิดกับลายวงจรพิมพ์ได้

สำหรับแผ่น BR002 ให้ทำการลงอุปกรณ์ตามคู่มือในหัวข้อ (2) ตัวหุ่นยนต์ ข้อ 5

3.การทดสอบ

เมื่อประกอบวงจรทั้งสองแผ่นเสร็จเรียบร้อยแล้ว ให้ทำการใส่ถ่านอัลคาไลน์ ขนาด AA จำนวน 4 ถ่าน ลงบนรังจาดัน จากนั้นให้ทำการจับคู่ระหว่างแอปพลิเคชันบนมือถือกับหุ่นยนต์ ทดลองสั่งให้หุ่นยนต์เดินหนา ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา ผ่านแอปพลิเคชันบนโทรศัพท์มือถือ

ROBOT CONTROLLED BY MOBILE PHONE

หุ่นยนต์ควบคุมด้วยโทรศัพท์มือถือ

CODE 1130

LEVEL 2

This robot is suitable for studying and learning about mechanics electronics and computer programming.

Benefits of learning

1. Learn about how to operate a robot through an application on a mobile phone.
2. Learn about ARDUINO programming
3. Learn about writing applications for Android mobile phones.
4. Learn and develop work orders through the Internet of Things (IoT) system.

Technical data

- Power supply : 6VDC.
- Electric current consumption : 500mA.
- Applications can be used on mobile phone to control robot via Wi-Fi.
- There are 3 LEDs for display.
- Application made only available on ANDROID mobile phone.
- IC board dimension : 2.54 in x 3.09 in.

(1) ROBOT CONTROL CIRCUIT

1.How does it work

The heart of this circuit is the IC1, which is a NODE MCU V.2 module as shown in Figure 1. In a normal condition that has not been activated via a mobile phone application, IC1 will not have the robot command to operate. But whenever doing a job via the application, for example, when forwarding, IC1 will order the motor to run forward by sending out the voltage through pins D5 and D7. The pins D6 and D8 will not send out the voltage. The motor rotates forward, but if instructing to reverse, IC1 will instruct the motor to run backwards etc.

2.Circuit Assembly

FK1130 has 2 PCB consisting of control circuit and robot mounting plate i.e. chassis, gear motor and wheels plus a battery compartment.

The FK1130 board assembly of components is shown in Fig. 2. For good looking and easy assembly, the least height components should be first installed-starting with diode followed by resistor, and keep chasing the height. An important thing is that diodes, electrolyte capacitors, and transistors shall be carefully assembled before mounting them onto their right anode/cathode of the IC board otherwise it might cause damage to the components or the circuit. Configuration of the anode and the cathode is shown in Fig 3. For the IDE pin, press down on the golden side of the short pin to be the black plastic first. Can be put on the printed circuit board, as shown in Fig 4. Use the soldering iron/gun not exceeding 40 watts and the solder of tin-lead 60:40 with flux within. Recheck the correctness of installation after soldering. In case of wrong position, just use lead absorber or lead extractor wire to avoid probable damage to the IC.

The BR002 circuit (robot mounting plate) is to be assembled as shown in Topic (2) ROBOT BODY no.5.

3.Testing

When the two circuits have been completed, put 4 AA size alkaline batteries into the battery holder, then pairing between the mobile application and the robot. Order the robot to go forward, backward, turn left and turn right through the application on mobile phone.

Figure 1. Robot Controlled by Mobile Phone Circuit

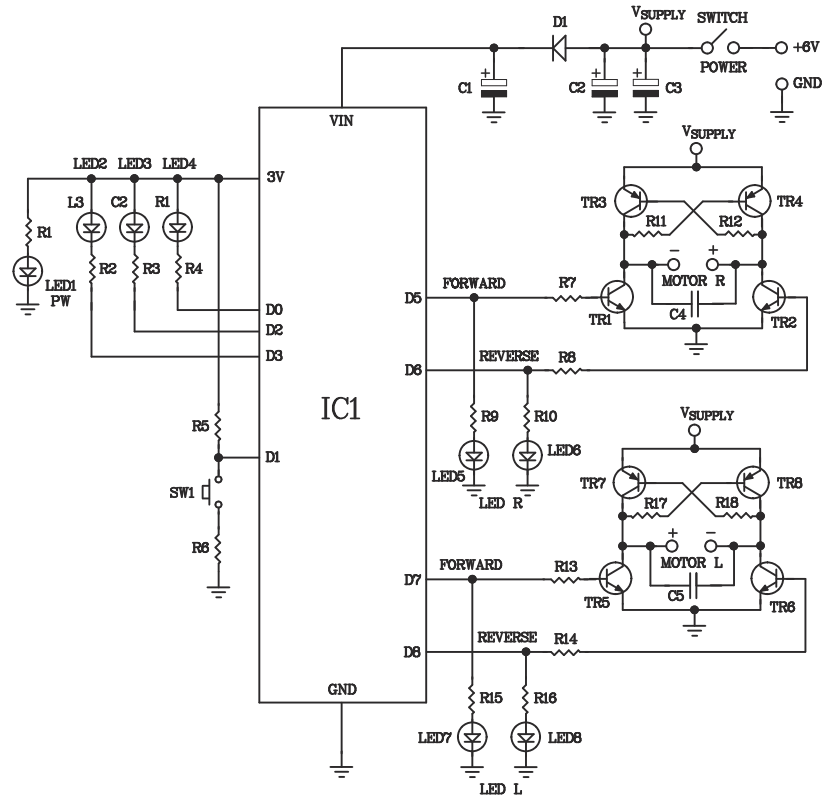
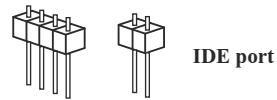
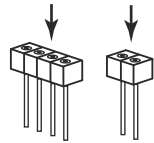


Figure 4. IDE Port Assembling To The FK1130 Circuit Board.



Press the pin of IDE port to be level with the black plastic.



Install IDE port to PC-board and soldering.

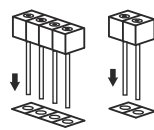
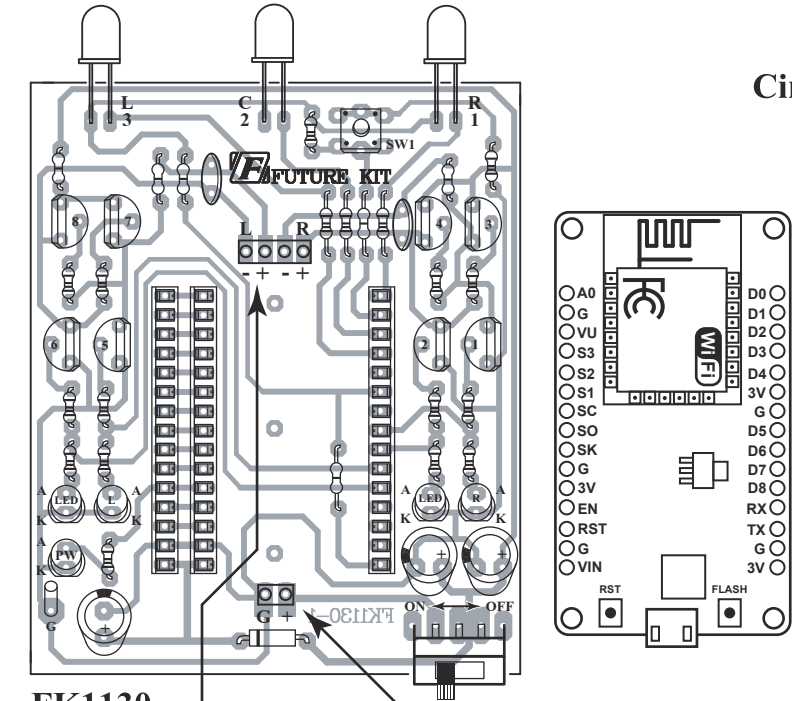
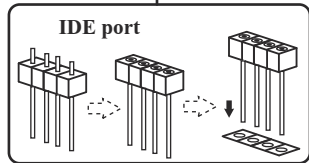


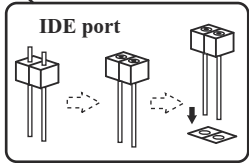
Figure 2. Circuit Assembling



FK1130



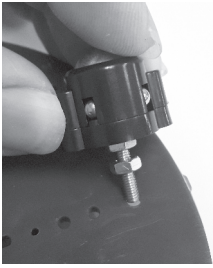
NO.1



(2) ROBOT BODY

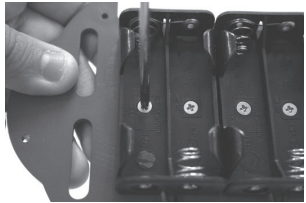
Assembling Steps of the Body set.

①



Mini Caster
ชุดล้อหลัง

Fix the mini caster wheel set to the Body, by using a bolt as a holder.
ประกอบชุดล้อหลังเข้ากับตัวหุ่นยนต์ทางด้านหลัง โดยใช้น็อตที่มากับชุดล้อหลัง เป็นตัวยึด



Flat head nut 2.5x10 and NUT M2.5
น็อตหัวแปปเปอร์ตัวผู้ 2.5x10 และตัวเมีย M2.5

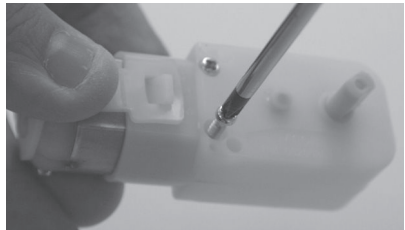
②

Install battery holder to robot body with flat head nut 2.5x10 and nut M2.5.
ประกอบถังถ่านขนาด AA ทั้งสองตัวเข้ากับตัวหุ่นยนต์โดยใช้น็อตหัวแปปเปอร์ ตัวผู้ 2.5x10 และ ตัวเมีย M2.5 เป็นตัวยึด

③



Insert the electric wire of battery holder into robot body.
สอดสายไฟของถังถ่านขึ้นมาด้านบน

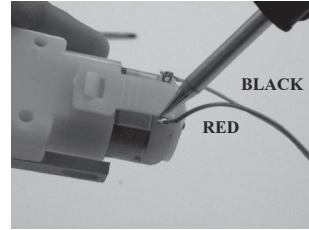
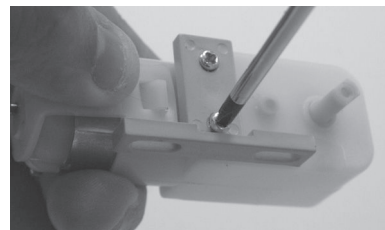


Take off the both screw of motor gear and then mount the motor lock.

Secure with the both screw of motor gear.

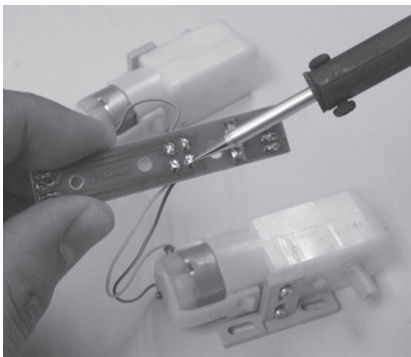
ทำการถอดน็อตของมอเตอร์เกียร์ออก จากนั้นให้ทำการยึดตัวล็อกเข้ากับมอเตอร์ โดยใช้น็อตที่ถอดออกมาจากตัวมอเตอร์เกียร์เป็นตัวยึด

④



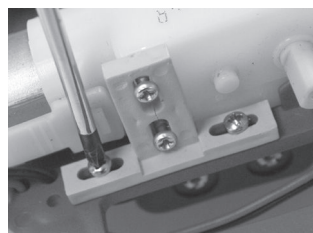
Solder electric wire at motor pole with red wire solders at left hand side and black wire solders at right hand side.
บัดกรีสายไฟที่ขั้วของมอเตอร์ โดยให้หันด้านท้ายของมอเตอร์เข้าหาตัวแล้วบัดกรีสายสีแดงทางด้านซ้ายและสายสีดำที่ด้านขวา

⑤



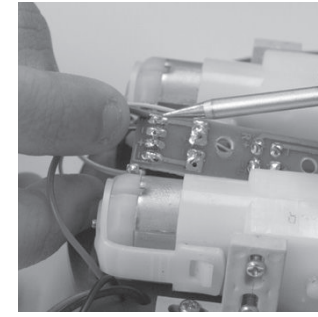
Solder motor wire to BR002 PC-board. Red wire is positive pole and black wire is negative pole. Character "L" is left motor gear and "R" is right motor gear.

