Cat(dog) detection

We can view the WiFi camera screen through the APP, and the car will react according to the recognition results.

```
Since the previous tutorials have explained the basic knowledge of all car
modules, the car control tutorial will not repeat it, and will mainly explain the
ideas for implementing the car functions!
```

Device connection

Hardware connection

Use Type-B data cable to connect Arduino Uno and computer.

Before this, you need to unplug the connection between the WiFi camera and Roboduino, otherwise the serial port will be occupied and the computer cannot recognize the serial port number of Arduino Uno.

Software connection

Open the "Arduino IDE" software and select the model and serial port number corresponding to the development board.

Implementation ideas

Analyze the data sent by the WiFi camera through the serial port and control the car according to the data.

Code analysis

Here is only a brief introduction to the code content. For detailed code, it is recommended to refer to the corresponding code file, which is provided in the download area!

1. esp32_wifi

- Define a macro `WIFI_MODE` to select different WiFi modes based on preprocessor directives (#if)

```
#if MODE_AP_STA
#define WIFI_MODE '2'

#elif MODE_STA
#define WIFI_MODE '1'

#elif MODE_AP
#define WIFI_MODE '0'

#else
#define WIFI_MODE '2'
#endif
```

- Enumerate AI modes, including cat and dog detection, face detection, color detection, etc.

```
typedef enum AI_mode_t
{
    Nornal_AI = 0,//No detection
    Cat_Dog_AI,   //Cat and Dog Detection
    FACE_AI,      //People Detection
    COLOR_AI ,   //Color Detection
    REFACE_AI,   //Face Recognition
    QR_AI = 5,   //QR code recognition
    AI_MAX      //Maximum
}AI_mode;
```

- `SET_ESP_WIFI_MODE` function sets the WiFi mode of the ESP32 camera.

```
void SET_ESP_WIFI_MODE(void) //Setting Mode Selection
{
  //Select mode STA+AP mode coexistence
  sprintf(send_buf,"wifi_mode:%c",WIFI_MODE);
  ESPWIFISerial.print(send_buf);
  memset(send_buf,0,sizeof(send_buf));

  delay(2000);// Wait for reset to restart successfully
}
```

- The `SET_ESP_AI_MODE` function sets the AI mode of the ESP32 camera.

```
void SET_ESP_AI_MODE(AI_mode Mode) //Setting AI Mode
{

  sprintf(send_buf,"ai_mode:%d",Mode);
  ESPWIFISerial.print(send_buf);
  memset(send_buf,0,sizeof(send_buf));
  runmode = Mode;

  delay(2000);//Wait for reset to restart successfully
}
```

2. bsp_RGB

- Include libraries such as `Adafruit_NeoPixel.h` and motor_car.hpp header files

```
#include <stdio.h>
#include <string.h>
#include <Arduino.h>
#include <Adafruit_NeoPixel.h>
#include "motor_car.hpp"
```

- Enumerate RGB light status, off, red, green, blue, yellow, etc.

```
// 枚举常见颜色 Enumerate common colors
enum ColorType {
  BLACK,
  RED,
  GREEN,
  BLUE,
  YELLOW,
  MAGENTA,
  CYAN,
  WHITE,
};
```

- Define RGB light control pins

```
// 定义RGB控制引脚和数量 Define RGB control pins and quantity
#define RGB_PIN 6
#define RGB_NUM 1
```

- Create an instance of `Adafruit_NeoPixel`

```
// 创建Adafruit_NeoPixel类的实例 Create an instance of the Adafruit_NeoPixel class
Adafruit_NeoPixel RGB = Adafruit_NeoPixel(RGB_NUM, RGB_PIN, NEO_GRB +
NEO_KHZ800);
```

- Initialize RGB light control

```
void RGB_init()
{
    RGB.begin();  // 初始化RGB Initialize RGB
    RGB.show();   // 刷新RGB显示 Refresh RGB display
    if(runmode == COLOR_AI || runmode==REFACE_AI )
    {
        setRGBColor(GREEN); //默认关闭  Default off

     }
     else
     {
        setRGBColor(BLACK); //默认关闭  Default off
     }

}
```

