

Button start

The purpose of the experiment:

After uploading the Button start program, turn on the power switch and press the start button K1. After a short whistle, the BatCar starts to move forward, backward, left turn, right turn, rotate left, rotate right.

List of components required for the experiment:

BatCar *1

USB cable *1



Experimental code analysis:

```
int Left_motor_back = 9;  
int Left_motor_go = 5;  
int Right_motor_go = 6;  
int Right_motor_back = 10;  
int Right_motor_en = 8;  
int Left_motor_en = 7;
```

```
/*Set Button port*/
```

```
int key=4;
```

```
/*Set BUZZER port*/
```

```
int beep=3;
```

```
void setup()
```

```
{
```

```
  //Initialize motor drive for output mode
```

```
  pinMode(Left_motor_go,OUTPUT);
```

```
  pinMode(Left_motor_back,OUTPUT);
```

```

pinMode(Right_motor_go,OUTPUT);
pinMode(Right_motor_back,OUTPUT);

pinMode(key,INPUT);// Set button as input
pinMode(beep,OUTPUT); // Set buzzer as output

digitalWrite(key,HIGH);//Initialize button
digitalWrite(beep,HIGH);// set buzzer mute
}

void run(int time)    // run
{
    digitalWrite(Right_motor_go,HIGH); // right motor go ahead
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200);//PWM--Pulse Width Modulation(0~255). It can
    be adjusted to control speed.
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,HIGH); // set left motor go ahead
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200);//PWM--Pulse Width Modulation(0~255). It can
    be adjusted to control speed.
    analogWrite(Left_motor_back,0);
    delay(time * 100); //Running time can be adjusted
}

void brake(int time)    //stop
{
    digitalWrite(Right_motor_go,LOW);
    digitalWrite(Right_motor_back,LOW);
    digitalWrite(Left_motor_go,LOW);
    digitalWrite(Left_motor_back,LOW);
    delay(time * 100);
}

void left(int time) //turn left
{

```

```
digitalWrite(Right_motor_go,HIGH);// right motor go ahead
digitalWrite(Right_motor_back,LOW);
analogWrite(Right_motor_go,200);// PWM--Pulse Width Modulation(0~255)
control speed
analogWrite(Right_motor_back,0);
digitalWrite(Left_motor_go,LOW); // left motor stop
digitalWrite(Left_motor_back,LOW);
analogWrite(Left_motor_go,0);
analogWrite(Left_motor_back,0);
delay(time * 100);
}
```

```
void spin_left(int time) //Left rotation
{
digitalWrite(Right_motor_go,HIGH);// right motor go ahead
digitalWrite(Right_motor_back,LOW);
analogWrite(Right_motor_go,200);// PWM--Pulse Width Modulation(0~255)
control speed
analogWrite(Right_motor_back,0);
digitalWrite(Left_motor_go,LOW); // left motor back off
digitalWrite(Left_motor_back,HIGH);
analogWrite(Left_motor_go,0);
analogWrite(Left_motor_back,200);// PWM--Pulse Width Modulation(0~255)
control speed
delay(time * 100);
}
```

```
void right(int time) //turn right
{
digitalWrite(Right_motor_go,LOW); // right motor stop
digitalWrite(Right_motor_back,LOW);
analogWrite(Right_motor_go,0);
analogWrite(Right_motor_back,0);
digitalWrite(Left_motor_go,HIGH);// left motor go ahead
digitalWrite(Left_motor_back,LOW);
analogWrite(Left_motor_go,200);// PWM--Pulse Width Modulation(0~255) control
speed
```

```

analogWrite(Left_motor_back,0);
delay(time * 100);
}

void spin_right(int time) //Right rotation
{
    digitalWrite(Right_motor_go,LOW); // right motor back off
    digitalWrite(Right_motor_back,HIGH);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,200);// PWM--Pulse Width Modulation(0~255)
control speed
    digitalWrite(Left_motor_go,HIGH);// left motor go ahead
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200);// PWM--Pulse Width Modulation(0~255) control
speed
    analogWrite(Left_motor_back,0);
    delay(time * 100);
}

void back(int time) //back off
{
    digitalWrite(Right_motor_go,LOW); //right motor back off
    digitalWrite(Right_motor_back,HIGH);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,150);// PWM--Pulse Width Modulation(0~255)
control speed
    digitalWrite(Left_motor_go,LOW); //left motor back off
    digitalWrite(Left_motor_back,HIGH);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,150);// PWM--Pulse Width Modulation(0~255)
control speed
    delay(time * 100);
}

void keysacn()
{
    int val;