บัดกรีสายมอเตอร์เข้ากับแผงวงจรพิมพ์ BR002 โดยบัดกรีที่ตำแหน่ง MOTOR สายสีแดง ให้บัดกรีที่ ตำแหน่งบวกและสายสีดำบัดกรีที่ ตำแหน่งลบ ส่วนตัวอักษร L คือ มอเตอร์เกียร์ทางด้านซ้ายและตัวอักษร R คือ มอเตอร์เกียร์ทางด้านขวา เมื่อบัดกรีสายไฟเรียบร้อยแล้ว



Mount motors, each with two #4 x 1/4" screws
ยึดมอเตอร์กับตัวหุ่นยนต์ โดยใช้สกรูขนาด 4x1/4

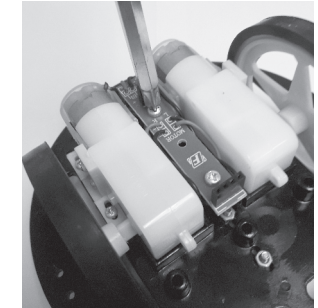
⑥



⑦

Solder battery holder wire to BR002 PC-board at B1 and B2. Red wire is positive pole and Black is negative pole.

บัดกรีสายถังถ่านเข้ากับแผงวงจรพิมพ์ BR002 โดยบัดกรีที่ตำแหน่ง B1 และ B2 สายสีแดงให้ทำการบัดกรีที่ตำแหน่งบวกและสายสีดำให้ทำการบัดกรีที่ตำแหน่งลบ

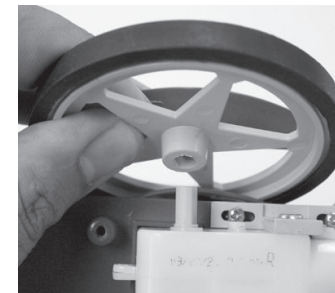


Screw 2x1/4
สกรู 2x1/4

⑧

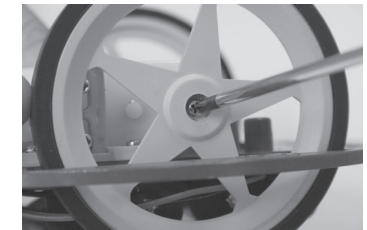
Mount BR002 PC-board into body robot and secure them with two #2 x 1/4" screws.
ยึดแผงวงจรพิมพ์ BR002 กับตัวหุ่นยนต์ โดยใช้สกรูขนาด 2x1/4

⑨



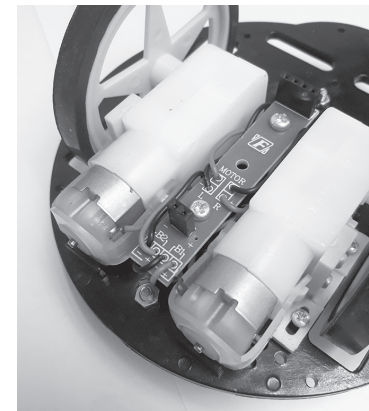
Install the wheels onto the shaft of the gear motors and secure them with the remaining two #4 x 1/4" pointy screws.

นำล้อหุ่นยนต์มาสวมเข้ากับแกนมอเตอร์เกียร์ จากนั้นให้ใช้สกรูขนาด 4x1/4 ยึดที่รูตรงกลางของล้อ



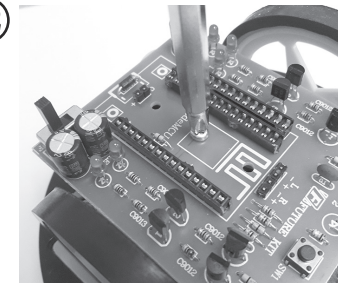
Screw 4x1/4
สกรู 4x1/4

⑩



Body robot is completely installed.
ตัวหุ่นยนต์ที่ประกอบเสร็จเรียบร้อยแล้ว

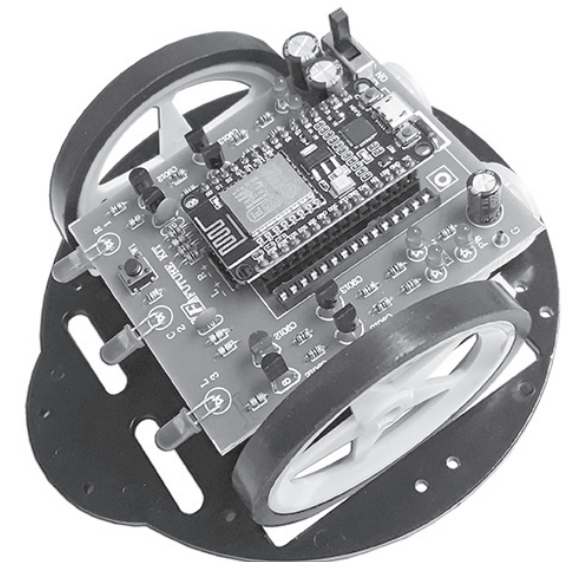
⑫



ทำการยึดแผ่นควบคุมด้วยสกรูขนาด 4x3/4
Secure control board with two #4 x 3/4" screws.

⑬

The robot is prompt working and playing.
หุ่นยนต์ที่ประกอบเสร็จเรียบร้อยแล้ว



⑪



Install the control board into robot body.
นำแผ่นควบคุมมาเสียบลงที่ตัวหุ่นยนต์ ซึ่งได้ติดตั้งแผ่น BR002 เรียบร้อยแล้ว

โค้ดควบคุมหุ่นยนต์ SOURCE CODE OF ROBOT
(สามารถดาวน์โหลดโค้ดนี้ได้ที่ www.futurekit.com)
(This code can be downloaded at www.futurekit.com)

```
1 | #define M2_A D5
2 | #define M2_B D6
3 | #define M1_A D7
4 | #define M1_B D8
5 |
6 | #define led_r D0
7 | #define led_ct D2
8 | #define led_l D3
9 |
10 | #define sw_setting D1
11 |
12 | #include <ESP8266WiFi.h>
13 | #include <WiFiClient.h>
14 | #include <ESP8266WebServer.h>
15 | #include <EEPROM.h>
16 |
17 | void HTTP_handleRoot(void);
18 | String command; //String to store app command state.
19 | int speedCar = 800; // 400 - 1023.
20 | int speed_Coeff = 3;
21 | const char* password = "12345678";
22 | char t_ssid;
23 | String ssid2;
24 |
25 | ESP8266WebServer server(80);
26 |
27 | #define led_r_off digitalWrite(led_r,HIGH);
28 | #define led_r_on digitalWrite(led_r,LOW);
29 | #define led_l_off digitalWrite(led_l,HIGH);
30 | #define led_l_on digitalWrite(led_l,LOW);
31 | #define led_ct_off digitalWrite(led_ct,HIGH);
32 | #define led_ct_on digitalWrite(led_ct,LOW);
33 |
34 | #define add_pwm 40
35 | #include "inc_control.h"
36 |
37 | void w_eeprom()
38 | {
39 |   for (int i = 10; i < 20; ++i){EEPROM.write(i, 0);} // clearing eeprom
40 |   for (int i = 0; i < ssid2.length(); ++i){EEPROM.write(i, ssid2[i]);}
41 |   EEPROM.commit();
42 |   delay(10);
43 | }
44 |
45 | void r_eeprom()
46 | {
47 |   t_ssid = EEPROM.read(0);
48 |   for (int i = 10; i < 20; ++i) { ssid2 += char(EEPROM.read(i)); }
49 | }
50 |
51 | void setup()
52 | {
53 |   pinMode(M1_A,OUTPUT);
54 |   pinMode(M1_B,OUTPUT);
55 |   pinMode(M2_A,OUTPUT);
56 |   pinMode(M2_B,OUTPUT);
57 |
58 |   pinMode(led_l, OUTPUT);
59 |   pinMode(led_r, OUTPUT);
60 |   pinMode(led_ct, OUTPUT);
61 |   pinMode(sw_setting, INPUT_PULLUP);
62 |
63 |   led_r_off;
64 |   led_l_off;
65 |   led_ct_off;
66 |   EEPROM.begin(512); //Initialize EEPROM
67 |   Serial.begin(115200);
```

```
68 | r_eeprom();
69 |
70 | chk_sw();
71 | IPAddress apIP(192, 168, 4, 1);/--set ip
72 | IPAddress subIP(255, 255, 255, 0);
73 |
74 | WiFi.mode(WIFI_AP);
75 | WiFi.softAPConfig(apIP, apIP, subIP);
76 | WiFi.softAP(ssid2, password);
77 |
78 | IPAddress myIP = WiFi.softAPIP();
79 | Serial.print("AP IP address: ");
80 | Serial.println(myIP);
81 |
82 | server.on ( "/", HTTP_handleRoot );
83 | server.onNotFound ( HTTP_handleRoot );
84 | server.on("/led", HTTP_GET, light_ct);
85 | server.on("/mo_fo", HTTP_GET, m_forward);
86 | server.on("/mo_re", HTTP_GET, m_reverse);
87 | server.on("/mo_le", HTTP_GET, m_turn_left);
88 | server.on("/mo_ni", HTTP_GET, m_turn_right);
89 | server.on("/mo_stp", HTTP_GET, m_stop);
90 | server.begin();
91 | }
92 |
93 | void loop()
94 | {
95 |   server.handleClient();
96 |   chk_sw();
97 | }
98 |
99 | void HTTP_handleRoot(void)
100 | {
101 |   char chk_ssid=0;
102 |   if( server.hasArg("State") ){
103 |     Serial.println(server.arg("State"));
104 |     command = server.arg("State");
105 |
106 |     if (command == "0") speedCar = 400;
107 |     else if (command == "1") speedCar = 470;
108 |     else if (command == "2") speedCar = 540;
109 |     else if (command == "3") speedCar = 610;
110 |     else if (command == "4") speedCar = 680;
111 |     else if (command == "5") speedCar = 750;
112 |     else if (command == "6") speedCar = 820;
113 |     else if (command == "7") speedCar = 890;
114 |     else if (command == "8") speedCar = 960;
115 |     else if (command == "9") speedCar = 1023;
116 |   }
117 |
118 |   if( server.hasArg("IP") ){ Serial.println(server.arg("IP")); }
119 |
120 |   if( server.hasArg("SSID") )
121 |   {
122 |     ssid2="";
123 |     ssid2 = server.arg("SSID");
124 |     Serial.println(ssid2);
125 |     chk_ssid=1;
126 |   }
127 |
128 |   server.send ( 200, "text/html", "" );
129 |   delay(1);
130 |
131 |   if(chk_ssid==1)
132 |   {
133 |     w_eeprom();
134 |     WiFi.disconnect();
135 |     delay(100);
136 |     WiFi.softAP(ssid2, password);
137 |   }
138 | }
```

คำอธิบายโค้ดควบคุมหุ่นยนต์

บรรทัดที่ 1 กำหนดให้ M2_A เท่ากับขา D5

บรรทัดที่ 2 กำหนดให้ M2_B เท่ากับขา D6

บรรทัดที่ 3 กำหนดให้ M1_A เท่ากับขา D7

บรรทัดที่ 4 กำหนดให้ M1_B เท่ากับขา D8

บรรทัดที่ 6 กำหนดให้ led_r เท่ากับขา D0

บรรทัดที่ 7 กำหนดให้ led_ct เท่ากับขา D2

บรรทัดที่ 8 กำหนดให้ led_l เท่ากับขา D3

บรรทัดที่ 10 กำหนดให้ sw_setting เท่ากับขา D1

บรรทัดที่ 12-15 เรียกใช้ไลบรารี ESP8266WiFi.h, WiFiClient.h, ESP8266WebServer.h และ EEPROM.h

บรรทัดที่ 19 กำหนดให้ speedCar มีค่าเท่ากับ 800

บรรทัดที่ 20 กำหนดให้ speed_Coeff มีค่าเท่ากับ 3

บรรทัดที่ 21 รหัสผ่านของ WiFi ที่ต้องการเชื่อมต่อ ในโค้ด คือ 123456789

บรรทัดที่ 25 สร้างออปเจ็ค server เพื่อเริ่มใช้งาน Web Server ที่พอร์ต 80

บรรทัดที่ 27 กำหนดให้ led_r_off มีสถานะเป็น HIGH

บรรทัดที่ 28 กำหนดให้ led_r_on มีสถานะเป็น LOW

บรรทัดที่ 29 กำหนดให้ led_l_off มีสถานะเป็น HIGH

บรรทัดที่ 30 กำหนดให้ led_l_on มีสถานะเป็น LOW

บรรทัดที่ 31 กำหนดให้ led_ct_off มีสถานะเป็น HIGH

บรรทัดที่ 32 กำหนดให้ led_ct_on มีสถานะเป็น LOW

บรรทัดที่ 34 กำหนดให้ add_pwm มีค่าเท่ากับ 40

บรรทัดที่ 35 เรียกใช้ไลบรารี inc_control.h

บรรทัดที่ 37 สร้างฟังก์ชันย่อย w_eeprom() ใช้เก็บชื่อหุ่นยนต์ที่เราตั้งไว้จากแอปพลิเคชัน ใน EEPROM ของ NODE MCU V.3

