# Basic Kit for Cherokey 4WD SKU:ROB0117

From Robot Wiki

## Contents


- 1 Function Introduction
  - 1.1 STEP 1: Assemble Robot
  - 1.2 STEP2: Debug Motor
  - 1.3 STEP3: Install Cherokey expansion plate
  - 1.4 STEP4: Debug Ultrasonic Sensor and Servo
  - 1.5 STEP 5: Debugging Robot


## Function Introduction

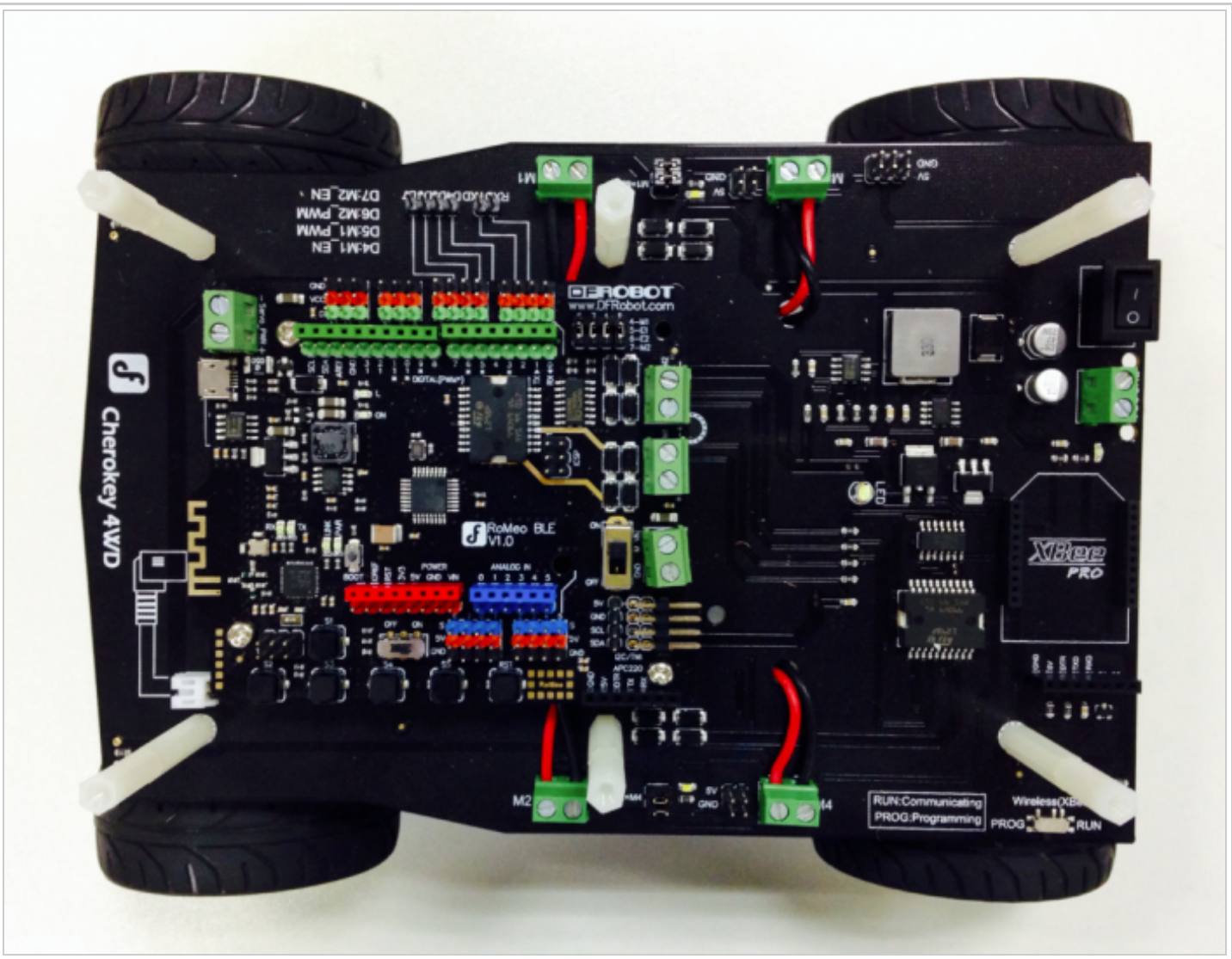
This Kit will teach you how to build a automatic obstacle - avoidance robot which is achieved on the platform of the Turtle Robot,based on ultrasonic sensor as distance measuring device,and combined with servo.

### STEP 1: Assemble Robot

Refer to Instruction Manual
(http://www.dfrobot.com.cn/image/data/ROB0102/Assembly%20tutorial.pdf)

Precautions:

