K210-QR code recognition

The tutorial mainly demonstrates the function of using a balance car with a K210 visual module to control the balance car with a QR code.

```
The tutorial only introduces the standard library project code
```

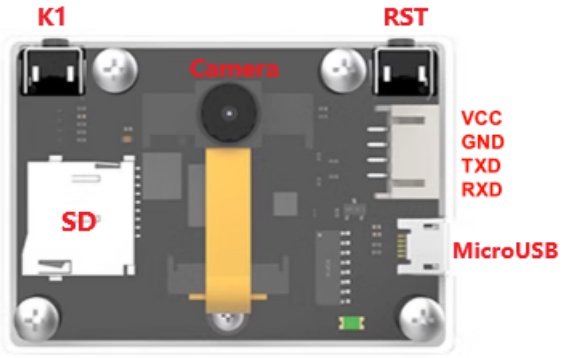
Hardware connection

Peripherals	Development board
K210 visual module: VCC	5V
K210 visual module: TXD	PA2
K210 visual module: RXD	PA3
K210 visual module: GND	GND

Control principle

The K210 visual recognition module identifies the control instructions in the QR code information, and sends the identified QR code information to the development board. The development board controls the balance car to achieve corresponding actions according to the instruction information of the QR code.

- K210 Vision Module




```
The K210 vision module itself is a development board. For detailed usage, please
refer to the module supporting tutorial
```

> **Download the program**

Connect the SD card of the K210 vision module to the computer through a card reader, rename the program file to main.py and copy it to the SD card, then reinstall the SD card back into the SD card slot of the K210 vision module.

```
The K210 vision module and development board case codes are in the same folder:
the folder name will distinguish the development board to which the code belongs
```

QR code

The QR code test can use the pictures in the tutorial, or use the webpage generated by QR Code online to generate a QR code picture.

Communication protocol

The program of the K210 visual module will parse the control instructions of the QR code, and the control instructions need to match the preset program.

QR code information	K210 sent content	Balance car action
goahead	$goahead#	Stop after 2 seconds forward
goback	$goback#	Stop after 2 seconds backward
turnleft	$turnleft#	Stop after 1 second left and 1 second forward

QR code information	K210 sent content	Balance car action
turnright	$turnright#	Stop after 1 second right and 1 second forward

Main code

The tutorial mainly explains the code for the K210 QR code recognition and control of the balance car function. For detailed code, please refer to the corresponding project file.

> **Deal_K210**

Receive valid data sent by K210.

```
void Deal_K210(uint8_t recv_msg)
{
    if (recv_msg == '$' && g_new_flag == 0)
    {
        g_new_flag = 1;
        memset(buf_msg, 0, sizeof(buf_msg)); // Clear old data
        return;
    }
    if(g_new_flag == 1)
    {
        if (recv_msg == '#')
        {
            g_new_flag = 0;
            g_index = 0;
            g_new_data = 1;
        }

        if (g_new_flag == 1 && recv_msg != '$')
        {
            buf_msg[g_index++] = recv_msg;

            if(g_index > 20) // Array overflow
            {
                g_index = 0;
                g_new_flag = 0;
                g_new_data = 0;
                memset(buf_msg, 0, sizeof(buf_msg)); // Clear old data
            }

        }
    }
}
```