บรรทัดที่ 45 สร้างฟังก์ชันย่อย r_eeprom() ใช้สำหรับอ่านชื่อ SSID ของหุ่นยนต์ ที่เราเก็บเอาไว้ใน EEPROM ของ NODE MCU V.3

บรรทัดที่ 51 สร้างฟังก์ชัน setup

บรรทัดที่ 53-60 กำหนดให้ M1_A,M1_B,M2_A,M2_B,led_l,led_r และ led_ct เป็นขา OUTPUT

บรรทัดที่ 61 กำหนดให้ sw_setting เป็นขา INPUT

บรรทัดที่ 71 ตัวแปร IP Address ของ NODE MCU V.3 คือ 192.168.4.1

บรรทัดที่ 72 ตัวแปร SubNet ของ NODE MCU V.3 คือ 255.255.255.0

บรรทัดที่ 74 ใช้งาน WiFi ในโหมด AP

บรรทัดที่ 75-76 กำหนดค่าไปยัง NODE MCU V.3

บรรทัดที่ 90 เปิดการใช้งาน Web Server

บรรทัดที่ 93 สร้างฟังก์ชัน loop

บรรทัดที่ 95 ตรวจสอบการเรียกใช้งาน Web Server

บรรทัดที่ 96 ตรวจสอบการกดสวิตซ์ที่หุ่นยนต์ ถ้ามีการกดสวิตซ์ จะเป็น การตั้งชื่อ SSID ของหุ่นยนต์ใหม่

บรรทัดที่ 99 สร้างออปเจ็ค HTTP_handleRoot

บรรทัดที่ 106-116 ใช้สำหรับรับคำสั่งจากแอปพลิเคชัน เพื่อปรับความเร็วของมอเตอร์

บรรทัดที่ 131-137 ใช้สำหรับเปลี่ยนชื่อหุ่นยนต์

DESCRIPTION OF SOURCE CODE

Line 1 Assign M2_A to D5 pin.

Line 2 Assign M2_B to D6 pin.

Line 3 Assign M1_A to D7 pin.

Line 4 Assign M1_B to D8 pin.

Line 6 Assign led_r to D0 pin.

Line 7 Assign led_ct to D2 pin.

Line 8 Assign led_l to D3 pin.

Line 10 Assign sw_setting to D1 pin.

Line 12-15 Load the library (ESP8266WiFi.h, WiFiClient.h, ESP8266WebServer.h and EEPROM.h).

Line 19 Define speedCar of 800.

Line 20 Define speed_Coeff of 3.

Line 21 The WiFi password you want to connect to is 123456789.

Line 25 Set web server port number to 80.

Line 27 Set led_r_off to HIGH.

Line 28 Set led_r_on to LOW.

Line 29 Set led_l_off to HIGH.

Line 30 Set led_l_on to LOW.

Line 31 Set led_ct_off to HIGH.

Line 32 Set led_ct_on to LOW.

Line 34 Define add_pwm of 40.

Line 35 Load inc_control.h library.

Line 37 Create a sub function w_eeprom () to store the robot name that we have set from the application in the EEPROM of NODE MCU V.3.

Line 45 Create a sub function r_eeprom () used for reading SSID of robot names. Which we have stored in EEPROM of NODE MCU V.3.

Line 51 Create the setup function.

Line 53-60 Assign output variables to M1_A, M1_B, M2_A, M2_B, led_l, led_r and led_ct.

Line 61 Assign input variables to sw_setting.

Line 71 The IP Address variable of NODE MCU V.3 is 192.168.4.1.

Line 72 The SubNet variable of the NODE MCU V.3 is 255.255.255.0.

Line 74 Use WiFi in AP mode.

Line 75-76 Set the value to the NODE MCU V.3.

Line 90 Enable Web Server

Line 93 Create the loop function.

Line 95 Check the web server usage.

Line 96 Check the switch on the robot. If the button is pressed, it will set SSID name of the new robot.

Line 99 Create HTTP_handleRoot object.

Line 106-116 Used for receiving commands from the application. To adjust the speed of the motor.

Line 131-137 Used for changing robot names.



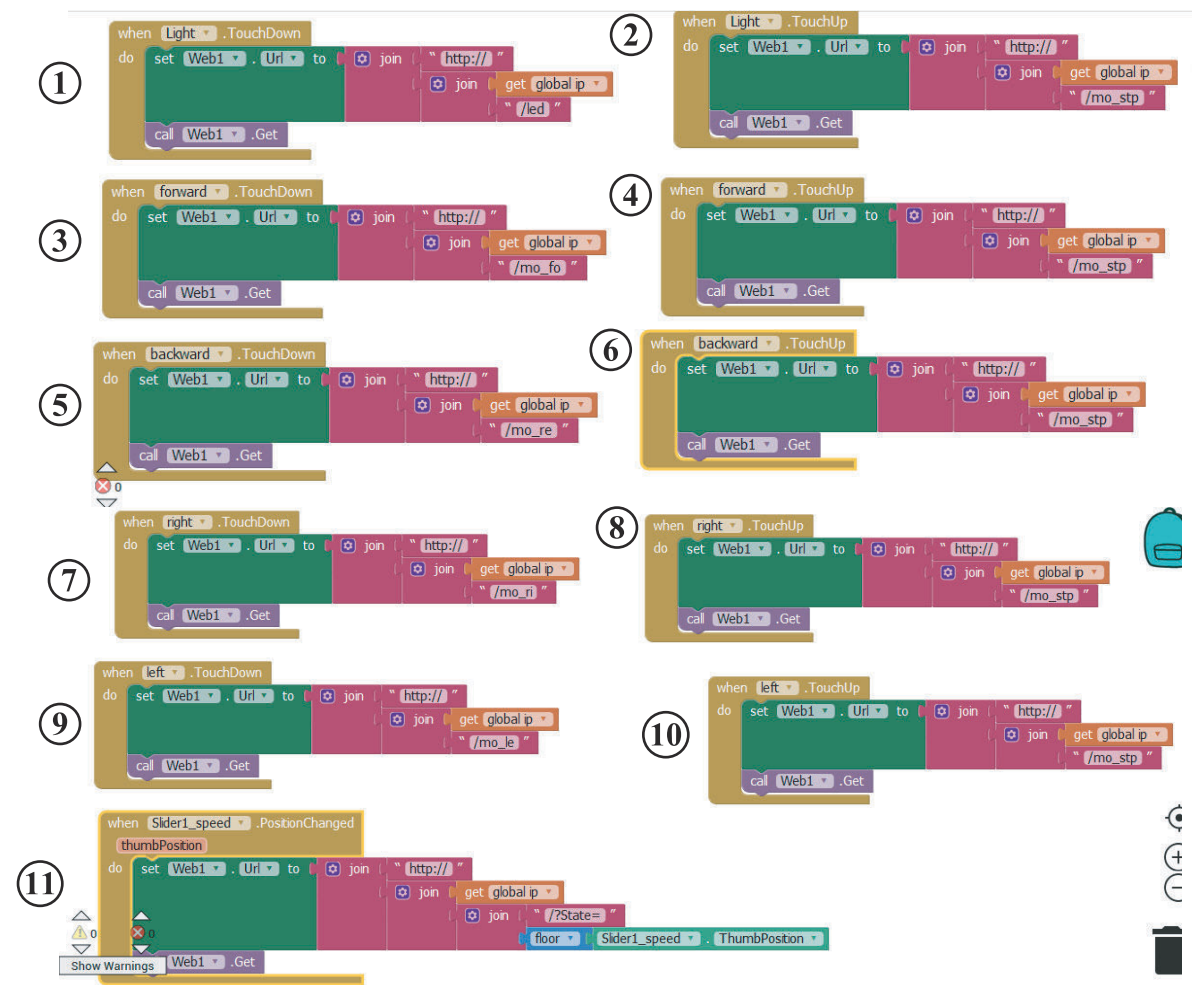
รูปที่ 5 แสดงหน้าต่างแอปพลิเคชันหน้าหลัก

Figure 5. The main page of Application



รูปที่ 6 แสดงหน้าต่างแอปพลิเคชัน เมื่อกดปุ่ม SSID

Figure 6. Application window when pressing the SSID button



รูปที่ 7 แสดงบล็อกคำสั่งของแอปพลิเคชันบนโปรแกรม APP INVENTOR

Figure 7. Block Editor of Application on App Inventor

แอปพลิเคชันบนโทรศัพท์มือถือ

จากรูปที่ 5 จะแสดงหน้าต่างของแอปพลิเคชันที่เป็นหน้าหลัก โดยในหน้าต่างจะประกอบไปด้วย

ปุ่ม LIGHT มีไว้สำหรับเปิดไฟหน้าของตัวหุ่นยนต์

ปุ่ม UP มีไว้สำหรับควบคุมหุ่นยนต์ให้วิ่งไปด้านหน้า

ปุ่ม LEFT มีไว้สำหรับควบคุมหุ่นยนต์ให้เลี้ยวซ้าย

ปุ่ม RIGHT มีไว้สำหรับควบคุมหุ่นยนต์ให้เลี้ยวขวา

ปุ่ม DOWN มีไว้สำหรับควบคุมหุ่นยนต์ให้ถอยหลัง

ปุ่ม SLIDER มีไว้สำหรับควบคุมหุ่นยนต์ให้วิ่งเร็วหรือช้าตามการปรับสไลด์ โดยถ้าปรับไปทางด้านซ้ายจะเป็นการลดความเร็ว ในทางตรงกันข้าม ถ้าปรับไปทางขวา จะเป็นการเร่งความเร็ว

ปุ่ม SSID ใช้สำหรับตั้งชื่อหุ่นยนต์ เมื่อกดปุ่มแล้ว แอปพลิเคชันจะเปลี่ยนไปยังอีกหน้าต่างหนึ่ง ดังรูปที่ 6 โดยช่อง IP จะเป็นการตั้ง IP Address ให้ตรงกับตัวหุ่นยนต์ ซึ่งในที่นี้จะใช้ 192.168.4.1 โดยถ้าต้องการเปลี่ยน จะต้องทำการเปลี่ยนที่ตัวแอปพลิเคชันและที่ตัวหุ่นยนต์ให้เหมือนกัน ช่อง SSID เป็นจุดตั้งชื่อหุ่นยนต์

ปุ่ม EXIT เป็นปุ่มออกจากหน้าต่างค่า IP และชื่อ ดังรูปที่ 6 เพื่อกลับไปหน้าจอแอปพลิเคชันหลัก

บล็อกคำสั่งของแอปพลิเคชัน

บล็อกชุดที่ 1 เมื่อทำการกดปุ่ม LIGHT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /led ไปยังตัวหุ่นยนต์เพื่อทำการเปิดไฟหน้า

บล็อกชุดที่ 2 เมื่อไม่กดปุ่ม LIGHT ตัวแอปพลิเคชันจะทำการส่ง

คำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 3 เมื่อทำการกดปุ่ม UP ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_fo ไปยังตัวหุ่นยนต์เพื่อให้หุ่นยนต์เดินหน้า

บล็อกชุดที่ 4 เมื่อไม่กดปุ่ม UP ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 5 เมื่อทำการกดปุ่ม DOWN ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_re ไปยังตัวหุ่นยนต์เพื่อให้หุ่นยนต์ถอยหลัง

บล็อกชุดที่ 6 เมื่อไม่กดปุ่ม DOWN ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 7 เมื่อทำการกดปุ่ม RIGHT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_ri ไปยังตัวหุ่นยนต์เพื่อให้หุ่นยนต์เลี้ยวขวา

บล็อกชุดที่ 8 เมื่อไม่กดปุ่ม RIGHT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 9 เมื่อทำการกดปุ่ม LEFT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_le ไปยังตัวหุ่นยนต์เพื่อให้หุ่นยนต์เลี้ยวซ้าย

บล็อกชุดที่ 10 เมื่อไม่กดปุ่ม LEFT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 11 เมื่อทำการเลื่อนปุ่ม SLIDER ตัวแอปพลิเคชันจะทำการส่งคำสั่งไปยังตัวหุ่นยนต์เพื่อควบคุมความเร็วในการวิ่ง

Mobile application

Figure 5 shows the application window that is the main page, in which the window consists of

LIGHT button is for turning on the headlight of the robot.