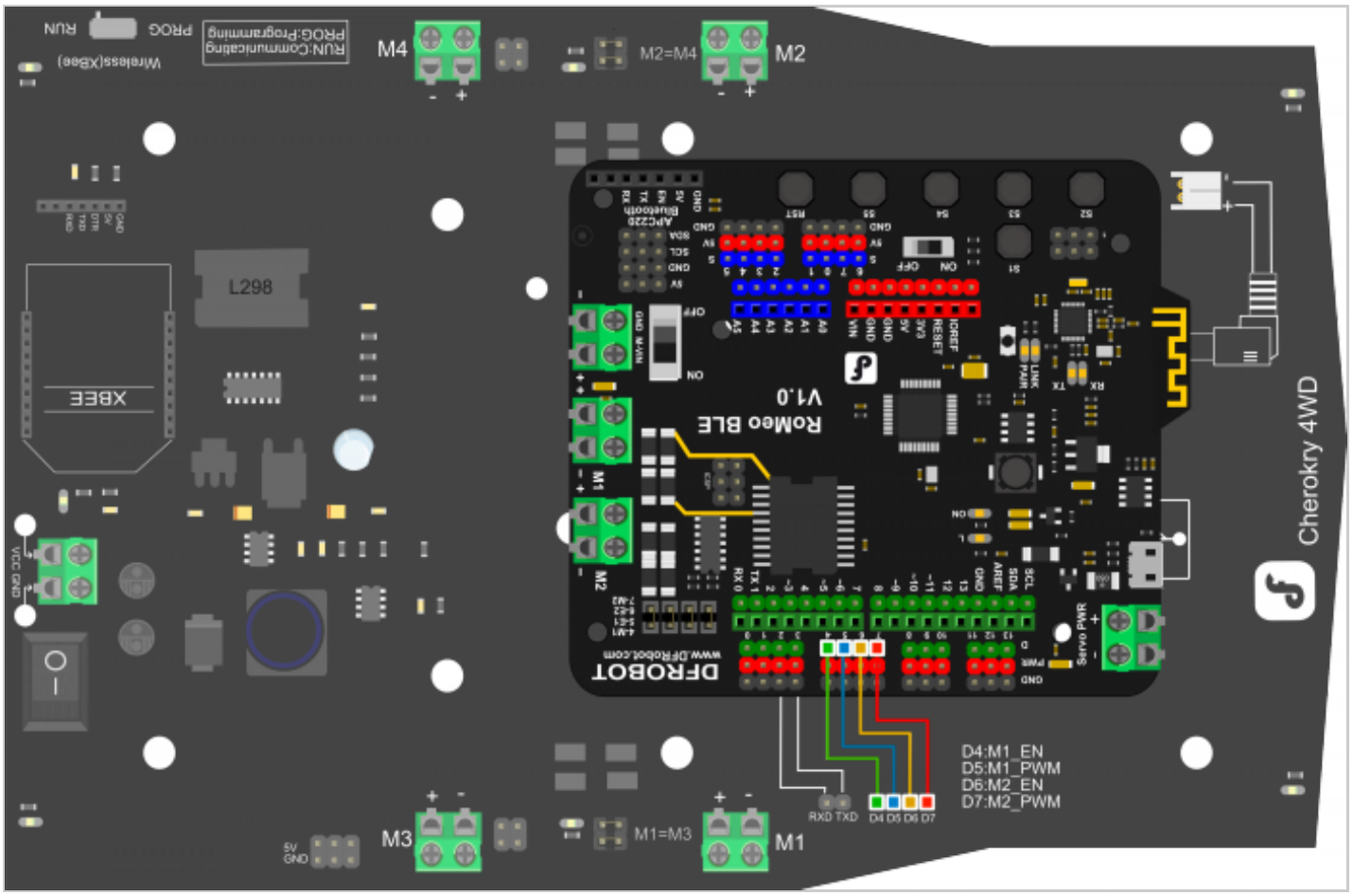
Romeo need to fix to the Cherokey.