```

```

val=digitalRead(key);// Reads the button ,the level value assigns to val
while(digitalRead(key))// When the button is not pressed
{
    val=digitalRead(key);
}
while(!digitalRead(key))// When the button is pressed
{

    delay(10); //delay 10ms

    val=digitalRead(key);// Reads the button ,the level value assigns to val
    if(val==LOW) //Double check the button is pressed
    {

        digitalWrite(beep,LOW);//The buzzer sounds
        delay(100);//delay 100ms

        while(!digitalRead(key)) //Determine if the button is released or not

            digitalWrite(beep,HIGH);//mute
    }
    else
        digitalWrite(beep,HIGH);//mute
    }
}

void loop()
{
    delay(2000); //delay 2s
    keysacn();

    back(10); //back off for 1s
    brake(5); //stop for 0.5s
    run(10);//go ahead for 1s
    brake(5); //stop for 0.5s
    left(10);//turn left for 1s
    right(10);//turn right for 1s
    spin_left(20);//Left rotation for 2s

```

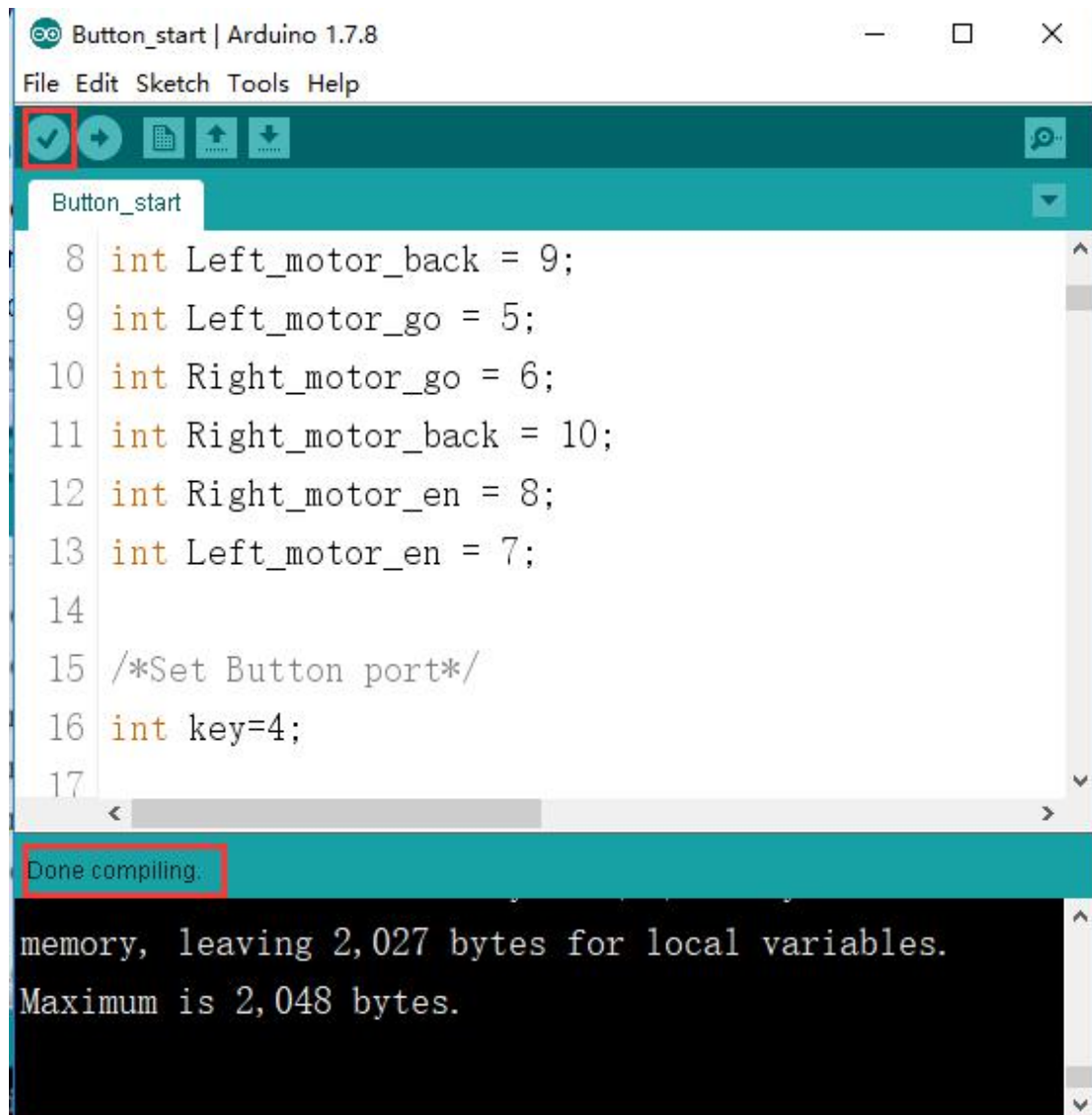
```
spin_right(20); //Right rotation for 2s
```

```
brake(5); //stop for 0.5s
```