UP button is for controlling the robot to run forward.

LEFT button is for controlling the robot to turn left.

RIGHT button is for controlling the robot to turn right.

DOWN button is used to control the robot backwards.

SLIDER button is for controlling the robot to run fast or slow. If, it is adjusted to the left, will reduce the speed. And if adjusted to the right, will be the acceleration.

SSID button is used to name the robot. When the button is pressed, the application will change to another window as shown in Figure 6. The IP box will set the IP Address to match the robot. In this case, 192.168.4.1 will be used. If you want to change, you must change the application and the robot to the same. SSID point is the robot name.

EXIT button is the button to exit the IP and name setting page (see Figure 6) to return to Main application page.

Block editor of the application

Block 1: When pressing LIGHT button, the application will send the command "/led" to the robot. In order to turn on the headlights.

Block 2: When LIGHT button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 3: When UP button is pressed, the application will send the command "/mo_fo" to the robot in order to move forward.

Block 4: When UP button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 5: When pressing DOWN button, the application will send the command "/mo_re" to the robot in order to reverse.

Block 6: When DOWN button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 7: When pressing RIGHT button, the application will send the command "/mo_ri" to the robot to turn right.

Block 8: When RIGHT button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 9: When pressing LEFT button, the application will send the command "/mo_le" to the robot in order to turn left.

Block 10: When LEFT button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 11: When sliding SLIDER button, the application will send sliding value to the robot to control the running speed.

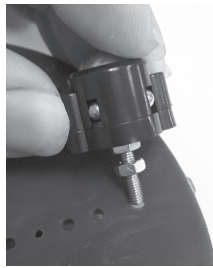
(2) ROBOT BODY

Assembling Steps of the Body set.

(2) ตัวหุ่นยนต์

ขั้นตอนการประกอบตัวหุ่นยนต์

1



Mini Caster
ชุดล้อหลัง

Fix the mini caster wheel set to the Body, by using a bolt as a holder.
ประกอบชุดล้อหลังเข้ากับตัวหุ่นยนต์ทางด้านหลัง โดยใช้น็อตที่มากับชุดล้อหลัง เป็นตัวยึด



Flat head nut 2.5x10 and NUT M2.5
น็อตหัวปเปอร์ตัวผู้ 2.5x10 และตัวเมีย M2.5

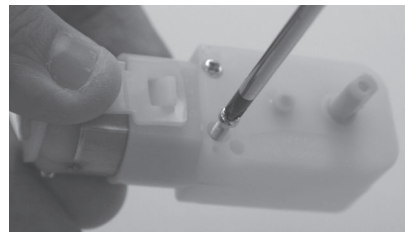
2

Install battery holder to robot body with flat head nut 2.5x10 and nut M2.5.
ประกอบปลั๊กถ่านขนาด AA ทั้งสองตัวเข้ากับตัวหุ่นยนต์โดยใช้น็อตหัวปเปอร์ ตัวผู้ 2.5x10 และ ตัวเมีย M2.5 เป็นตัวยึด

3



Insert the electric wire of battery holder into robot body.
สอดสายไฟของปลั๊กถ่านเข้ามาด้านบน

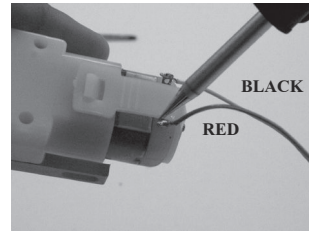
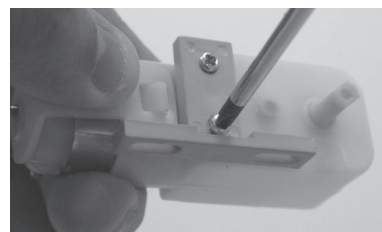


Take off the both screw of motor gear and then mount the motor lock.

Secure with the both screw of motor gear.

ทำการถอดน็อตของมอเตอร์เกียร์ออก จากนั้นให้ทำการยึดตัวล็อกเข้ากับมอเตอร์ โดยใช้น็อตที่ถอดออกมาจากตัวมอเตอร์เกียร์เป็นตัวยึด

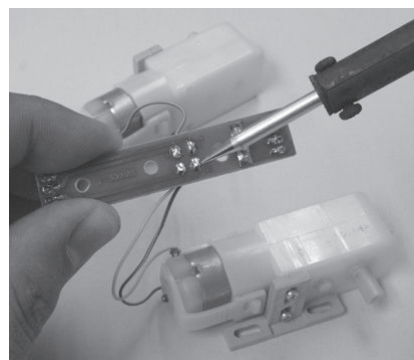
4



Solder electric wire at motor pole with red wire solders at left hand side and black wire solders at right hand side.

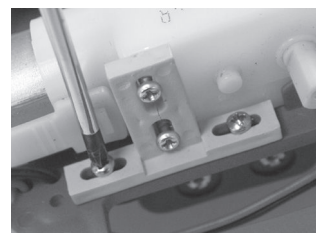
บัดกรีสายไฟที่ขั้วของมอเตอร์ โดยให้หันด้านท้ายของมอเตอร์เข้าหาตัวแล้วบัดกรีสายสีแดงทางด้านซ้ายและสายสีดำที่ด้านขวา

5



Solder motor wire to BR002 PC-board. Red wire is positive pole and black wire is negative pole. Character "L" is left motor gear and "R" is right motor gear.

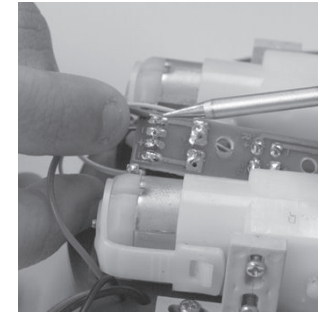
บัดกรีสายมอเตอร์เข้ากับแผงวงจรพิมพ์ BR002 โดยบัดกรีที่ตำแหน่ง MOTOR สายสีแดง ให้บัดกรีที่ ตำแหน่งบวกและสายสีดำบัดกรีที่ ตำแหน่งลบ ส่วนตัวอักษร L คือ มอเตอร์เกียร์ทางด้านซ้ายและตัวอักษร R คือ มอเตอร์เกียร์ทางด้านขวา เมื่อบัดกรีสายไฟเรียบร้อยแล้ว



6

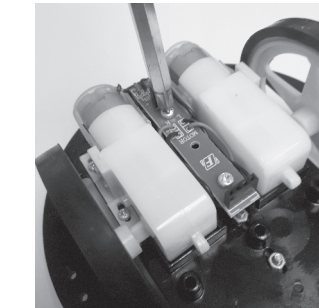
Mount motors, each with two #4 x 1/4" screws
ยึดมอเตอร์กับตัวหุ่นยนต์ โดยใช้สกรูขนาด 4x1/4

7



Solder battery holder wire to BR002 PC-board at B1 and B2. Red wire is positive pole and Black is negative pole.

บัดกรีสายปลั๊กถ่านเข้ากับแผงวงจรพิมพ์ BR002 โดยบัดกรีที่ตำแหน่ง B1 และ B2 สายสีแดงให้ทำการบัดกรีที่ตำแหน่งบวกและสายสีดำให้ทำการบัดกรีที่ตำแหน่งลบ



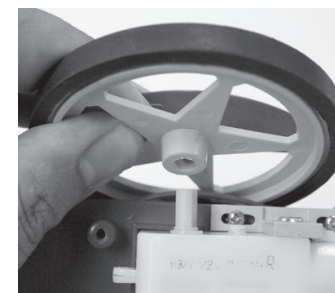
Screw 2x1/4
สกรู 2x1/4

8

Mount BR002 PC-board into body robot and secure them with two #2 x 1/4" screws.

ยึดแผงวงจรพิมพ์ BR002 กับตัวหุ่นยนต์ โดยใช้สกรูขนาด 2x1/4

9



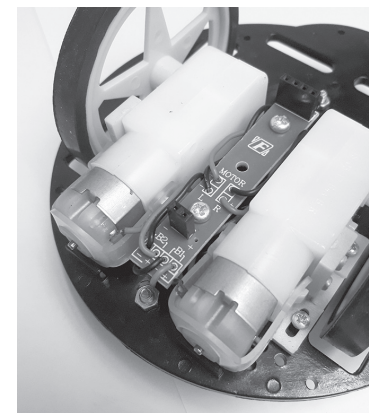
Install the wheels onto the shaft of the gear motors and secure them with the remaining two #4 x 1/4" pointy screws.

นำล้อหุ่นยนต์มาสวมเข้ากับแกนมอเตอร์เกียร์ จากนั้นให้ใช้สกรูขนาด 4x1/4 ยึดที่รูตรงกลางของล้อ



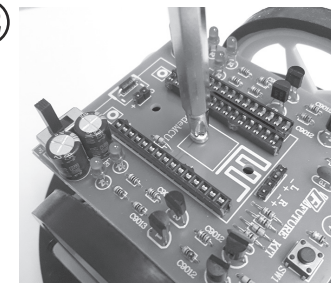
Screw 4x1/4
สกรู 4x1/4

10



Body robot is completely installed.
ตัวหุ่นยนต์ที่ประกอบเสร็จเรียบร้อยแล้ว

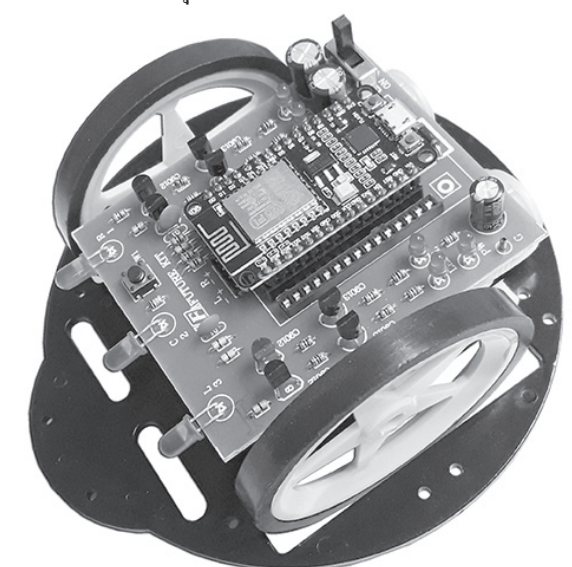
12



ทำการยึดแผ่นควบคุมด้วยสกรูขนาด 4x3/4
Secure control board with two #4 x 3/4" screws.

13

The robot is prompt working and playing.
หุ่นยนต์ที่ประกอบเสร็จเรียบร้อยแล้ว



11



Install the control board into robot body.