- Set the color of the RGB light

```
/**
 * @brief 设置RGB显示的颜色 Set RGB display color
 * @param color: 显示的颜色 Set the color
 * @retval 无 None
 */
void setRGBColor(ColorType color) {
  switch (color) {
    case RED:
      RGB.setPixelColor(0, RGB.Color(255, 0, 0));
      RGB.show();
      break;
    case GREEN:
      RGB.setPixelColor(0, RGB.Color(0, 255, 0));
      RGB.show();
      break;
    case BLUE:
      RGB.setPixelColor(0, RGB.Color(0, 0, 255));
      RGB.show();
      break;
    case YELLOW:
      RGB.setPixelColor(0, RGB.Color(255, 255, 0));
      RGB.show();
      break;
    case MAGENTA:
      RGB.setPixelColor(0, RGB.Color(255, 0, 255));
      RGB.show();
      break;
    case CYAN:
      RGB.setPixelColor(0, RGB.Color(0, 255, 255));
      RGB.show();
      break;
    case WHITE:
      RGB.setPixelColor(0, RGB.Color(255, 255, 255));
      RGB.show();
      break;
    default:
      RGB.setPixelColor(0, RGB.Color(0, 0, 0));
      RGB.show();
      break;
  }
}
```

2. Cat_dog_esp32.ino

- Import header files, including the implementation of ESP32's WiFi function, motor control, button management, and RGB light control.

```
#include <stdio.h>
#include "esp32_wifi.hpp"
#include "motor_car.hpp"
#include "esp_key.hpp"
#include "bsp_RGB.hpp"
```

- `AI_set_mode` sets the current AI mode to `Cat_Dog_AI` (cat and dog recognition). `cmd_flag` is an external variable used to indicate the current command status.

```
#define AI_set_mode Cat_Dog_AI  //Setting AI Mode
extern uint8_t cmd_flag;
```

- `mode_chage()` is a mode switching function that sets the value of `cmd_flag` according to the current operating mode to indicate different processing modes.

```
void mode_chage()
{
  if(runmode == Nornal_AI)
  {
    cmd_flag = 2;//进入透传模式 Enter transparent mode
  }
  else if(runmode == REFACE_AI )
  {
    cmd_flag = 3;//解析人脸识别数据   Parsing facial recognition data
  }
  else if(runmode == QR_AI )
  {
    cmd_flag = 4;//解析二维码数据  Parsing QR code data
  }
  else
  {
    cmd_flag = 5;//其它AI模式数据   Other AI mode data
   }
}
```

- The `decect_cat_dog()` function is used to turn on the yellow RGB light, play the buzzer tone, and then turn off the RGB light when a cat or dog is detected.

```
#define BUZZER_PIN 10
void decect_cat_dog()
{
  setRGBColor(YELLOW); //亮黄色 Bright yellow
  tone(BUZZER_PIN, 1000);  // 播放频率为1000HZ的音调 Play a 1000 Hz tone
  delay(300);
  noTone(BUZZER_PIN);  // 停止发声 Stop play
  delay(300);
  setRGBColor(BLACK);  //RGB关闭 RGB Off
}
```

- The `setup()` function is the entry point for initializing the program, including the initialization of the buzzer, button, serial port, WiFi, AI mode, motor and RGB light, and calls the `mode_chage` function to set the command flag.

```
void setup()
{
  pinMode(BUZZER_PIN, OUTPUT); //蜂鸣器初始化 Buzzer initialization
  init_key();//按键初始化 Button initialization
  serial_init();//串口初始化 Serial port initialization
  SET_ESP_WIFI_MODE();//设置wifi模式 Set Wi-Fi mode

  SET_ESP_AI_MODE(AI_set_mode);//设置AI模式 Setting AI Mode

  Motor_init();//电机初始化 Motor initialization
  RGB_init();//RGB初始化 RGB initialization

  mode_chage();//根据模式选择解析数据办法 Choose the method to parse data according
to the mode
```

```
  //setCarMove(FORWARD,35); //35-255
}
```