> **Change_state**

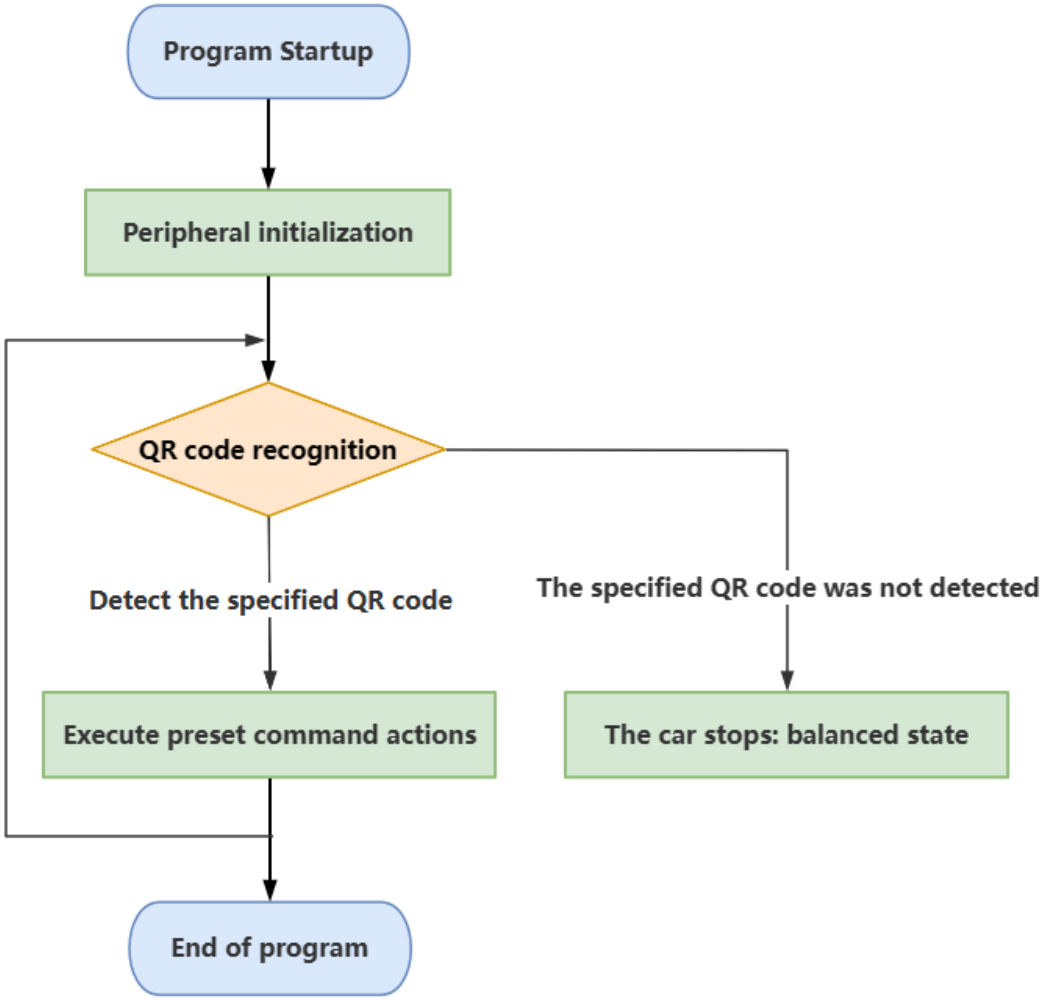
Execute different actions according to different instructions sent by k210.

```
void Change_state(void)
{
    if(g_new_data == 1)
```

```
    {
        g_new_data = 0;
        if (strcmp("goback", buf_msg) == 0 )
        {
            // The car moves back for two seconds and then stops
            Move_X = -Go_speed;
            my_delay(2);
            Move_X = 0;
        }
        else if (strcmp("goahead", buf_msg) == 0 )
        {
            // The car moves back for two seconds and then stops
            Move_X = Go_speed;
            my_delay(2);
            Move_X = 0;
        }
        else if (strcmp("turnleft", buf_msg) == 0)
        {
            // The car turns left for 1 second and then moves forward for 1
second before stopping
            Move_Z = -Trun_speed;
            my_delay(1);

            Move_Z = 0;
            Move_X = Go_speed;

            my_delay(1);
            Move_X = 0;
        }
        else if (strcmp("turnright", buf_msg) == 0 )
        {
            // The car turns right for 1 second and then moves forward for 1
second before stopping
            Move_Z = Trun_speed;
            my_delay(1);

            Move_Z = 0;
            Move_X = Go_speed;

            my_delay(1);
            Move_X = 0;
        }
    }
}
```

Program flow chart

Brief introduction to the function implementation process:



Experimental phenomenon

> **Software code**

The K210_BalancedCar_QR.hex file generated by the project compilation is located in the OBJ folder of the K210_BalancedCar_QR project. Find the K210_BalancedCar_QR.hex file corresponding to the project and use the FlyMcu software to download the program to the development board.

```
Both the K210 visual module and the development board need to download the
program to realize the corresponding function
Product supporting materials source code path: Attachment → Source code summary →
5.Balanced_Car_Extended → 07.K210_BalancedCar_QR
```

> **Experimental phenomenon**

After the program is started, press KEY1 according to the OLED prompt to start the QR code recognition function of the balance car: OLED displays the program function name in real time (Start discern QR!); K210 visual recognition module recognizes the QR code and performs the corresponding action; if the QR code is not recognized, the balance car will stop moving and maintain balance.

The program has voltage detection. If the voltage is less than 9.6V, the low voltage alarm is triggered and the buzzer will sound.
Common situations in which the voltage alarm is triggered:
1. The power switch of the development board is not turned on, and only the Type-C data cable is connected for power supply
2. The battery pack voltage is lower than 9.6V and needs to be charged in time