นำแผ่นควบคุมมาเสียบลงที่ตัวหุ่นยนต์ ซึ่งได้ติดตั้งแผ่น BR002 เรียบร้อยแล้ว

โค้ดควบคุมหุ่นยนต์ SOURCE CODE OF ROBOT
(สามารถดาวน์โหลดโค้ดนี้ได้ที่ www.futurekit.com)
(This code can be downloaded at www.futurekit.com)

```
1 | #define M2_A D5
2 | #define M2_B D6
3 | #define M1_A D7
4 | #define M1_B D8
5 |
6 | #define led_r D0
7 | #define led_ct D2
8 | #define led_l D3
9 |
10 | #define sw_setting D1
11 |
12 | #include <ESP8266WiFi.h>
13 | #include <WiFiClient.h>
14 | #include <ESP8266WebServer.h>
15 | #include <EEPROM.h>
16 |
17 | void HTTP_handleRoot(void);
18 | String command; //String to store app command state.
19 | int speedCar = 800; // 400 - 1023.
20 | int speed_Coeff = 3;
21 | const char* password = "12345678";
22 | char t_ssid;
23 | String ssid2;
24 |
25 | ESP8266WebServer server(80);
26 |
27 | #define led_r_off digitalWrite(led_r,HIGH);
28 | #define led_r_on digitalWrite(led_r,LOW);
29 | #define led_l_off digitalWrite(led_l,HIGH);
30 | #define led_l_on digitalWrite(led_l,LOW);
31 | #define led_ct_off digitalWrite(led_ct,HIGH);
32 | #define led_ct_on digitalWrite(led_ct,LOW);
33 |
34 | #define add_pwm 40
35 | #include "inc_control.h"
36 |
37 | void w_eeprom()
38 | {
39 |   for (int i = 10; i < 20; ++i){EEPROM.write(i, 0);} // clearing eeprom
40 |   for (int i = 0; i < ssid2.length(); ++i){EEPROM.write(i, ssid2[i]);}
41 |   EEPROM.commit();
42 |   delay(10);
43 | }
44 |
45 | void r_eeprom()
46 | {
47 |   t_ssid = EEPROM.read(0);
48 |   for (int i = 10; i < 20; ++i) { ssid2 += char(EEPROM.read(i)); }
49 | }
50 |
51 | void setup()
52 | {
53 |   pinMode(M1_A,OUTPUT);
54 |   pinMode(M1_B,OUTPUT);
55 |   pinMode(M2_A,OUTPUT);
56 |   pinMode(M2_B,OUTPUT);
57 |
58 |   pinMode(led_l, OUTPUT);
59 |   pinMode(led_r, OUTPUT);
60 |   pinMode(led_ct, OUTPUT);
61 |   pinMode(sw_setting, INPUT_PULLUP);
62 |
63 |   led_r_off;
64 |   led_l_off;
65 |   led_ct_off;
66 |   EEPROM.begin(512); //Initialize EEPROM
67 |   Serial.begin(115200);
```

```
68 | r_eeprom();
69 |
70 | chk_sw();
71 | IPAddress apIP(192, 168, 4, 1);/--set ip
72 | IPAddress subIP(255, 255, 255, 0);
73 |
74 | WiFi.mode(WIFI_AP);
75 | WiFi.softAPConfig(apIP, apIP, subIP);
76 | WiFi.softAP(ssid2, password);
77 |
78 | IPAddress myIP = WiFi.softAPIP();
79 | Serial.print("AP IP address: ");
80 | Serial.println(myIP);
81 |
82 | server.on ( "/", HTTP_handleRoot );
83 | server.onNotFound ( HTTP_handleRoot );
84 | server.on("/led", HTTP_GET, light_ct);
85 | server.on("/mo_fo", HTTP_GET, m_forward);
86 | server.on("/mo_re", HTTP_GET, m_reverse);
87 | server.on("/mo_le", HTTP_GET, m_turn_left);
88 | server.on("/mo_ni", HTTP_GET, m_turn_right);
89 | server.on("/mo_stp", HTTP_GET, m_stop);
90 | server.begin();
91 | }
92 |
93 | void loop()
94 | {
95 |   server.handleClient();
96 |   chk_sw();
97 | }
98 |
99 | void HTTP_handleRoot(void)
100 | {
101 |   char chk_ssid=0;
102 |   if( server.hasArg("State") ){
103 |     Serial.println(server.arg("State"));
104 |     command = server.arg("State");
105 |
106 |     if (command == "0") speedCar = 400;
107 |     else if (command == "1") speedCar = 470;
108 |     else if (command == "2") speedCar = 540;
109 |     else if (command == "3") speedCar = 610;
110 |     else if (command == "4") speedCar = 680;
111 |     else if (command == "5") speedCar = 750;
112 |     else if (command == "6") speedCar = 820;
113 |     else if (command == "7") speedCar = 890;
114 |     else if (command == "8") speedCar = 960;
115 |     else if (command == "9") speedCar = 1023;
116 |   }
117 |
118 |   if( server.hasArg("IP") ){ Serial.println(server.arg("IP")); }
119 |
120 |   if( server.hasArg("SSID") )
121 |   {
122 |     ssid2="";
123 |     ssid2 = server.arg("SSID");
124 |     Serial.println(ssid2);
125 |     chk_ssid=1;
126 |   }
127 |
128 |   server.send ( 200, "text/html", "" );
129 |   delay(1);
130 |
131 |   if(chk_ssid==1)
132 |   {
133 |     w_eeprom();
134 |     WiFi.disconnect();
135 |     delay(100);
136 |     WiFi.softAP(ssid2, password);
137 |   }
138 | }
```

คำอธิบายโค้ดควบคุมหุ่นยนต์

บรรทัดที่ 1 กำหนดให้ M2_A เท่ากับขา D5

บรรทัดที่ 2 กำหนดให้ M2_B เท่ากับขา D6

บรรทัดที่ 3 กำหนดให้ M1_A เท่ากับขา D7

บรรทัดที่ 4 กำหนดให้ M1_B เท่ากับขา D8

บรรทัดที่ 6 กำหนดให้ led_r เท่ากับขา D0

บรรทัดที่ 7 กำหนดให้ led_ct เท่ากับขา D2

บรรทัดที่ 8 กำหนดให้ led_l เท่ากับขา D3

บรรทัดที่ 10 กำหนดให้ sw_setting เท่ากับขา D1

บรรทัดที่ 12-15 เรียกใช้ไลบรารี ESP8266WiFi.h, WiFiClient.h,

ESP8266WebServer.h และ EEPROM.h

บรรทัดที่ 19 กำหนดให้ speedCar มีค่าเท่ากับ 800

บรรทัดที่ 20 กำหนดให้ speed_Coeff มีค่าเท่ากับ 3

บรรทัดที่ 21 รหัสผ่านของ WiFi ที่ต้องการเชื่อมต่อ ในโค้ด คือ 123456789

บรรทัดที่ 25 สร้างออปเจ็ค server เพื่อเริ่มใช้งาน Web Server ที่พอร์ต 80

บรรทัดที่ 27 กำหนดให้ led_r_off มีสถานะเป็น HIGH

บรรทัดที่ 28 กำหนดให้ led_r_on มีสถานะเป็น LOW

บรรทัดที่ 29 กำหนดให้ led_l_off มีสถานะเป็น HIGH

บรรทัดที่ 30 กำหนดให้ led_l_on มีสถานะเป็น LOW

บรรทัดที่ 31 กำหนดให้ led_ct_off มีสถานะเป็น HIGH

บรรทัดที่ 32 กำหนดให้ led_ct_on มีสถานะเป็น LOW

บรรทัดที่ 34 กำหนดให้ add_pwm มีค่าเท่ากับ 40

บรรทัดที่ 35 เรียกใช้ไลบรารี inc_control.h

บรรทัดที่ 37 สร้างฟังก์ชันย่อย w_eeprom() ใช้เก็บชื่อหุ่นยนต์ที่เราตั้งไว้จากแอปพลิเคชัน ใน EEPROM ของ NODE MCU V.3

บรรทัดที่ 45 สร้างฟังก์ชันย่อย r_eeprom() ใช้สำหรับอ่านชื่อ SSID ของหุ่นยนต์ ที่เราเก็บเอาไว้ใน EEPROM ของ NODE MCU V.3

บรรทัดที่ 51 สร้างฟังก์ชัน setup

บรรทัดที่ 53-60 กำหนดให้ M1_A,M1_B,M2_A,M2_B,led_l,led_r และ led_ct เป็นขา OUTPUT

บรรทัดที่ 61 กำหนดให้ sw_setting เป็นขา INPUT

บรรทัดที่ 71 ตัวแปร IP Address ของ NODE MCU V.3 คือ 192.168.4.1

บรรทัดที่ 72 ตัวแปร SubNet ของ NODE MCU V.3 คือ 255.255.255.0

บรรทัดที่ 74 ใช้งาน WiFi ในโหมด AP

บรรทัดที่ 75-76 กำหนดค่าไปยัง NODE MCU V.3

บรรทัดที่ 90 เปิดการใช้งาน Web Server

บรรทัดที่ 93 สร้างฟังก์ชัน loop

บรรทัดที่ 95 ตรวจสอบการเรียกใช้งาน Web Server

บรรทัดที่ 96 ตรวจสอบการกดสวิตซ์ที่หุ่นยนต์ ถ้ามีการกดสวิตซ์ จะเป็น การตั้งชื่อ SSID ของหุ่นยนต์ใหม่

บรรทัดที่ 99 สร้างออปเจ็ค HTTP_handleRoot

บรรทัดที่ 106-116 ใช้สำหรับรับคำสั่งจากแอปพลิเคชัน เพื่อปรับความเร็วของมอเตอร์

บรรทัดที่ 131-137 ใช้สำหรับเปลี่ยนชื่อหุ่นยนต์

DESCRIPTION OF SOURCE CODE

Line 1 Assign M2_A to D5 pin.

Line 2 Assign M2_B to D6 pin.

Line 3 Assign M1_A to D7 pin.

Line 4 Assign M1_B to D8 pin.

Line 6 Assign led_r to D0 pin.

Line 7 Assign led_ct to D2 pin.

Line 8 Assign led_l to D3 pin.

Line 10 Assign sw_setting to D1 pin.

Line 12-15 Load the library (ESP8266WiFi.h, WiFiClient.h, ESP8266WebServer.h and EEPROM.h).

Line 19 Define speedCar of 800.

Line 20 Define speed_Coeff of 3.

Line 21 The WiFi password you want to connect to is 123456789.

Line 25 Set web server port number to 80.

Line 27 Set led_r_off to HIGH.

Line 28 Set led_r_on to LOW.

Line 29 Set led_l_off to HIGH.

Line 30 Set led_l_on to LOW.

Line 31 Set led_ct_off to HIGH.

Line 32 Set led_ct_on to LOW.

Line 34 Define add_pwm of 40.

Line 35 Load inc_control.h library.

Line 37 Create a sub function w_eeprom () to store the robot name that we have set from the application in the EEPROM of NODE MCU V.3.

Line 45 Create a sub function r_eeprom () used for reading SSID of robot names. Which we have stored in EEPROM of NODE MCU V.3.

Line 51 Create the setup function.

Line 53-60 Assign output variables to M1_A, M1_B, M2_A, M2_B, led_l, led_r and led_ct.

Line 61 Assign input variables to sw_setting.

Line 71 The IP Address variable of NODE MCU V.3 is 192.168.4.1.

Line 72 The SubNet variable of the NODE MCU V.3 is 255.255.255.0.

Line 74 Use WiFi in AP mode.

Line 75-76 Set the value to the NODE MCU V.3.

Line 90 Enable Web Server

Line 93 Create the loop function.

Line 95 Check the web server usage.

Line 96 Check the switch on the robot. If the button is pressed, it will set SSID name of the new robot.

Line 99 Create HTTP_handleRoot object.

Line 106-116 Used for receiving commands from the application. To adjust the speed of the motor.

Line 131-137 Used for changing robot names.



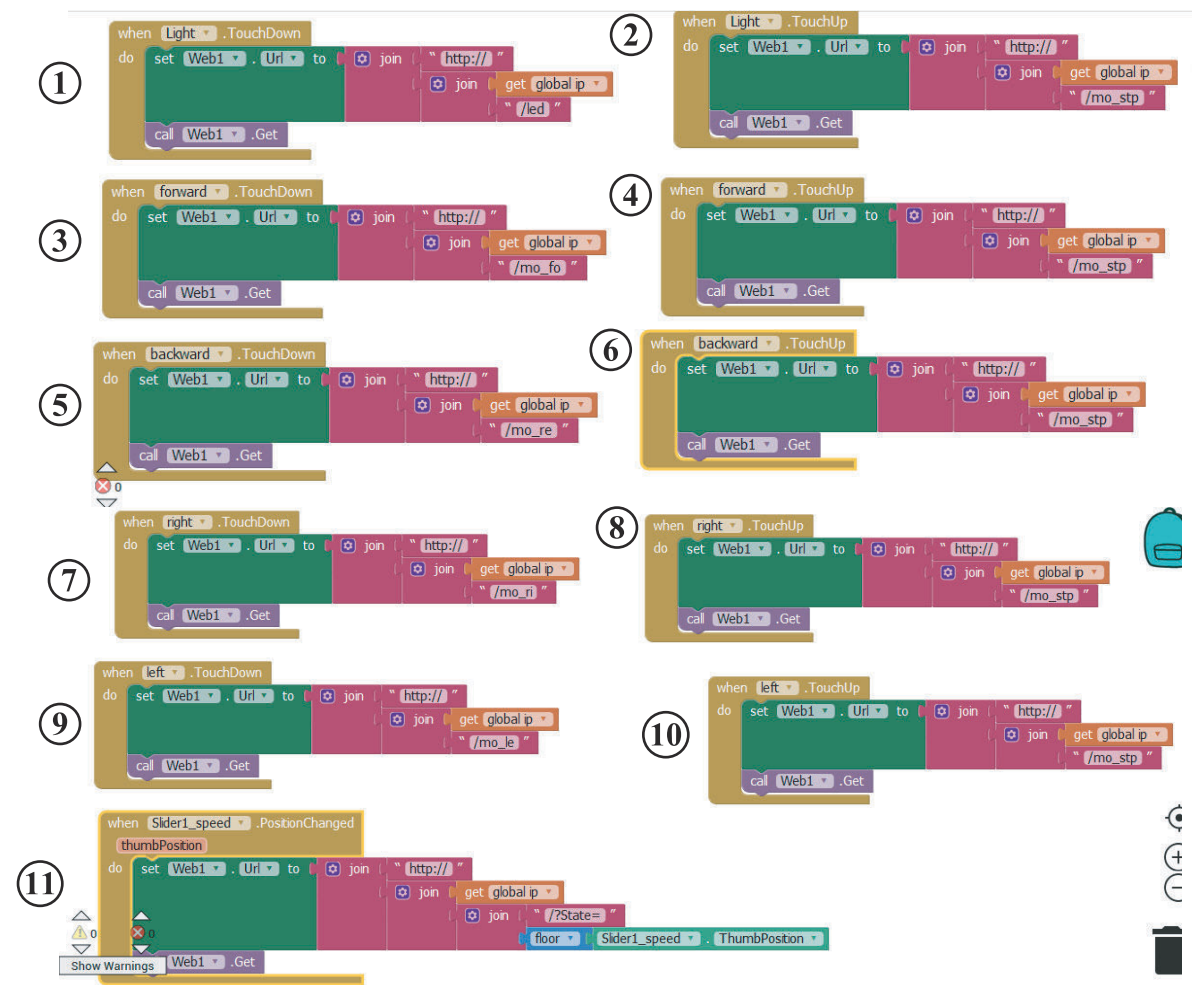
รูปที่ 5 แสดงหน้าต่างแอปพลิเคชันหน้าหลัก