fix Romeo BLE

## STEP2：Debug Motor

1. Connect Motor



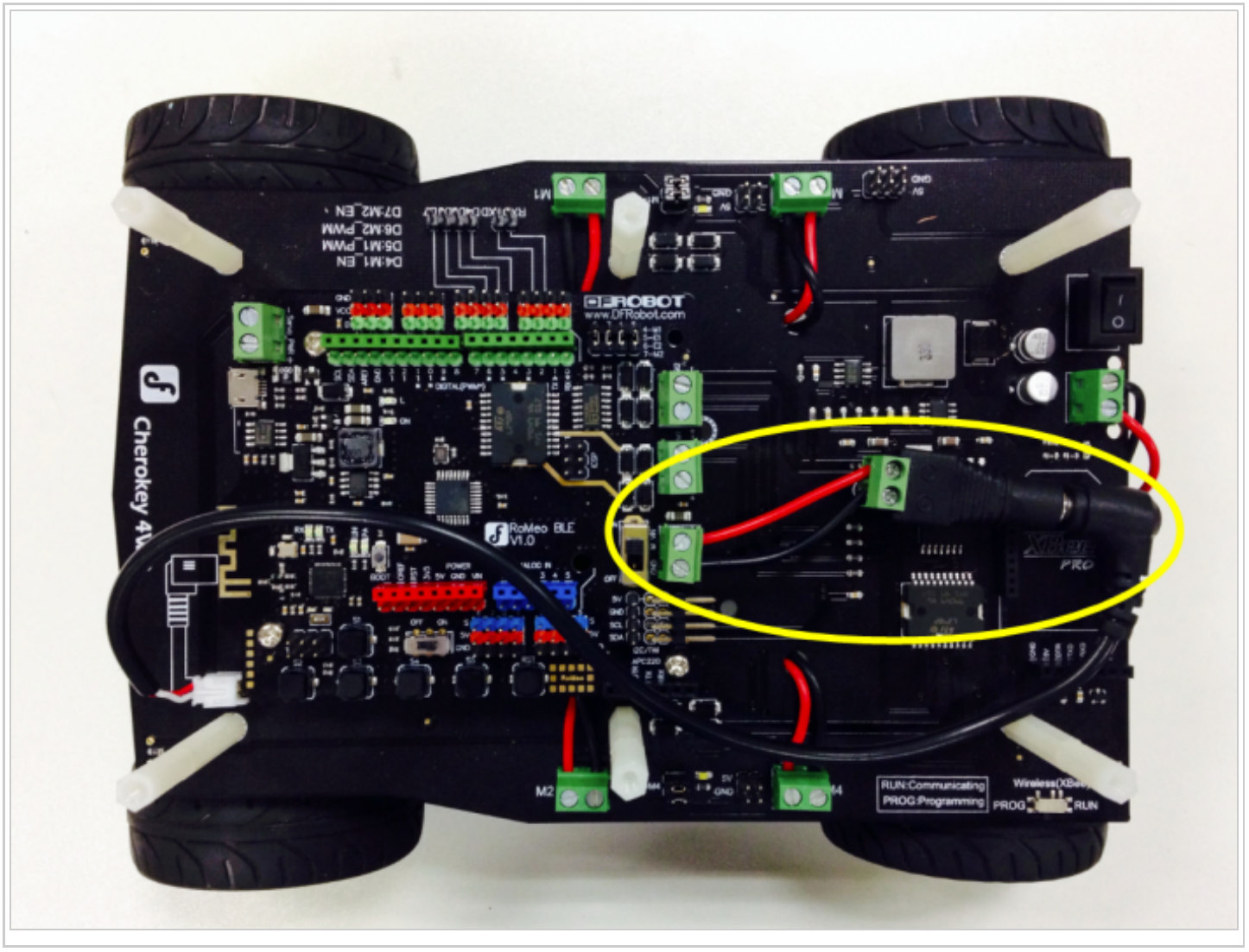

Wiring

## 2. Download Code

```
int speedPin_M1 = 5;     //M1 Speed Control
int speedPin_M2 = 6;     //M2 Speed Control
int directionPin_M1 = 4;     //M1 Direction Control
int directionPin_M2 = 7;     //M1 Direction Control

void setup(){
}

void loop(){
    carAdvance(100,100);
    delay(1000);
    carBack(100,100);
    delay(1000);
    carTurnLeft(250,250);
    delay(1000);
    carTurnRight(250,250);
    delay(1000);
}

void carStop(){                 //  Motor Stop
  digitalWrite(speedPin_M2,0);
  digitalWrite(directionPin_M1,LOW);
  digitalWrite(speedPin_M1,0);
  digitalWrite(directionPin_M2,LOW);
}

void carBack(int leftSpeed,int rightSpeed){         //Move backward
  analogWrite (speedPin_M2,leftSpeed);              //PWM Speed Control
  digitalWrite(directionPin_M1,HIGH);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,HIGH);
```

```
}

void carAdvance(int leftSpeed,int rightSpeed){        //Move forward
  analogWrite (speedPin_M2,leftSpeed);
  digitalWrite(directionPin_M1,LOW);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,LOW);
}

void carTurnLeft(int leftSpeed,int rightSpeed){       //Turn Left
  analogWrite (speedPin_M2,leftSpeed);
  digitalWrite(directionPin_M1,LOW);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,HIGH);
}
void carTurnRight(int leftSpeed,int rightSpeed){       //Turn Right
  analogWrite (speedPin_M2,leftSpeed);
  digitalWrite(directionPin_M1,HIGH);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,LOW);
}
```

