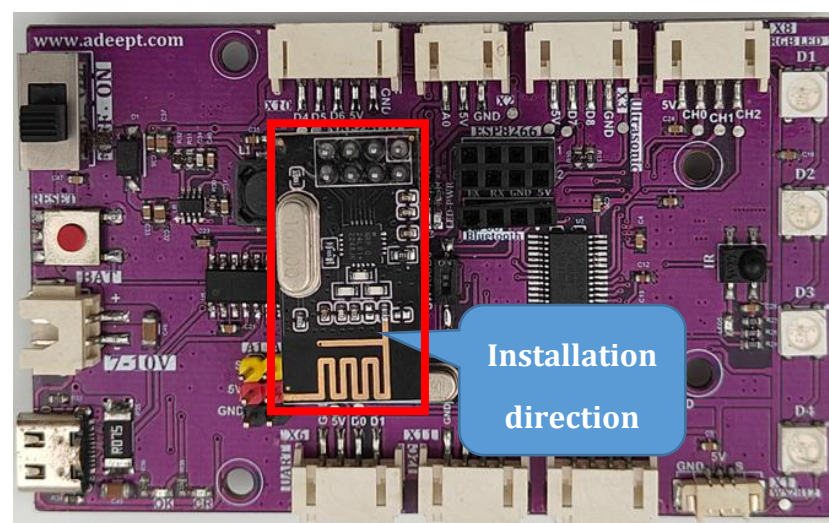
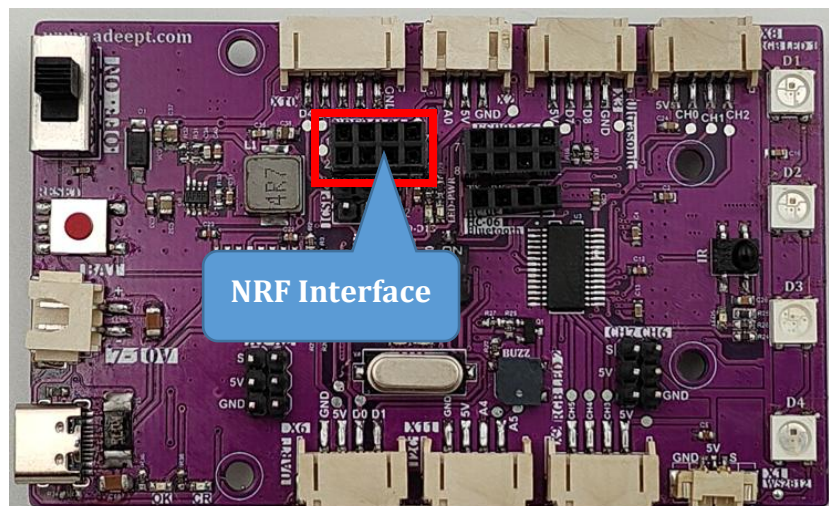


## Lesson 19 How to use the Handle Controller to control the car

In this tutorial, we learn how to use the **Handle Controller** to control the car.

### 19.1 Install an external NRF module.



## 19.2 Introduce the communication principle of NRF module

The communication principle of NRF modules (such as NRF24L01) is based on radio frequency (RF) technology. These modules use specific communication protocols, such as SPI (Serial Peripheral Interface), to communicate with microcontrollers or other devices. Here is the basic principle of communication for NRF modules:

**RF Communication Frequency Band:** NRF modules operate in the 2.4GHz RF frequency band, which is a frequency band allowed for short-range wireless communication and does not require special licenses.

**Data Encapsulation:** To send data, it is first encapsulated into data packets. Data packets typically include the destination address, data content, error detection, and other information.

**Channels and Addresses:** NRF modules can be set to different communication channels and addresses to avoid interference or differentiate between multiple devices. Both the transmitter and receiver must be configured with the same channel and address to communicate.

**Transmission and Reception:** NRF modules can operate in transmit mode and receive mode. In transmit mode, they convert data packets into RF signals and transmit them to the destination address. In receive mode, they listen for data packets from other devices.

**Data Transmission:** In transmit mode, the module converts data packets into RF signals and transmits them. In receive mode, it receives RF signals and converts them back into data packets.