Figure 5. The main page of Application



รูปที่ 6 แสดงหน้าต่างแอปพลิเคชัน เมื่อกดปุ่ม SSID

Figure 6. Application window when pressing the SSID button



รูปที่ 7 แสดงบล็อกคำสั่งของแอปพลิเคชันบนโปรแกรม APP INVENTOR

Figure 7. Block Editor of Application on App Inventor

แอปพลิเคชันบนโทรศัพท์มือถือ

จากรูปที่ 5 จะแสดงหน้าต่างของแอปพลิเคชันที่เป็นหน้าหลัก โดยในหน้าต่างจะประกอบไปด้วย

ปุ่ม LIGHT มีไว้สำหรับเปิดไฟหน้าของตัวหุ่นยนต์

ปุ่ม UP มีไว้สำหรับควบคุมหุ่นยนต์ให้วิ่งไปด้านหน้า

ปุ่ม LEFT มีไว้สำหรับควบคุมหุ่นยนต์ให้เลี้ยวซ้าย

ปุ่ม RIGHT มีไว้สำหรับควบคุมหุ่นยนต์ให้เลี้ยวขวา

ปุ่ม DOWN มีไว้สำหรับควบคุมหุ่นยนต์ให้ถอยหลัง

ปุ่ม SLIDER มีไว้สำหรับควบคุมหุ่นยนต์ให้วิ่งเร็วหรือช้าตามการปรับสไลด์ โดยถ้าปรับไปทางด้านซ้ายจะเป็นการลดความเร็ว ในทางตรงกันข้าม ถ้าปรับไปทางขวา จะเป็นการเร่งความเร็ว

ปุ่ม SSID ใช้สำหรับตั้งชื่อหุ่นยนต์ เมื่อกดปุ่มแล้ว แอปพลิเคชันจะเปลี่ยนไปยังอีกหน้าต่างหนึ่ง ดังรูปที่ 6 โดยช่อง IP จะเป็นการตั้ง IP Address ให้ตรงกับตัวหุ่นยนต์ ซึ่งในที่นี้จะใช้ 192.168.4.1 โดยถ้าต้องการเปลี่ยน จะต้องทำการเปลี่ยนที่ตัวแอปพลิเคชันและที่ตัวหุ่นยนต์ให้เหมือนกัน ช่อง SSID เป็นจุดตั้งชื่อหุ่นยนต์

ปุ่ม EXIT เป็นปุ่มออกจากหน้าต่างค่า IP และชื่อ ดังรูปที่ 6 เพื่อกลับไปหน้าจอแอปพลิเคชันหลัก

บล็อกคำสั่งของแอปพลิเคชัน

บล็อกชุดที่ 1 เมื่อทำการกดปุ่ม LIGHT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /led ไปยังตัวหุ่นยนต์เพื่อทำการเปิดไฟหน้า

บล็อกชุดที่ 2 เมื่อไม่กดปุ่ม LIGHT ตัวแอปพลิเคชันจะทำการส่ง

คำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 3 เมื่อทำการกดปุ่ม UP ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_fo ไปยังตัวหุ่นยนต์เพื่อให้หุ่นยนต์เดินหน้า

บล็อกชุดที่ 4 เมื่อไม่กดปุ่ม UP ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 5 เมื่อทำการกดปุ่ม DOWN ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_re ไปยังตัวหุ่นยนต์เพื่อให้หุ่นยนต์ถอยหลัง

บล็อกชุดที่ 6 เมื่อไม่กดปุ่ม DOWN ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 7 เมื่อทำการกดปุ่ม RIGHT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_rh ไปยังตัวหุ่นยนต์เพื่อให้หุ่นยนต์เลี้ยวขวา

บล็อกชุดที่ 8 เมื่อไม่กดปุ่ม RIGHT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 9 เมื่อทำการกดปุ่ม LEFT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_le ไปยังตัวหุ่นยนต์เพื่อให้หุ่นยนต์เลี้ยวซ้าย

บล็อกชุดที่ 10 เมื่อไม่กดปุ่ม LEFT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 11 เมื่อทำการเลื่อนปุ่ม SLIDER ตัวแอปพลิเคชันจะทำการส่งคำสั่งสไลด์ไปยังตัวหุ่นยนต์เพื่อควบคุมความเร็วในการวิ่ง

Mobile application

Figure 5 shows the application window that is the main page, in which the window consists of

LIGHT button is for turning on the headlight of the robot.

UP button is for controlling the robot to run forward.

LEFT button is for controlling the robot to turn left.

RIGHT button is for controlling the robot to turn right.

DOWN button is used to control the robot backwards.

SLIDER button is for controlling the robot to run fast or slow. If, it is adjusted to the left, will reduce the speed. And if adjusted to the right, will be the acceleration.

SSID button is used to name the robot. When the button is pressed, the application will change to another window as shown in Figure 6. The IP box will set the IP Address to match the robot. In this case, 192.168.4.1 will be used. If you want to change, you must change the application and the robot to the same. SSID point is the robot name.

EXIT button is the button to exit the IP and name setting page (see Figure 6) to return to Main application page.

Block editor of the application

Block 1: When pressing LIGHT button, the application will send the command "/led" to the robot. In order to turn on the headlights.

Block 2: When LIGHT button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 3: When UP button is pressed, the application will send the command "/mo_fo" to the robot in order to move forward.

Block 4: When UP button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 5: When pressing DOWN button, the application will send the command "/mo_re" to the robot in order to reverse.

Block 6: When DOWN button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 7: When pressing RIGHT button, the application will send the command "/mo_rh" to the robot to turn right.

Block 8: When RIGHT button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 9: When pressing LEFT button, the application will send the command "/mo_le" to the robot in order to turn left.

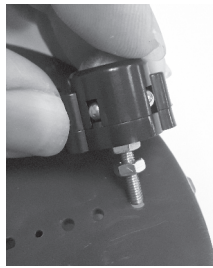
Block 10: When LEFT button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 11: When sliding SLIDER button, the application will send sliding value to the robot to control the running speed.

(2) ROBOT BODY

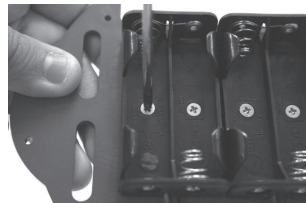
Assembling Steps of the Body set.

①



Mini Caster
ชุดล้อหลัง

Fix the mini caster wheel set to the Body, by using a bolt as a holder.
ประกอบชุดล้อหลังเข้ากับตัวหุ่นยนต์ทางด้านหลัง โดยใช้น็อตที่มากับชุดล้อหลัง เป็นตัวยึด



Flat head nut 2.5x10 and NUT M2.5
น็อตหัวแปปเปอร์ตัวผู้ 2.5x10 และตัวเมีย M2.5

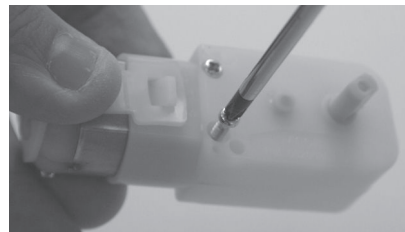
②

Install battery holder to robot body with flat head nut 2.5x10 and nut M2.5.
ประกอบถังถ่านขนาด AA ทั้งสองตัวเข้ากับตัวหุ่นยนต์โดยใช้น็อตหัวแปปเปอร์ ตัวผู้ 2.5x10 และ ตัวเมีย M2.5 เป็นตัวยึด

③



Insert the electric wire of battery holder into robot body.
สอดสายไฟของถังถ่านขึ้นมาด้านบน

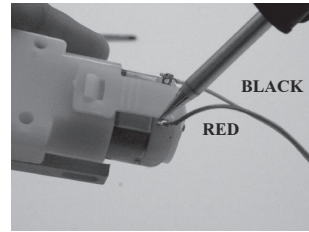
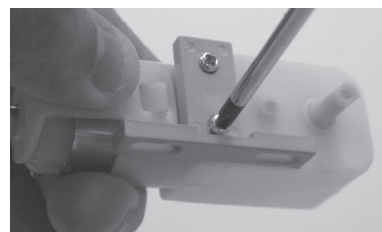


Take off the both screw of motor gear and then mount the motor lock.

Secure with the both screw of motor gear.

ทำการถอดน็อตของมอเตอร์เกียร์ออก จากนั้นให้ทำการยึดตัวล็อกเข้ากับมอเตอร์ โดยใช้น็อตที่ถอดออกมาจากตัวมอเตอร์เกียร์เป็นตัวยึด

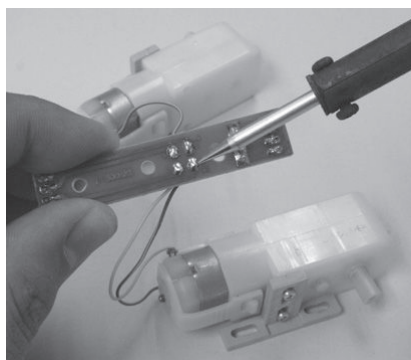
④



Solder electric wire at motor pole with red wire solders at left hand side and black wire solders at right hand side.

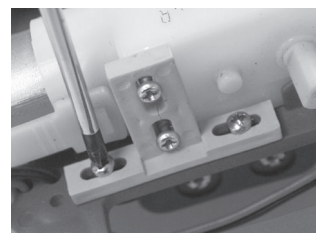
บัดกรีสายไฟที่ขั้วของมอเตอร์ โดยให้หันด้านท้ายของมอเตอร์เข้าหาตัวแล้วบัดกรีสายสีแดงทางด้านซ้ายและสายสีดำที่ด้านขวา

⑤



Solder motor wire to BR002 PC-board. Red wire is positive pole and black wire is negative pole. Character "L" is left motor gear and "R" is right motor gear.

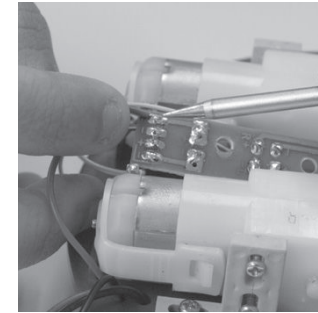
บัดกรีสายมอเตอร์เข้ากับแผงวงจรพิมพ์ BR002 โดยบัดกรีที่ตำแหน่ง MOTOR สายสีแดง ให้บัดกรีที่ ตำแหน่งบวกและสายสีดำบัดกรีที่ ตำแหน่งลบ ส่วนตัวอักษร L คือ มอเตอร์เกียร์ทางด้านซ้ายและตัวอักษร R คือ มอเตอร์เกียร์ทางด้านขวา เมื่อบัดกรีสายไฟเรียบร้อยแล้ว



⑥

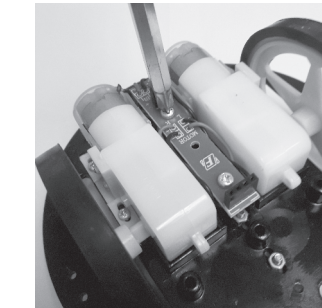
Mount motors, each with two #4 x 1/4" screws
ยึดมอเตอร์กับตัวหุ่นยนต์ โดยใช้สกรูขนาด 4x1/4

⑦



Solder battery holder wire to BR002 PC-board at B1 and B2. Red wire is positive pole and Black is negative pole.