- In the main loop, when the operation mode is color recognition or face recognition, the state detection of the virtual button is called. If new data is received, the `decect_cat_dog()` method is called to perform cat and dog detection.

```
void loop()
{
 //Key logo and RGB light color Only these two modes will use key interaction
 if(runmode == COLOR_AI || runmode==REFACE_AI )
 {
  key_goto_state();//虚拟按键 Virtual buttons
 }

 if(newlines == 1)//接收到新数据 New data received
 {
   newlines = 0;
   //识别到猫狗 蜂鸣器叫 Cat or dog detected, buzzer sounds
   decect_cat_dog();
 }
}
```

Experimental results

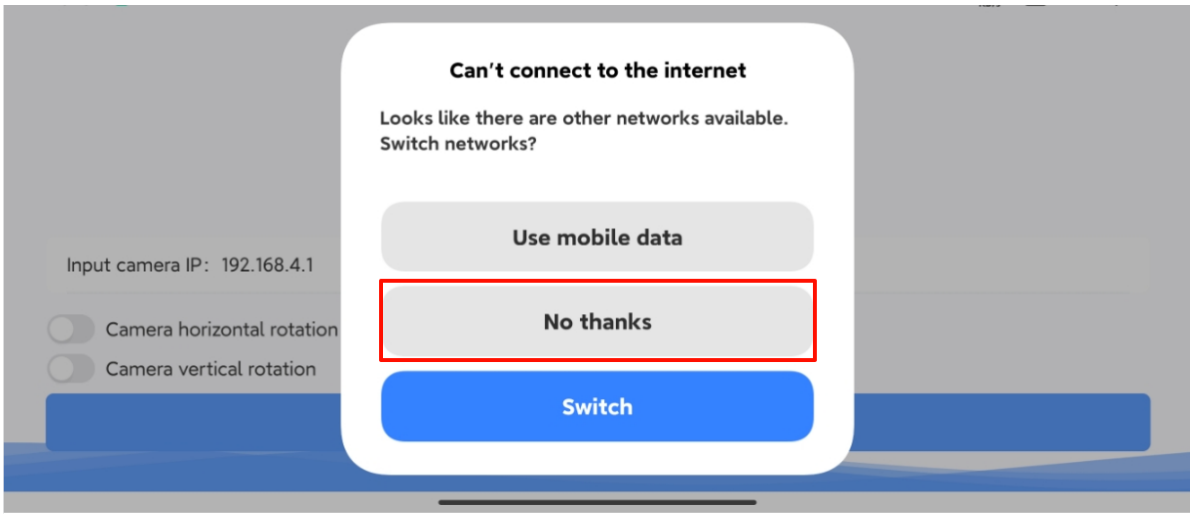
After compiling the program successfully, upload the code to the Arduino Uno development board, disconnect the car from the computer, and connect the WiFi camera to the serial port on the expansion board.

After the program starts, we can observe the camera screen through the YahboomCAM APP.

APP connection

Connect the mobile phone to the hotspot of the WiFi camera (the name of the built-in hotspot: Yahboom_ESP32_WIFI), and then open the YahboomCam software.