===STEP 3: Fit Battery===



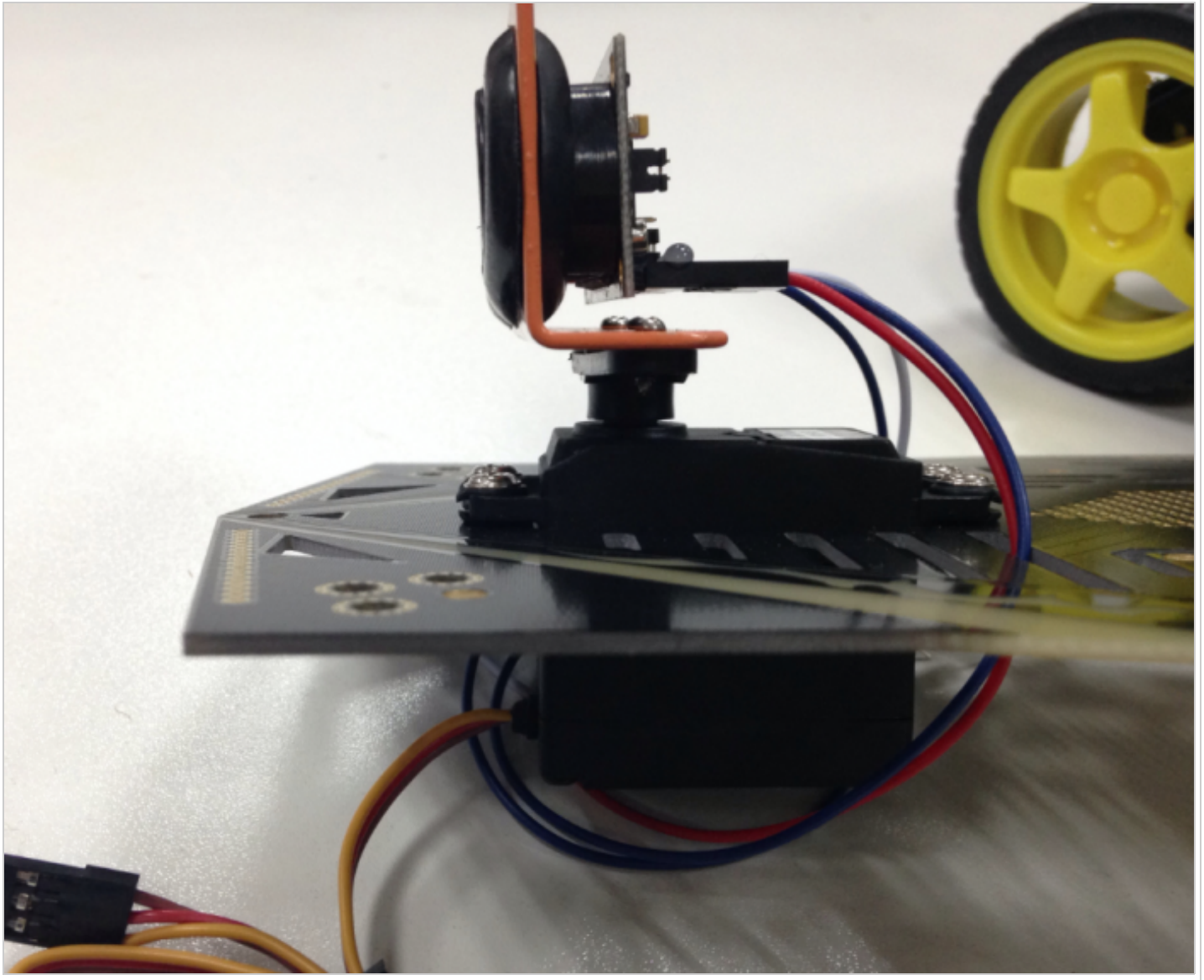

## STEP3: Install Cherokey expansion plate

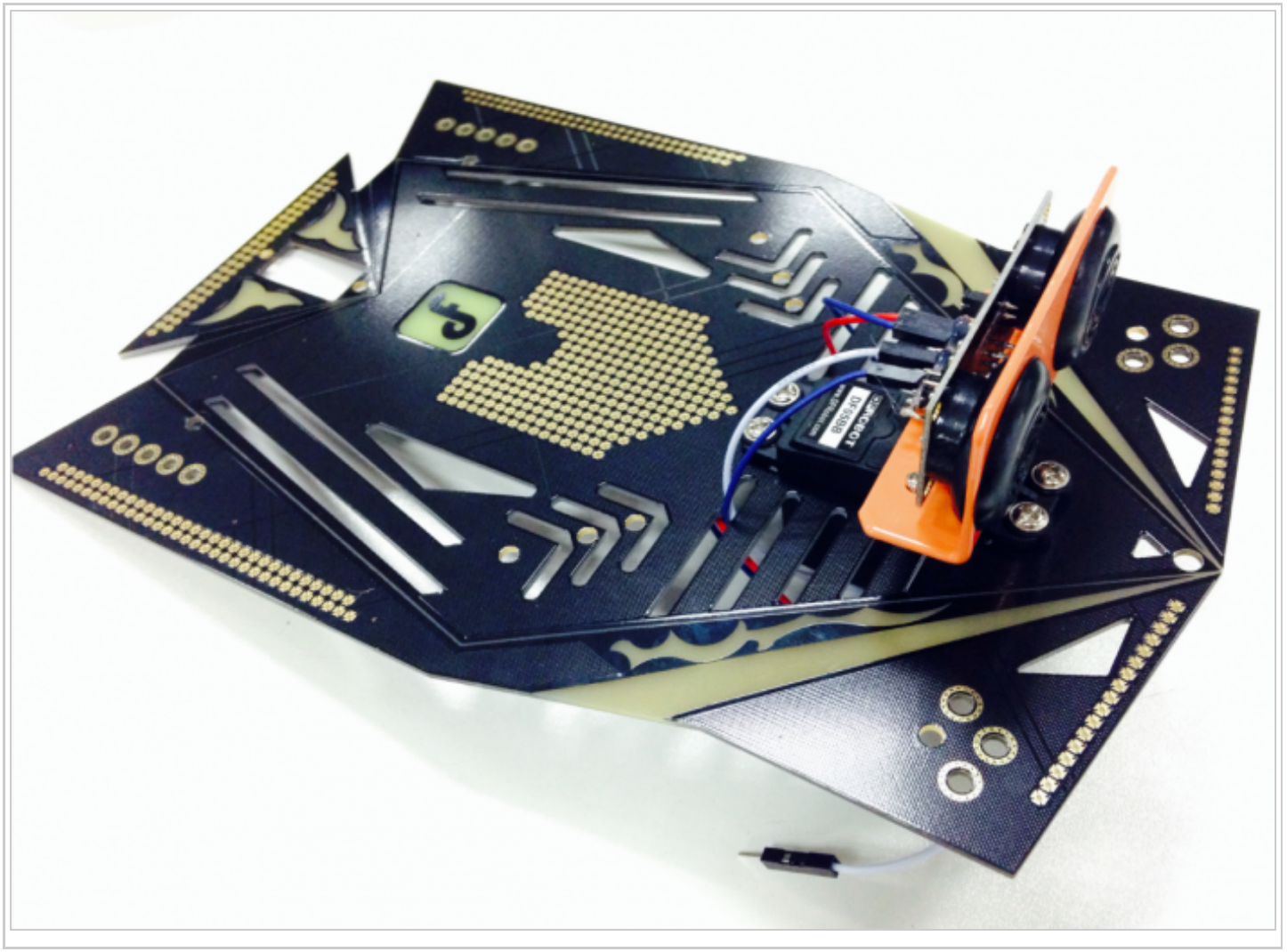
## 1. Prepare the Materials


## 2. Fixed Ultrasonic Sensor Position

Please see the Installation Manual
(http://www.dfrobot.com.cn/images/upload/File/20141030183325g7lofm.pdf)