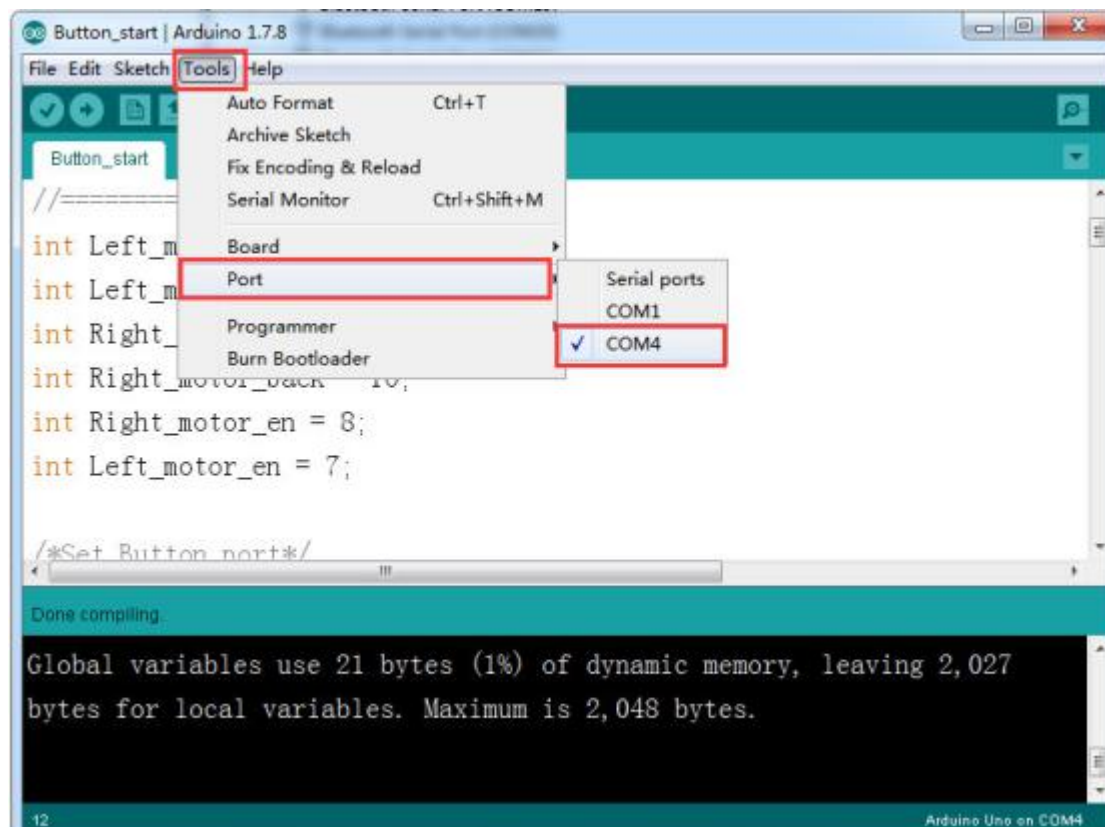
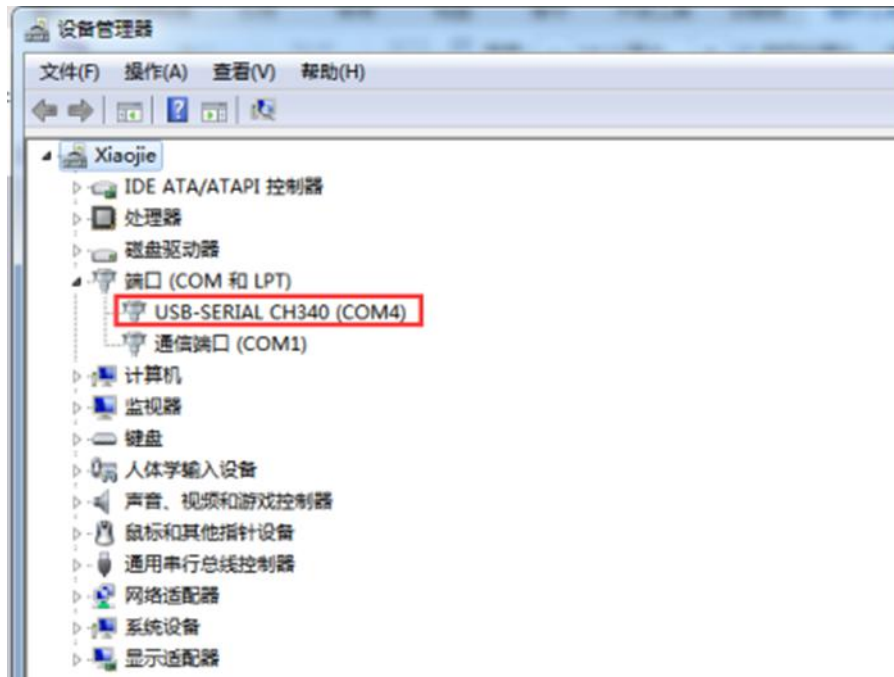
```
}
```