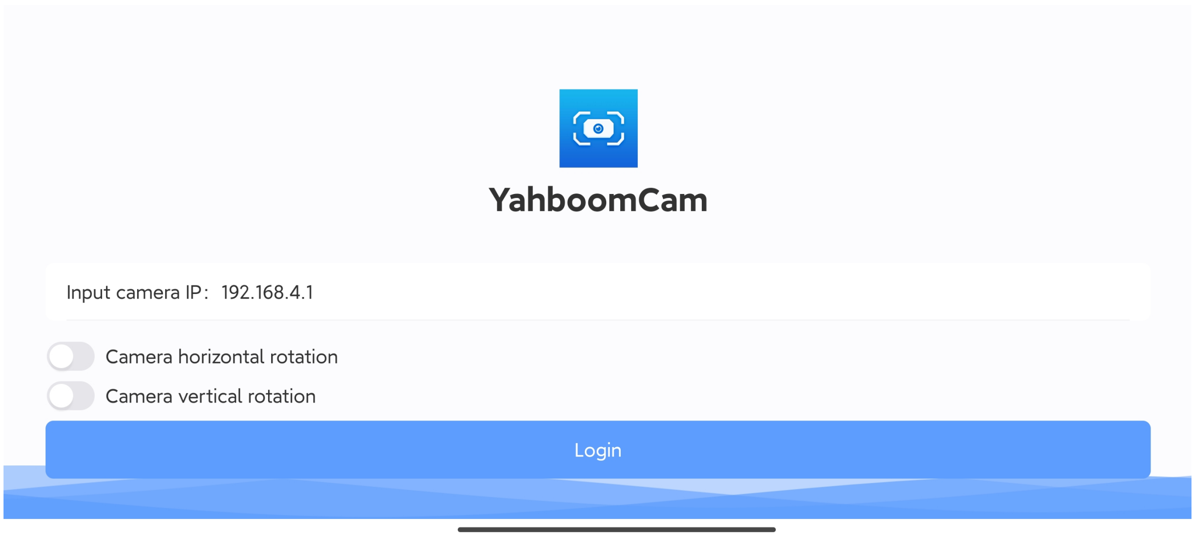
```
Some mobile phones will prompt for connecting to a hotspot without a network, and
we need to click to keep connected!
```


Can't connect to the internet

Looks like there are other networks available. Switch networks?

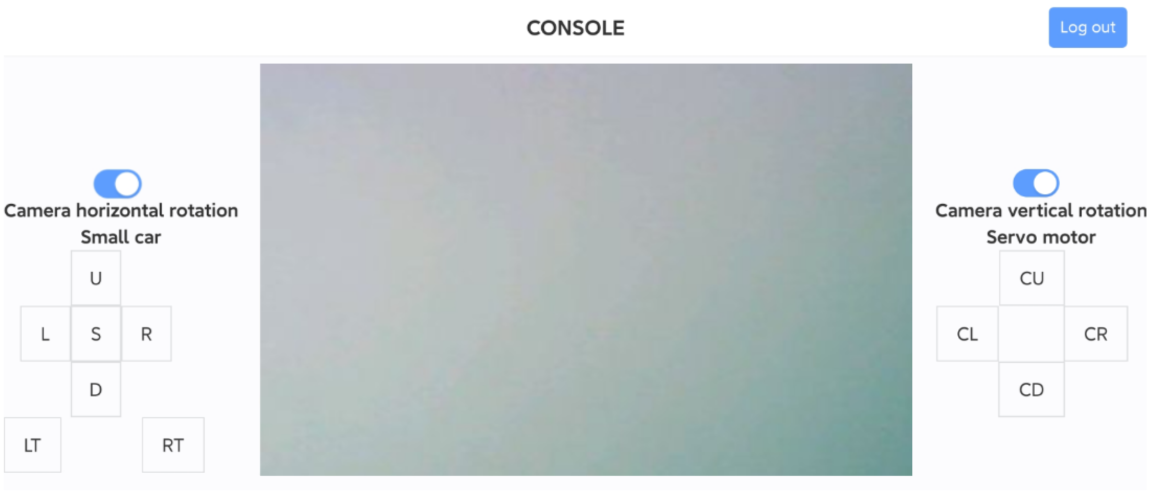
Enter IP:192.168.4.1 in the YahboomCam software, then click Login to enter the APP control interface.

```
The IP of the WiFi camera's built-in hotspot is 192.168.4.1
```



After entering the APP interface, the APP will display the camera screen.

```
If no screen is displayed, check whether the phone is connected to the WiFi
camera hotspot normally
```



Effect

Aim the camera at the picture or cat or dog to be identified. When a cat or dog is identified, the car buzzer beeps and the RGB light turns yellow.

Cat

Dog

Notes

1. The screen needs to be turned on to start recognition. If the screen is turned off, the recognition will also be turned off.
2. Due to the model, this mode can only detect cats and dogs, but cannot distinguish between cats and dogs.
3. Make sure the direction of the camera image is positive, and turn on the vertical rotation and horizontal rotation switches on the APP.