บัดกรีสายถังถ่านเข้ากับแผงวงจรพิมพ์ BR002 โดยบัดกรีที่ตำแหน่ง B1 และ B2 สายสีแดงให้ทำการบัดกรีที่ตำแหน่งบวกและสายสีดำให้บัดกรีทำการบัดกรีที่ตำแหน่งลบ



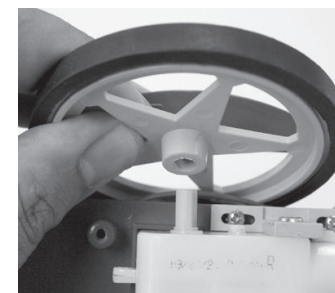
Screw 2x1/4
สกรู 2x1/4

⑧

Mount BR002 PC-board into body robot and secure them with two #2 x 1/4" screws.

ยึดแผงวงจรพิมพ์ BR002 กับตัวหุ่นยนต์ โดยใช้สกรูขนาด 2x1/4

⑨



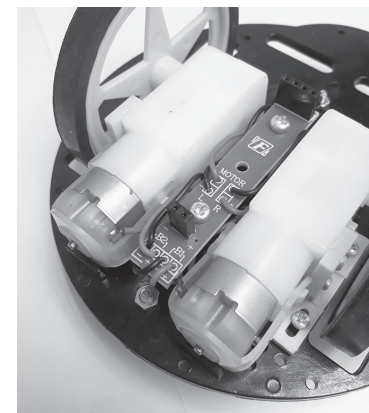
Install the wheels onto the shaft of the gear motors and secure them with the remaining two #4 x 1/4" pointy screws.

นำล้อหุ่นยนต์มาสวมเข้ากับแกนมอเตอร์เกียร์ จากนั้นให้ใช้สกรูขนาด 4x1/4 ยึดที่รูตรงกลางของล้อ



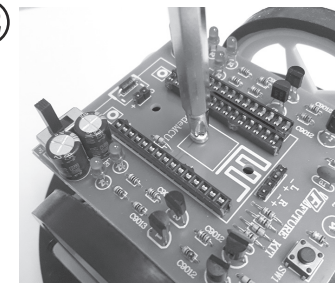
Screw 4x1/4
สกรู 4x1/4

⑩



Body robot is completely installed.
ตัวหุ่นยนต์ที่ประกอบเสร็จเรียบร้อยแล้ว

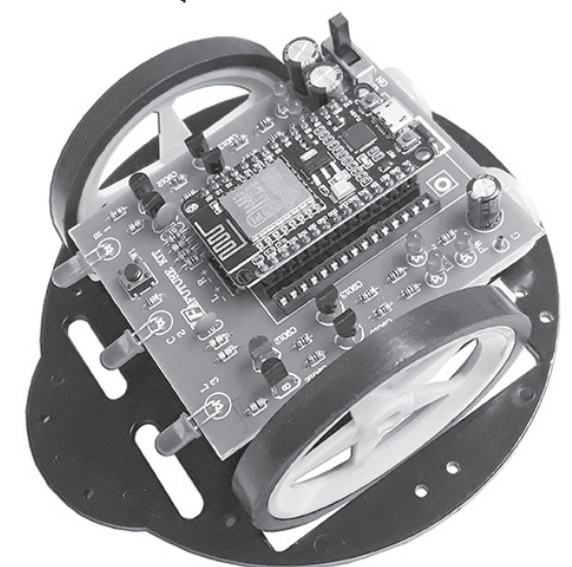
⑫



ทำการยึดแผ่นควบคุมด้วยสกรูขนาด 4x3/4
Secure control board with two #4 x 3/4" screws.

⑬

The robot is prompt working and playing.
หุ่นยนต์ที่ประกอบเสร็จเรียบร้อยแล้ว



⑪



Install the control board into robot body.

นำแผ่นควบคุมมาเสียบลงที่ตัวหุ่นยนต์ ซึ่งได้ติดตั้งแผ่น BR002 เรียบร้อยแล้ว

โค้ดควบคุมหุ่นยนต์ SOURCE CODE OF ROBOT
(สามารถดาวน์โหลดโค้ดนี้ได้ที่ www.futurekit.com)
(This code can be downloaded at www.futurekit.com)

```
1 | #define M2_A D5
2 | #define M2_B D6
3 | #define M1_A D7
4 | #define M1_B D8
5 |
6 | #define led_r D0
7 | #define led_ct D2
8 | #define led_l D3
9 |
10 | #define sw_setting D1
11 |
12 | #include <ESP8266WiFi.h>
13 | #include <WiFiClient.h>
14 | #include <ESP8266WebServer.h>
15 | #include <EEPROM.h>
16 |
17 | void HTTP_handleRoot(void);
18 | String command; //String to store app command state.
19 | int speedCar = 800; // 400 - 1023.
20 | int speed_Coeff = 3;
21 | const char* password = "12345678";
22 | char t_ssid;
23 | String ssid2;
24 |
25 | ESP8266WebServer server(80);
26 |
27 | #define led_r_off digitalWrite(led_r,HIGH);
28 | #define led_r_on digitalWrite(led_r,LOW);
29 | #define led_l_off digitalWrite(led_l,HIGH);
30 | #define led_l_on digitalWrite(led_l,LOW);
31 | #define led_ct_off digitalWrite(led_ct,HIGH);
32 | #define led_ct_on digitalWrite(led_ct,LOW);
33 |
34 | #define add_pwm 40
35 | #include "inc_control.h"
36 |
37 | void w_eeprom()
38 | {
39 |   for (int i = 10; i < 20; ++i){EEPROM.write(i, 0);} // clearing eeprom
40 |   for (int i = 0; i < ssid2.length(); ++i){EEPROM.write(i, ssid2[i]);}
41 |   EEPROM.commit();
42 |   delay(10);
43 | }
44 |
45 | void r_eeprom()
46 | {
47 |   t_ssid = EEPROM.read(0);
48 |   for (int i = 10; i < 20; ++i) { ssid2 += char(EEPROM.read(i)); }
49 | }
50 |
51 | void setup()
52 | {
53 |   pinMode(M1_A,OUTPUT);
54 |   pinMode(M1_B,OUTPUT);
55 |   pinMode(M2_A,OUTPUT);
56 |   pinMode(M2_B,OUTPUT);
57 |
58 |   pinMode(led_l, OUTPUT);
59 |   pinMode(led_r, OUTPUT);
60 |   pinMode(led_ct, OUTPUT);
61 |   pinMode(sw_setting, INPUT_PULLUP);
62 |
63 |   led_r_off;
64 |   led_l_off;
65 |   led_ct_off;
66 |   EEPROM.begin(512); //Initialize EEPROM
67 |   Serial.begin(115200);
```

```
68 | r_eeprom();
69 |
70 | chk_sw();
71 | IPAddress apIP(192, 168, 4, 1);/--set ip
72 | IPAddress subIP(255, 255, 255, 0);
73 |
74 | WiFi.mode(WIFI_AP);
75 | WiFi.softAPConfig(apIP, apIP, subIP);
76 | WiFi.softAP(ssid2, password);
77 |
78 | IPAddress myIP = WiFi.softAPIP();
79 | Serial.print("AP IP address: ");
80 | Serial.println(myIP);
81 |
82 | server.on ( "/", HTTP_handleRoot );
83 | server.onNotFound ( HTTP_handleRoot );
84 | server.on("/led", HTTP_GET, light_ct);
85 | server.on("/mo_fo", HTTP_GET, m_forward);
86 | server.on("/mo_re", HTTP_GET, m_reverse);
87 | server.on("/mo_le", HTTP_GET, m_turn_left);
88 | server.on("/mo_ni", HTTP_GET, m_turn_right);
89 | server.on("/mo_stp", HTTP_GET, m_stop);
90 | server.begin();
91 | }
92 |
93 | void loop()
94 | {
95 |   server.handleClient();
96 |   chk_sw();
97 | }
98 |
99 | void HTTP_handleRoot(void)
100 | {
101 |   char chk_ssid=0;
102 |   if( server.hasArg("State") ){
103 |     Serial.println(server.arg("State"));
104 |     command = server.arg("State");
105 |
106 |     if (command == "0") speedCar = 400;
107 |     else if (command == "1") speedCar = 470;
108 |     else if (command == "2") speedCar = 540;
109 |     else if (command == "3") speedCar = 610;
110 |     else if (command == "4") speedCar = 680;
111 |     else if (command == "5") speedCar = 750;
112 |     else if (command == "6") speedCar = 820;
113 |     else if (command == "7") speedCar = 890;
114 |     else if (command == "8") speedCar = 960;
115 |     else if (command == "9") speedCar = 1023;
116 |   }
117 |
118 |   if( server.hasArg("IP") ){ Serial.println(server.arg("IP")); }
119 |
120 |   if( server.hasArg("SSID") )
121 |   {
122 |     ssid2="";
123 |     ssid2 = server.arg("SSID");
124 |     Serial.println(ssid2);
125 |     chk_ssid=1;
126 |   }
127 |
128 |   server.send ( 200, "text/html", "" );
129 |   delay(1);
130 |
131 |   if(chk_ssid==1)
132 |   {
133 |     w_eeprom();
134 |     WiFi.disconnect();
135 |     delay(100);
136 |     WiFi.softAP(ssid2, password);
137 |   }
138 | }
```

คำอธิบายโค้ดควบคุมหุ่นยนต์

บรรทัดที่ 1 กำหนดให้ M2_A เท่ากับขา D5

บรรทัดที่ 2 กำหนดให้ M2_B เท่ากับขา D6

บรรทัดที่ 3 กำหนดให้ M1_A เท่ากับขา D7

บรรทัดที่ 4 กำหนดให้ M1_B เท่ากับขา D8

บรรทัดที่ 6 กำหนดให้ led_r เท่ากับขา D0

บรรทัดที่ 7 กำหนดให้ led_ct เท่ากับขา D2

บรรทัดที่ 8 กำหนดให้ led_l เท่ากับขา D3

บรรทัดที่ 10 กำหนดให้ sw_setting เท่ากับขา D1

บรรทัดที่ 12-15 เรียกใช้ไลบรารี ESP8266WiFi.h, WiFiClient.h, ESP8266WebServer.h และ EEPROM.h

บรรทัดที่ 19 กำหนดให้ speedCar มีค่าเท่ากับ 800

บรรทัดที่ 20 กำหนดให้ speed_Coeff มีค่าเท่ากับ 3

บรรทัดที่ 21 รหัสผ่านของ WiFi ที่ต้องการเชื่อมต่อ ในโค้ด คือ 123456789

บรรทัดที่ 25 สร้างออปเจ็ค server เพื่อเริ่มใช้งาน Web Server ที่พอร์ต 80

บรรทัดที่ 27 กำหนดให้ led_r_off มีสถานะเป็น HIGH

บรรทัดที่ 28 กำหนดให้ led_r_on มีสถานะเป็น LOW

บรรทัดที่ 29 กำหนดให้ led_l_off มีสถานะเป็น HIGH

บรรทัดที่ 30 กำหนดให้ led_l_on มีสถานะเป็น LOW

บรรทัดที่ 31 กำหนดให้ led_ct_off มีสถานะเป็น HIGH

บรรทัดที่ 32 กำหนดให้ led_ct_on มีสถานะเป็น LOW

บรรทัดที่ 34 กำหนดให้ add_pwm มีค่าเท่ากับ 40

บรรทัดที่ 35 เรียกใช้ไลบรารี inc_control.h

บรรทัดที่ 37 สร้างฟังก์ชันย่อย w_eeprom() ใช้เก็บชื่อหุ่นยนต์ที่เราตั้งไว้จากแอปพลิเคชัน ใน EEPROM ของ NODE MCU V.3

บรรทัดที่ 45 สร้างฟังก์ชันย่อย r_eeprom() ใช้สำหรับอ่านชื่อ SSID ของหุ่นยนต์ ที่เราเก็บเอาไว้ใน EEPROM ของ NODE MCU V.3

บรรทัดที่ 51 สร้างฟังก์ชัน setup

บรรทัดที่ 53-60 กำหนดให้ M1_A,M1_B,M2_A,M2_B,led_l,led_r และ led_ct เป็นขา OUTPUT

บรรทัดที่ 61 กำหนดให้ sw_setting เป็นขา INPUT

บรรทัดที่ 71 ตัวแปร IP Address ของ NODE MCU V.3 คือ 192.168.4.1

บรรทัดที่ 72 ตัวแปร SubNet ของ NODE MCU V.3 คือ 255.255.255.0

บรรทัดที่ 74 ใช้งาน WiFi ในโหมด AP

บรรทัดที่ 75-76 กำหนดค่าไปยัง NODE MCU V.3

บรรทัดที่ 90 เปิดการใช้งาน Web Server

บรรทัดที่ 93 สร้างฟังก์ชัน loop

บรรทัดที่ 95 ตรวจสอบการเรียกใช้งาน Web Server

บรรทัดที่ 96 ตรวจสอบการกดสวิตซ์ที่หุ่นยนต์ ถ้ามีการกดสวิตซ์ จะเป็นการตั้งชื่อ SSID ของหุ่นยนต์ใหม่

บรรทัดที่ 99 สร้างออปเจ็ค HTTP_handleRoot

บรรทัดที่ 106-116 ใช้สำหรับรับคำสั่งจากแอปพลิเคชัน เพื่อปรับความเร็วของมอเตอร์

บรรทัดที่ 131-137 ใช้สำหรับเปลี่ยนชื่อหุ่นยนต์

DESCRIPTION OF SOURCE CODE

Line 1 Assign M2_A to D5 pin.