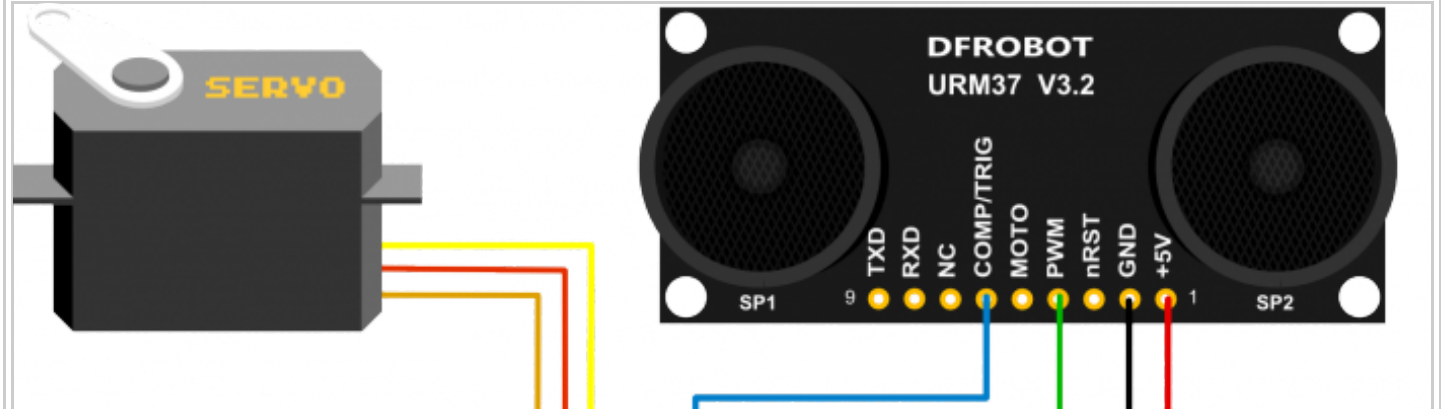
## 3. Fixed Servo Position

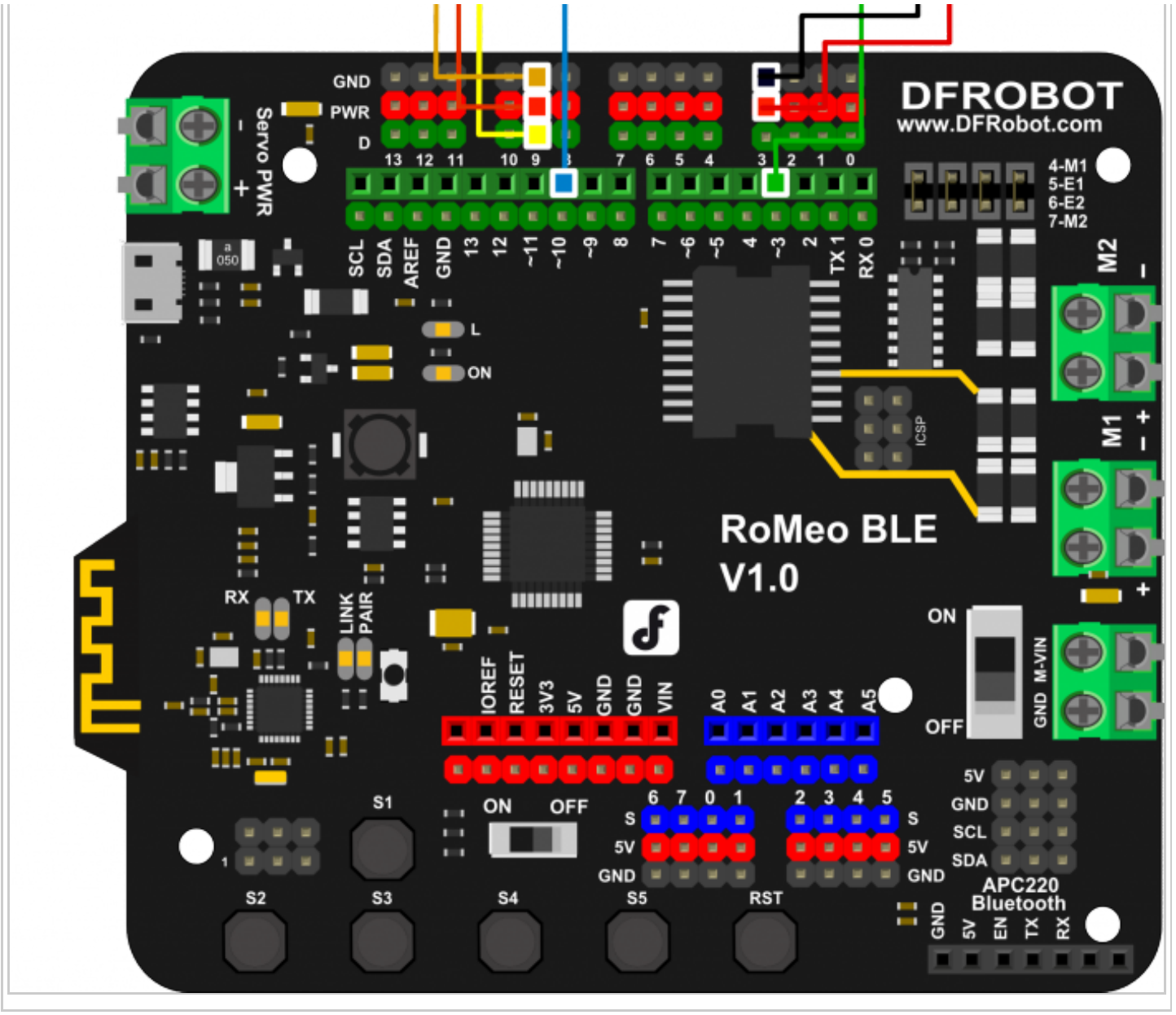


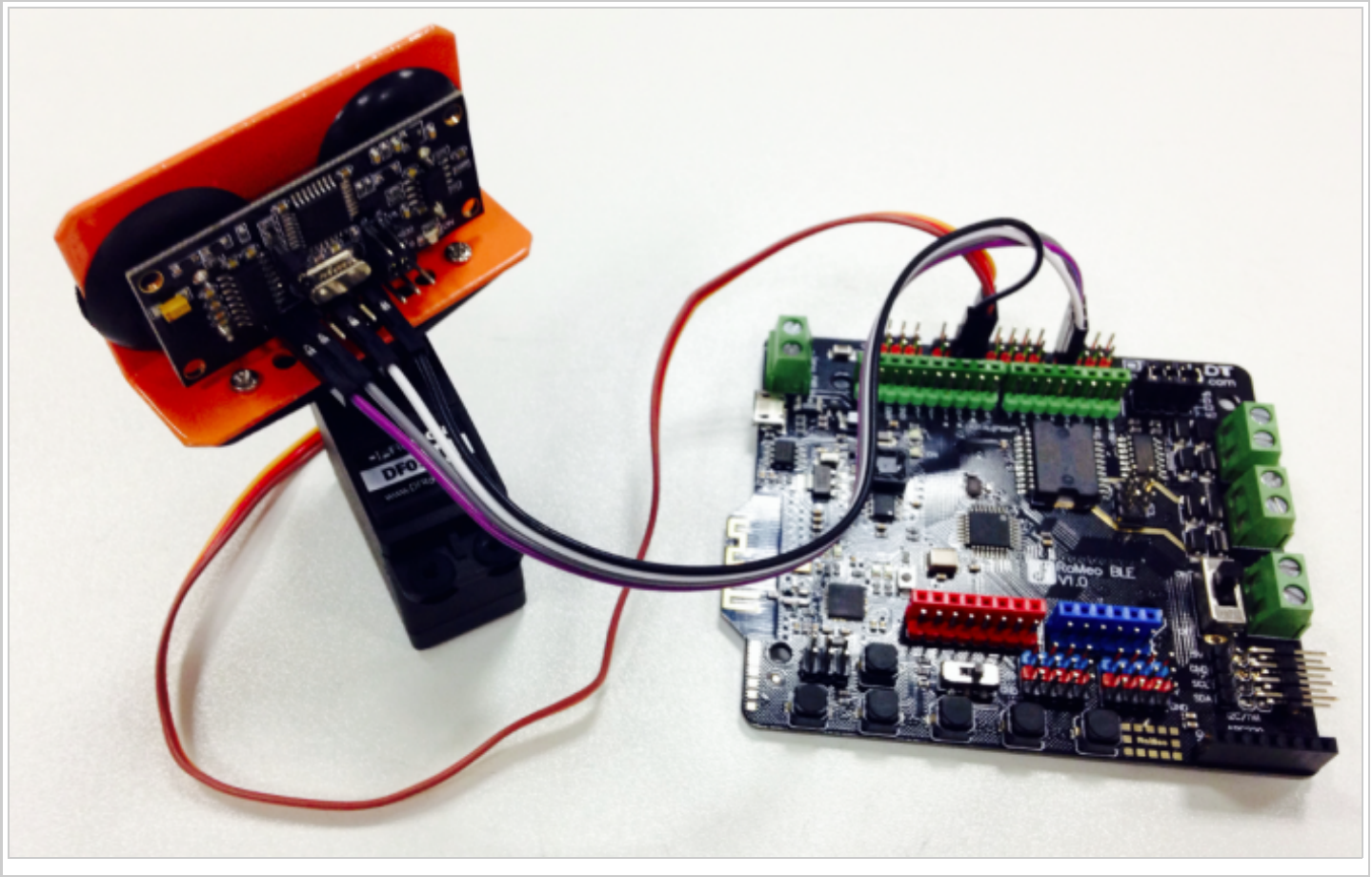


## STEP4: Debug Ultrasonic Sensor and Servo

1. Hardware Connection




DFROBOT
www.DFRobot.com
RoMeo BLE
V1.0
Servo PWR
GND
PWR
D
4-M1
5-E1
6-E2
7-M2
M2
M1
ICSP
RX
TX
LINK
PAIR
IOREF
RESET
3V3
5V
GND
VIN
A0
A1
A2
A3
A4
A5
ON
OFF
M-VIN
S1
S2
S3
S4
S5
RST
APC220
Bluetooth
EN