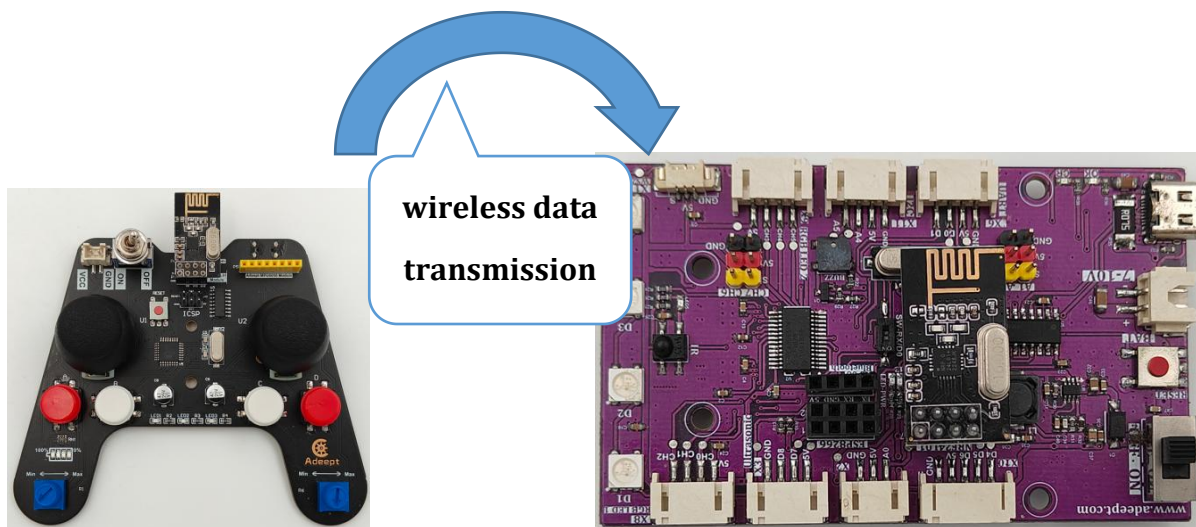
**Acknowledgment and Retransmission:** NRF modules support automatic acknowledgment and retransmission to ensure reliable data transmission. If the sender does not receive an acknowledgment from the receiver, it will attempt to resend the data packet.

**Error Detection:** The modules use techniques such as CRC (Cyclic Redundancy Check) to detect errors in the data packet transmission to ensure data integrity.

**Power Management:** NRF modules typically have low power consumption features and can enter sleep modes when not actively communicating to save energy.

**Protocol Stack:** NRF modules typically run specific communication protocol stacks. For example, NRF24L01 uses Nordic Semiconductor's Enhanced ShockBurst protocol.

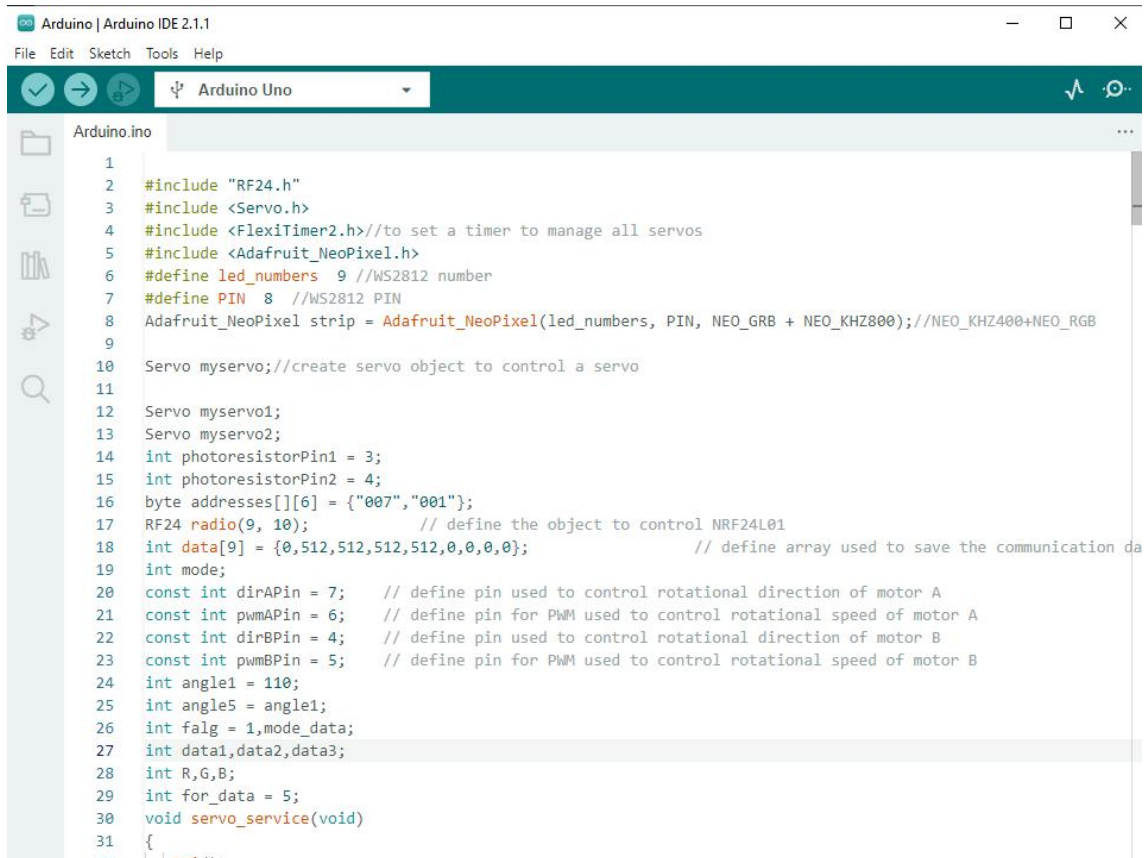
In summary, NRF modules achieve wireless data transmission in the 2.4GHz RF frequency band through RF communication. They can be configured with channels and addresses, support data acknowledgment and retransmission for reliable communication. These modules are suitable for various applications, including sensor networks, remote controls, and IoT devices.