Experimental steps:

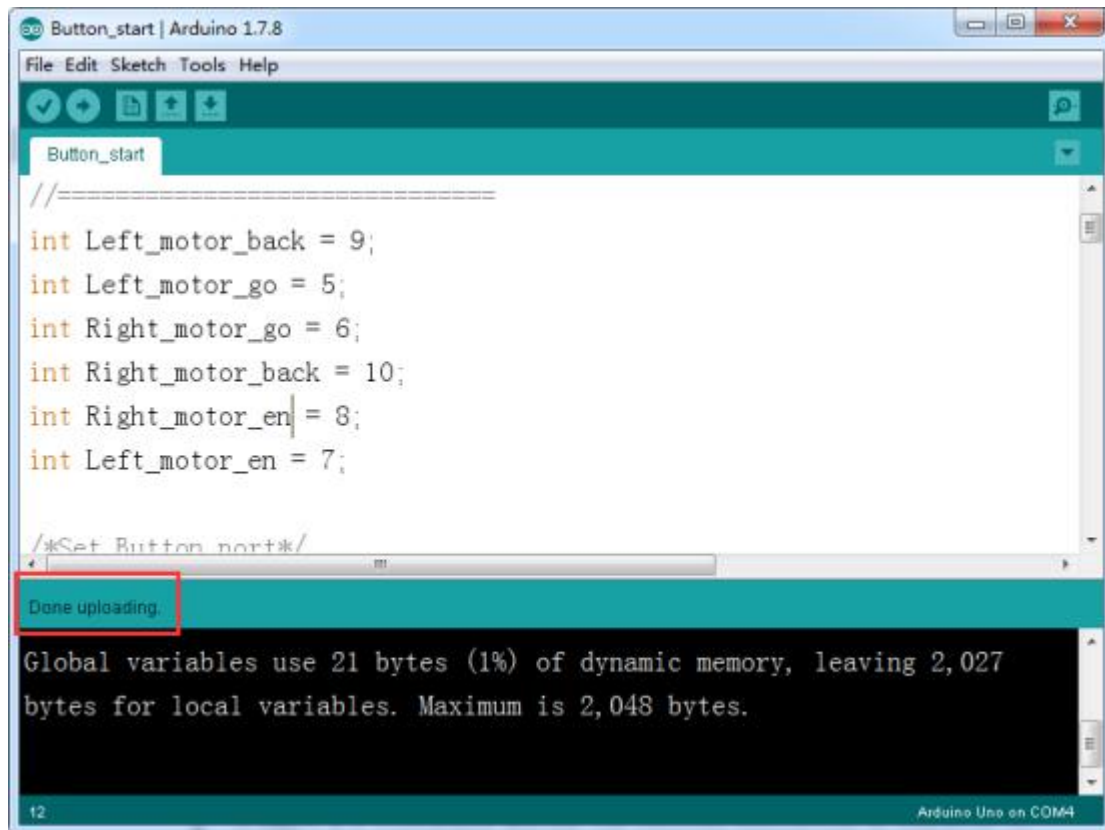
1. We need to open the code of this experiment: **Button_start.ino**, click “ ✓ ” under the menu bar to compile the code, and wait for the word "**Done compiling** " in the lower right corner, as shown in the figure below.



2. In the menu bar of Arduino IDE, we need to select **Tools** --- **Port** --- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



3. After the selection is completed, you need to click “→” under the menu bar to upload the code to the Arduino UNO board. When the word “**Done uploading**” appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



4. Unplug the USB cable, put the BatCar in an open place, turn on the power switch, and the BatCar is still at rest until the start button K1 is pressed. After a short whistle, the BatCar starts to move forward and backward. Turn left, turn right, etc.