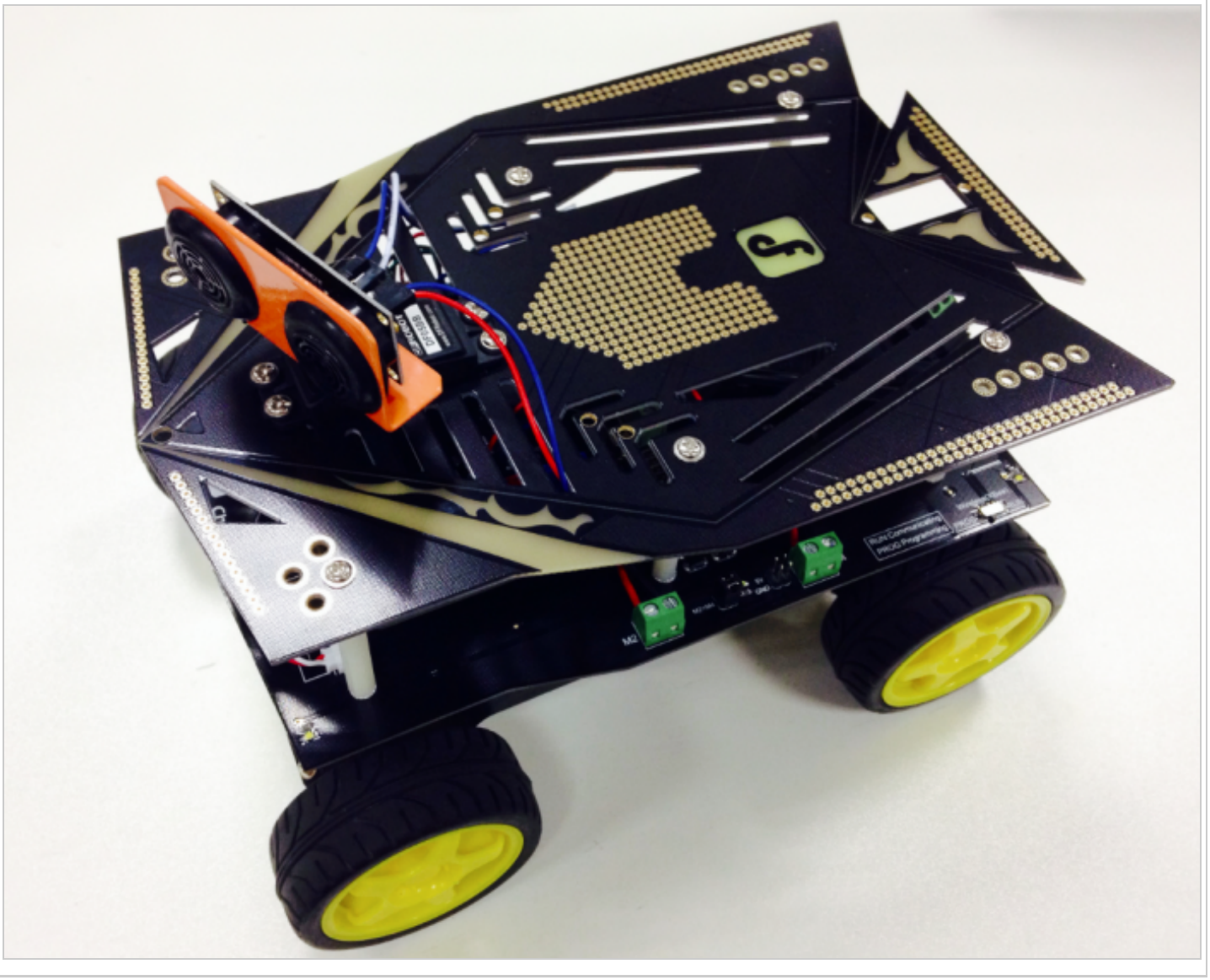
2. Download Code
Install the library firstly. Metro libray (http://www.dfrobot.com.cn/images/upload/File/20141031110246wu4065.rar)

```
#include <Servo.h>
#include <Metro.h>
Metro measureDistance = Metro(50);
Metro sweepServo = Metro(20);

unsigned long actualDistance = 0;

Servo myservo;  // create servo object to control a servo
int pos = 60;
int sweepFlag = 1;


int URPWM = 3; // PWM Output 0—25000US, Every 50US represent 1cm
int URTRIG= 10; // PWM trigger pin
uint8_t EnPwmCmd[4]={0x44,0x02,0xbb,0x01};    // distance measure command

void setup(){                                 // Serial initialization
  myservo.attach(9);
  Serial.begin(9600);                         // Sets the baud rate to 9600
  SensorSetup();
}

void loop(){
 if(measureDistance.check() == 1){
      actualDistance = MeasureDistance();
//      Serial.println(actualDistance);
//      delay(100);
 }

 if(sweepServo.check() == 1){
      servoSweep();
 }
```

```
}

void SensorSetup(){
  pinMode(URTRIG,OUTPUT);                     // A low pull on pin COMP/TRIG
  digitalWrite(URTRIG,HIGH);                  // Set to HIGH
  pinMode(URPWM, INPUT);                      // Sending Enable PWM mode command
  for(int i=0;i<4;i++){
      Serial.write(EnPwmCmd[i]);
   }
}

int MeasureDistance(){        // a low pull on pin COMP/TRIG  triggering a sensor reading
    digitalWrite(URTRIG, LOW);
    digitalWrite(URTRIG, HIGH);               // reading Pin PWM will output pulses
    unsigned long distance=pulseIn(URPWM,LOW);
    if(distance==50000){              // the reading is invalid.
      Serial.print("Invalid");
    }else{
      distance=distance/50;           // every 50us low level stands for 1cm
    }
    return distance;
}

void servoSweep(){
  if(sweepFlag ){
     if(pos>=60 && pos<=120){
        pos=pos+1;                                  // in steps of 1 degree
        myservo.write(pos);                         // tell servo to go to position in variable 'pos'
    }
      if(pos>119)  sweepFlag = false;                       // assign the variable again
  }else {
      if(pos>=60 && pos<=120){
        pos=pos-1;
        myservo.write(pos);
      }
      if(pos<61)  sweepFlag = true;
   }
}
```