## 19.3 How to use NRF to control the car

1. Connect your computer and Adeept Robot Control Board with a USB cable.

2. Open “19\_NRF\_Control\_Car/Arduino” folder in “[Adeept\\_UnoCar-B/Code](#)”, double-click “[Arduino.ino](#)”.



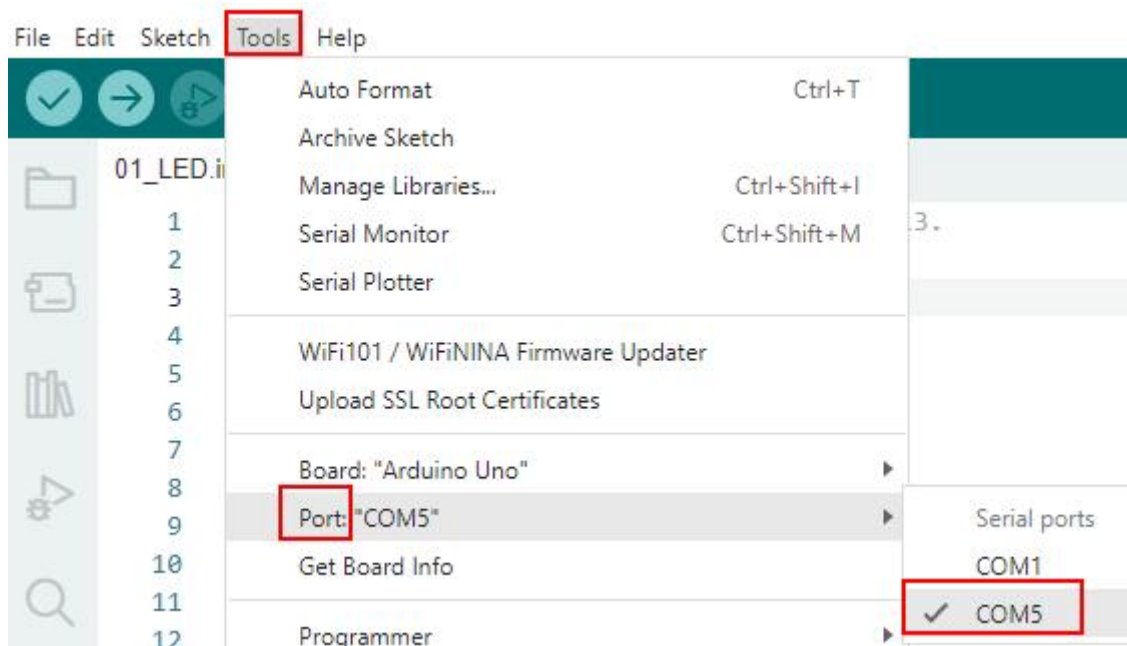
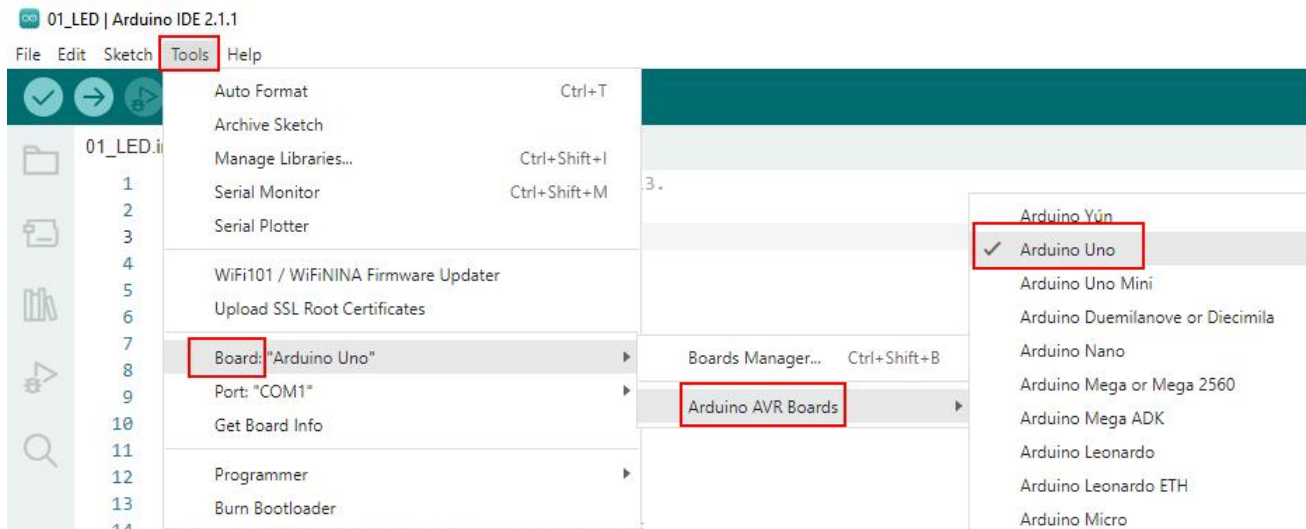
```
1
2 #include "RF24.h"
3 #include <Servo.h>
4 #include <FlexiTimer2.h> //to set a timer to manage all servos
5 #include <Adafruit_NeoPixel.h>
6 #define led_numbers 9 //WS2812 number
7 #define PIN 8 //WS2812 PIN
8 Adafruit_NeoPixel strip = Adafruit_NeoPixel(led_numbers, PIN, NEO_GRB + NEO_KHZ800); //NEO_KHZ400+NEO_RGB
9
10 Servo myservo; //create servo object to control a servo
11
12 Servo myservo1;
13 Servo myservo2;
14 int photoresistorPin1 = 3;
15 int photoresistorPin2 = 4;
16 byte addresses[][6] = {"007", "001"};
17 RF24 radio(9, 10); // define the object to control NRF24L01
18 int data[9] = {0, 512, 512, 512, 512, 0, 0, 0, 0}; // define array used to save the communication data
19 int mode;
20 const int dirAPin = 7; // define pin used to control rotational direction of motor A
21 const int pwmAPin = 6; // define pin for PWM used to control rotational speed of motor A
22 const int dirBPin = 4; // define pin used to control rotational direction of motor B
23 const int pwmBPin = 5; // define pin for PWM used to control rotational speed of motor B
24 int angle1 = 110;
25 int angle5 = angle1;
26 int falg = 1, mode_data;
27 int data1, data2, data3;
28 int R, G, B;
29 int for_data = 5;
30 void servo_service(void)
31 {
32 }
```


3. Select development board and serial port.

Board: [Tools](#)---->[Board](#)---->[Arduino AVR Boards](#)---->[Arduino Uno](#)