Line 2 Assign M2_B to D6 pin.

Line 3 Assign M1_A to D7 pin.

Line 4 Assign M1_B to D8 pin.

Line 6 Assign led_r to D0 pin.

Line 7 Assign led_ct to D2 pin.

Line 8 Assign led_l to D3 pin.

Line 10 Assign sw_setting to D1 pin.

Line 12-15 Load the library (ESP8266WiFi.h, WiFiClient.h, ESP8266WebServer.h and EEPROM.h).

Line 19 Define speedCar of 800.

Line 20 Define speed_Coeff of 3.

Line 21 The WiFi password you want to connect to is 123456789.

Line 25 Set web server port number to 80.

Line 27 Set led_r_off to HIGH.

Line 28 Set led_r_on to LOW.

Line 29 Set led_l_off to HIGH.

Line 30 Set led_l_on to LOW.

Line 31 Set led_ct_off to HIGH.

Line 32 Set led_ct_on to LOW.

Line 34 Define add_pwm of 40.

Line 35 Load inc_control.h library.

Line 37 Create a sub function w_eeprom () to store the robot name that we have set from the application in the EEPROM of NODE MCU V.3.

Line 45 Create a sub function r_eeprom () used for reading SSID of robot names. Which we have stored in EEPROM of NODE MCU V.3.

Line 51 Create the setup function.

Line 53-60 Assign output variables to M1_A, M1_B, M2_A, M2_B, led_l, led_r and led_ct.

Line 61 Assign input variables to sw_setting.

Line 71 The IP Address variable of NODE MCU V.3 is 192.168.4.1.

Line 72 The SubNet variable of the NODE MCU V.3 is 255.255.255.0.

Line 74 Use WiFi in AP mode.

Line 75-76 Set the value to the NODE MCU V.3.

Line 90 Enable Web Server

Line 93 Create the loop function.

Line 95 Check the web server usage.

Line 96 Check the switch on the robot. If the button is pressed, it will set SSID name of the new robot.

Line 99 Create HTTP_handleRoot object.

Line 106-116 Used for receiving commands from the application. To adjust the speed of the motor.

Line 131-137 Used for changing robot names.



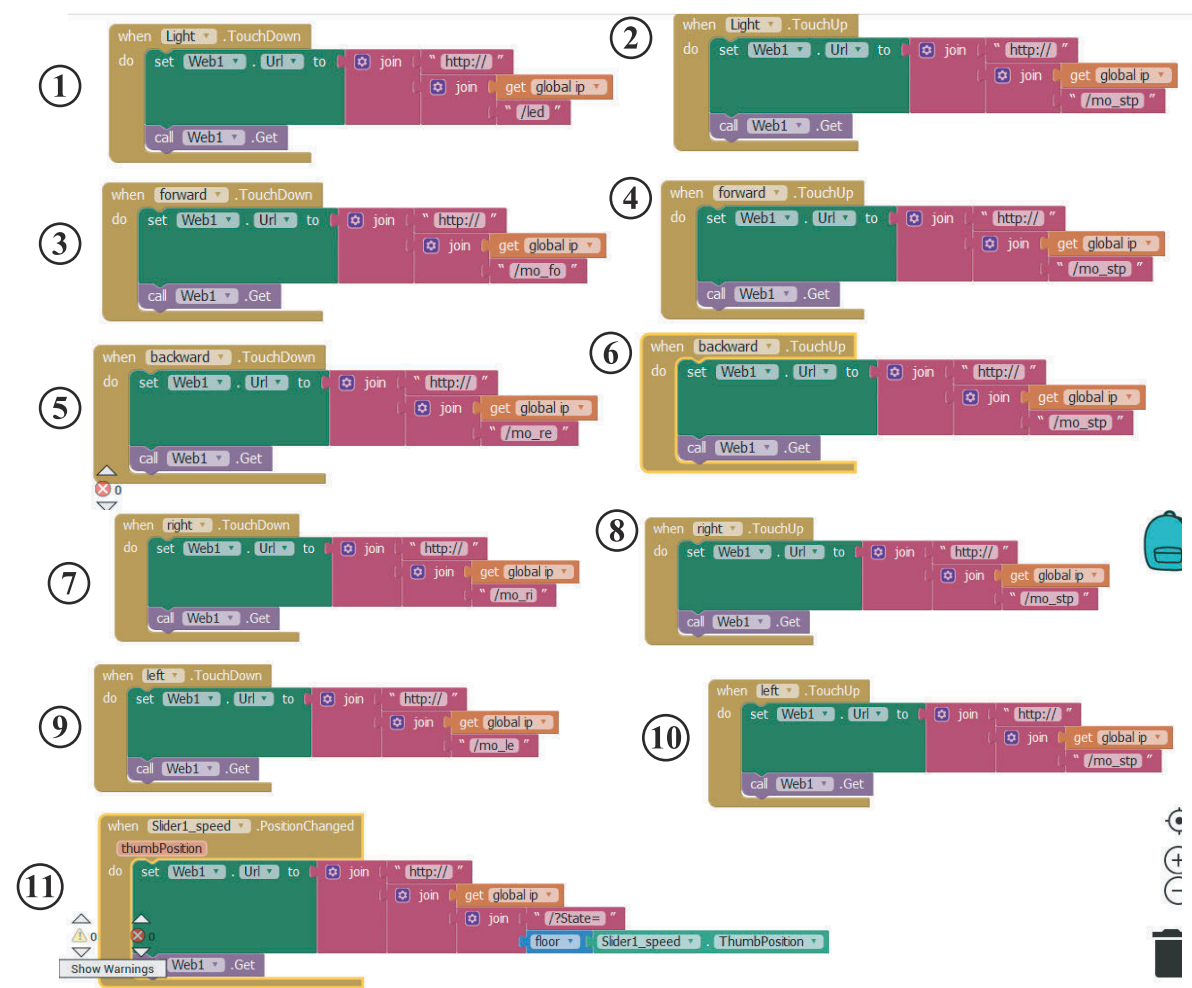
รูปที่ 5 แสดงหน้าต่างแอปพลิเคชันหน้าหลัก

Figure 5. The main page of Application



รูปที่ 6 แสดงหน้าต่างแอปพลิเคชัน เมื่อกดปุ่ม SSID

Figure 6. Application window when pressing the SSID button



รูปที่ 7 แสดงบล็อกคำสั่งของแอปพลิเคชันบนโปรแกรม APP INVENTOR

Figure 7. Block Editor of Application on App Inventor

แอปพลิเคชันบนโทรศัพท์มือถือ

จากรูปที่ 5 จะแสดงหน้าต่างของแอปพลิเคชันที่เป็นหน้าหลัก โดยในหน้าต่างจะประกอบไปด้วย

ปุ่ม LIGHT มีไว้สำหรับเปิดไฟหน้าของตัวหุ่นยนต์

ปุ่ม UP มีไว้สำหรับควบคุมหุ่นยนต์ให้วิ่งไปด้านหน้า

ปุ่ม LEFT มีไว้สำหรับควบคุมหุ่นยนต์ให้เลี้ยวซ้าย

ปุ่ม RIGHT มีไว้สำหรับควบคุมหุ่นยนต์ให้เลี้ยวขวา

ปุ่ม DOWN มีไว้สำหรับควบคุมหุ่นยนต์ให้ถอยหลัง

ปุ่ม SLIDER มีไว้สำหรับควบคุมหุ่นยนต์ให้วิ่งเร็วหรือช้าตามการปรับสไลด์ โดยถ้าปรับไปทางด้านซ้ายจะเป็นการลดความเร็ว ในทางตรงกันข้าม ถ้าปรับไปทางขวา จะเป็นการเร่งความเร็ว

ปุ่ม SSID ใช้สำหรับตั้งชื่อหุ่นยนต์ เมื่อกดปุ่มแล้ว แอปพลิเคชันจะเปลี่ยนไปยังอีกหน้าต่างหนึ่ง ดังรูปที่ 6 โดยของ IP จะเป็นการตั้ง IP Address ให้ตรงกับตัวหุ่นยนต์ ซึ่งในที่นี้จะใช้ 192.168.4.1 โดยถ้าต้องการเปลี่ยน จะต้องทำการเปลี่ยนที่ตัวแอปพลิเคชันและที่ตัวหุ่นยนต์ให้เหมือนกัน ของ SSID เป็นจุดตั้งชื่อหุ่นยนต์

ปุ่ม EXIT เป็นปุ่มออกจากหน้าต่างค่า IP และชื่อ ดังรูปที่ 6 เพื่อกลับไปหน้าจอแอปพลิเคชันหลัก

บล็อกคำสั่งของแอปพลิเคชัน

บล็อกชุดที่ 1 เมื่อทำการกดปุ่ม LIGHT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /led ไปยังตัวหุ่นยนต์เพื่อทำการเปิดไฟหน้า

บล็อกชุดที่ 2 เมื่อไม่กดปุ่ม LIGHT ตัวแอปพลิเคชันจะทำการส่ง

คำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 3 เมื่อทำการกดปุ่ม UP ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_fo ไปยังตัวหุ่นยนต์ เพื่อให้หุ่นยนต์เดินหน้า

บล็อกชุดที่ 4 เมื่อไม่กดปุ่ม UP ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 5 เมื่อทำการกดปุ่ม DOWN ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_re ไปยังตัวหุ่นยนต์ เพื่อให้หุ่นยนต์ถอยหลัง

บล็อกชุดที่ 6 เมื่อไม่กดปุ่ม DOWN ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 7 เมื่อทำการกดปุ่ม RIGHT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_ri ไปยังตัวหุ่นยนต์ เพื่อให้หุ่นยนต์เลี้ยวขวา

บล็อกชุดที่ 8 เมื่อไม่กดปุ่ม RIGHT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 9 เมื่อทำการกดปุ่ม LEFT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_le ไปยังตัวหุ่นยนต์ เพื่อให้หุ่นยนต์เลี้ยวซ้าย

บล็อกชุดที่ 10 เมื่อไม่กดปุ่ม LEFT ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์

บล็อกชุดที่ 11 เมื่อทำการเลื่อนปุ่ม SLIDER ตัวแอปพลิเคชันจะทำการส่งคำสั่ง /mo_stp ไปยังตัวหุ่นยนต์ เพื่อควบคุมความเร็วในการวิ่ง

Mobile application

Figure 5 shows the application window that is the main page, in which the window consists of

LIGHT button is for turning on the headlight of the robot.

UP button is for controlling the robot to run forward.

LEFT button is for controlling the robot to turn left.

RIGHT button is for controlling the robot to turn right.

DOWN button is used to control the robot backwards.

SLIDER button is for controlling the robot to run fast or slow. If, it is adjusted to the left, will reduce the speed. And if adjusted to the right, will be the acceleration.

SSID button is used to name the robot. When the button is pressed, the application will change to another window as shown in Figure 6. The IP box will set the IP Address to match the robot. In this case, 192.168.4.1 will be used. If you want to change, you must change the application and the robot to the same. SSID point is the robot name.

EXIT button is the button to exit the IP and name setting page (see Figure 6) to return to Main application page.

Block editor of the application

Block 1: When pressing LIGHT button, the application will send the command "/led" to the robot. In order to turn on the headlights.

Block 2: When LIGHT button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 3: When UP button is pressed, the application will send the command "/mo_fo" to the robot in order to move forward.

Block 4: When UP button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 5: When pressing DOWN button, the application will send the command "/mo_re" to the robot in order to reverse.

Block 6: When DOWN button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 7: When pressing RIGHT button, the application will send the command "/mo_ri" to the robot to turn right.

Block 8: When RIGHT button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 9: When pressing LEFT button, the application will send the command "/mo_le" to the robot in order to turn left.

Block 10: When LEFT button is not pressed, the application will send the command "/mo_stp" to the robot.

Block 11: When sliding SLIDER button, the application will send sliding value to the robot to control the running speed.