## STEP 5: Debugging Robot

1. Fix the upper Plate



## 2. Download Code

```
#include <Servo.h>
#include <Metro.h>
Metro measureDistance = Metro(50);
Metro sweepServo = Metro(20);

int speedPin_M1 = 5;     //M1 Speed Control
int speedPin_M2 = 6;     //M2 Speed Control
int directionPin_M1 = 4;     //M1 Direction Control
int directionPin_M2 = 7;     //M1 Direction Control
unsigned long actualDistance = 0;

Servo myservo;  // create servo object to control a servo
int pos = 60;
int sweepFlag = 1;

int URPWM = 3; // PWM Output 0—25000US, Every 50US represent 1cm
int URTRIG= 10; // PWM trigger pin
uint8_t EnPwmCmd[4]={0x44,0x02,0xbb,0x01};    // distance measure command

void setup(){                                 // Serial initialization
  myservo.attach(9);
  Serial.begin(9600);                         // Sets the baud rate to 9600
  SensorSetup();
}

void loop(){

 if(measureDistance.check() == 1){
```

```
      actualDistance = MeasureDistance();
//      Serial.println(actualDistance);
//      delay(100);
 }

 if(sweepServo.check() == 1){
      servoSweep();
 }

 if(actualDistance <= 30){
            myservo.write(90);
            if(pos>=90){
                  carBack(100,100);
//                  Serial.println("carBack");
                  delay(300);
                  carTurnRight(250,250);
//                  Serial.println("carTurnRight");
                  delay(500);
               }else{
                   carBack(100,100);
//                   Serial.println("carBack");
                   delay(300);
                   carTurnLeft(250,250);
//                   Serial.println("carTurnLeft");
                   delay(500);
               }
    }else{
                   carAdvance(100,100);
//                   Serial.println("carAdvance");
                   delay(300);
     }
}

void SensorSetup(){
  pinMode(URTRIG,OUTPUT);                     // A low pull on pin COMP/TRIG
  digitalWrite(URTRIG,HIGH);                  // Set to HIGH
  pinMode(URPWM, INPUT);                      // Sending Enable PWM mode command
  for(int i=0;i<4;i++){
      Serial.write(EnPwmCmd[i]);
   }
}

int MeasureDistance(){  // a low pull on pin COMP/TRIG  triggering a sensor reading
    digitalWrite(URTRIG, LOW);
    digitalWrite(URTRIG, HIGH);               // reading Pin PWM will output pulses
    unsigned long distance=pulseIn(URPWM,LOW);
    if(distance==50000){              // the reading is invalid.
      Serial.print("Invalid");
    }else{
      distance=distance/50;           // every 50us low level stands for 1cm
    }
    return distance;
}

void carStop(){                 //  Motor Stop
  digitalWrite(speedPin_M2,0);
  digitalWrite(directionPin_M1,LOW);
  digitalWrite(speedPin_M1,0);
  digitalWrite(directionPin_M2,LOW);
}

void carBack(int leftSpeed,int rightSpeed){         //Move forward
  analogWrite (speedPin_M2,leftSpeed);              //PWM Speed Control
  digitalWrite(directionPin_M1,HIGH);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,HIGH);
}

void carAdvance(int leftSpeed,int rightSpeed){       //Move backward
  analogWrite (speedPin_M2,leftSpeed);
  digitalWrite(directionPin_M1,LOW);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,LOW);
```

```
}

void carTurnLeft(int leftSpeed,int rightSpeed){              //Turn Left
  analogWrite (speedPin_M2,leftSpeed);
  digitalWrite(directionPin_M1,LOW);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,HIGH);
}
void carTurnRight(int leftSpeed,int rightSpeed){            //Turn Right
  analogWrite (speedPin_M2,leftSpeed);
  digitalWrite(directionPin_M1,HIGH);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,LOW);
}

void servoSweep(){
  if(sweepFlag){
     if(pos>=60 && pos<=120){
        pos=pos+1;                                  // in steps of 1 degree
        myservo.write(pos);                         // tell servo to go to position in variable 'pos'
    }
      if(pos>119)   sweepFlag = false;                        // assign the variable again
  }
   else {
      if(pos>=60 && pos<=120){
        pos=pos-1;
        myservo.write(pos);
      }
      if(pos<61)   sweepFlag = true;
    }
}
```

Your own car was born!

Retrieved from "https://www.dfrobot.com/wiki/index.php?title=Basic_Kit_for_Cherokey_4WD_SKU:ROB0117&oldid=28238"

---

- This page was last modified on 10 February 2015, at 06:02.
- This page has been accessed 1,633 times.