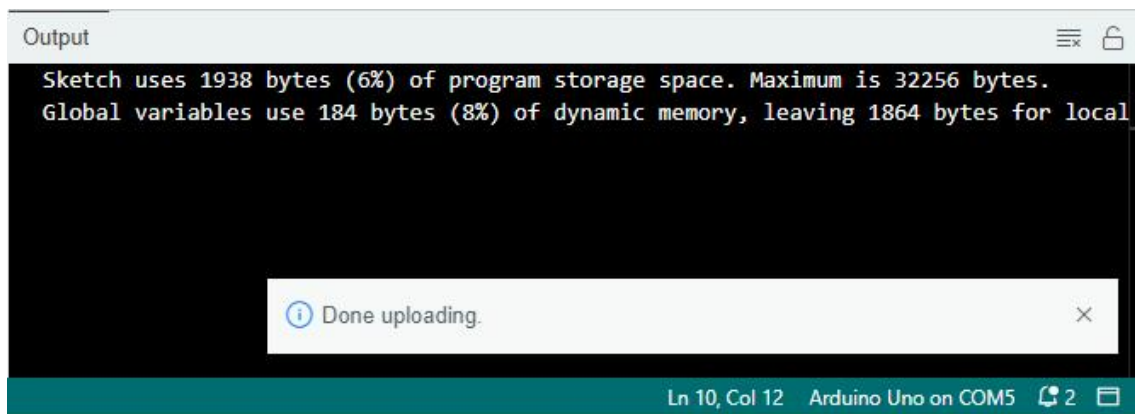
Port: [Tools](#) ---->[Port](#)---->[COMx](#)

Note: The port number will be different in different computers.



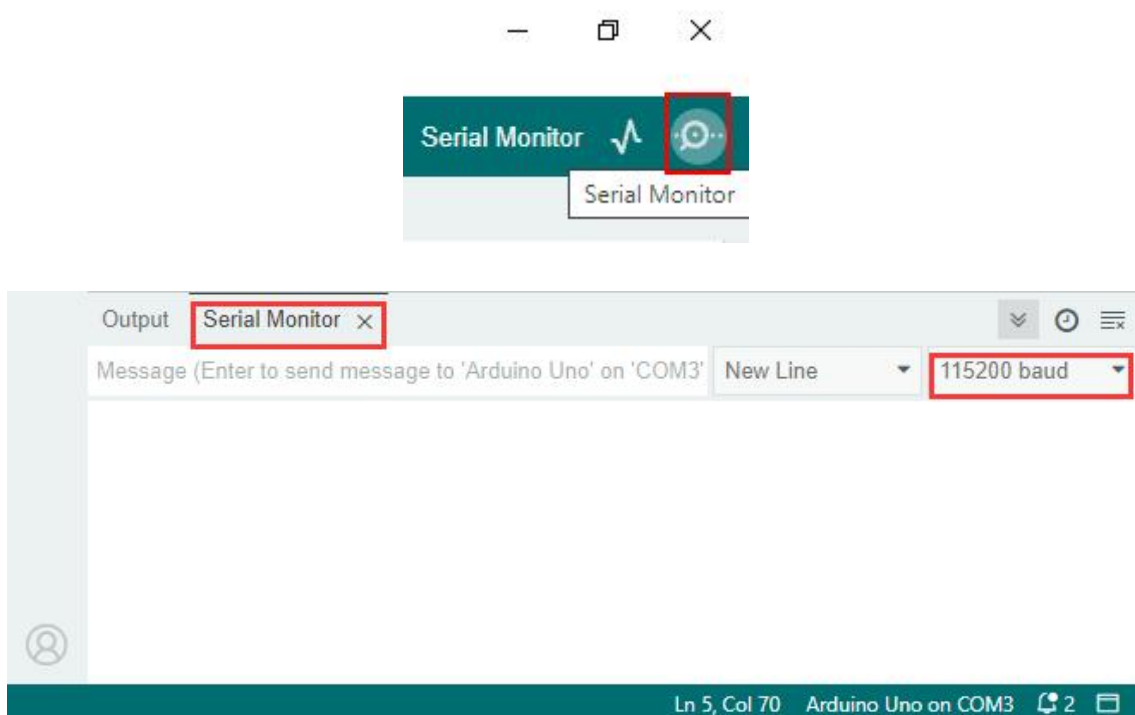
4. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



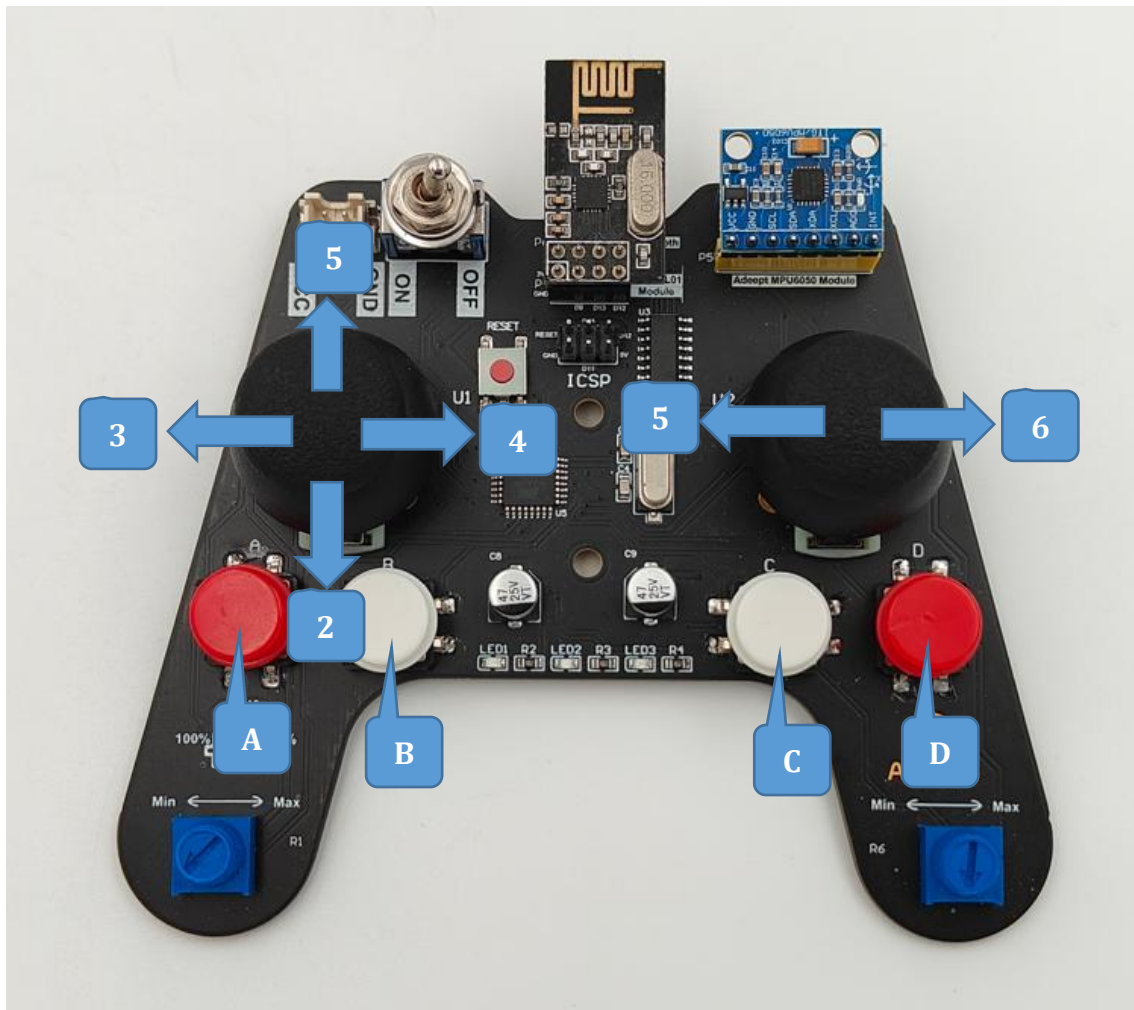


After assembling the car, please use the 18650 battery to provide power when uploading the program, otherwise the program may not be uploaded successfully due to excessive load.

5. click Serial Monitor, Set the baud rate as 115200.



6. Moving the joystick of the handle remote control or pressing the button will have different functions.



Joystick	Function	Joystick	Function
1	Forward	4	Turn Right
2	Backward	5	Head Turn Left
3	Turn Left	6	Head Turn Right

Button	Function
A	Start joystick control mode/stop B, C, D functions.

B	The car start the Light Tracking Function.
C	The car start the Gyro Control Mode.
D	The car start the Avoid Obstacles Function.