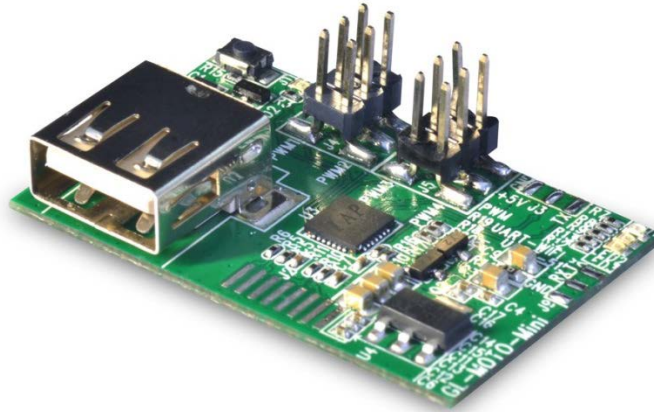# GL-Moto-Mini user manual

1. Introduction

Powered by Domino Core, GL-Moto-Mini is a Wi-Fi controller for R/C hobbies. Using your smartphone, you can control R/C cars, robots or even Helis easily. Plugin a UVC Webcam and you can have live video streaming to your smartphone.

The controller have 4-channel PWM output, USB-A, 4 LEDs, one Reset. It also have UART for both CPU and MCU and Ethernet port reserved for debugging.

You need to install it in your vehicle, as in this picture.
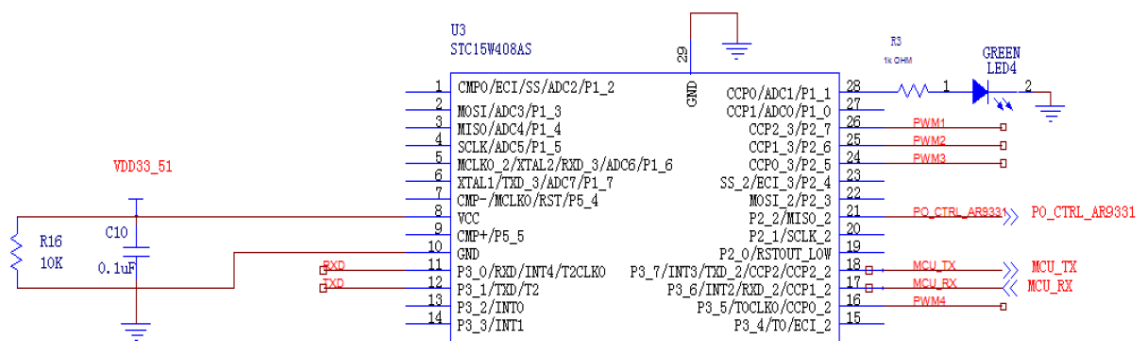
2. Hardware
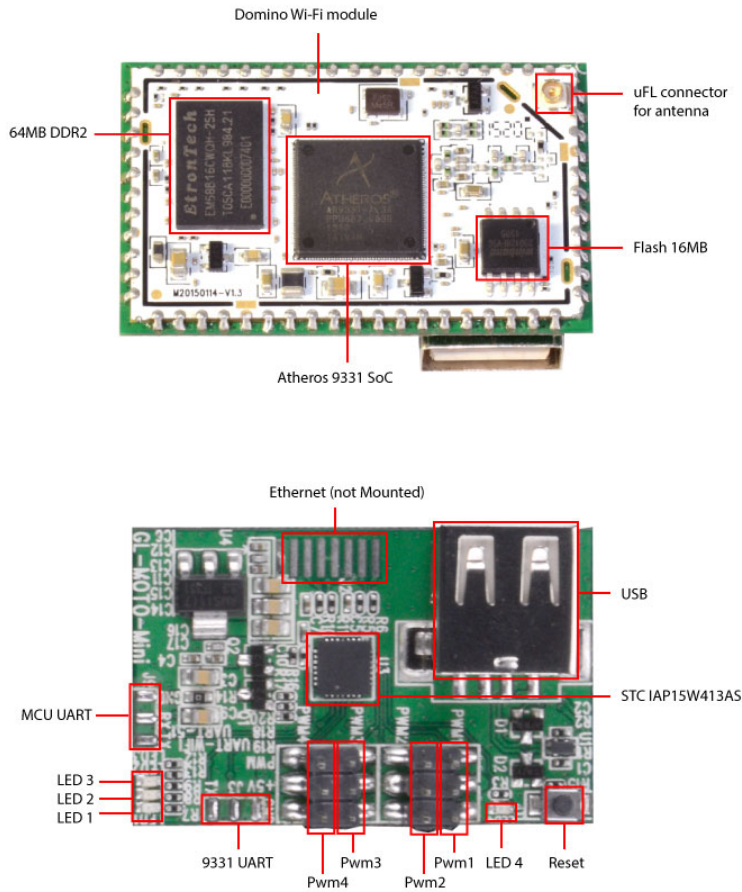
2.1. MCU footprint

The footprint of the MCU is as attached. You can check how the MCU is connected to the interfaces of the board and how it is connected to CPU.
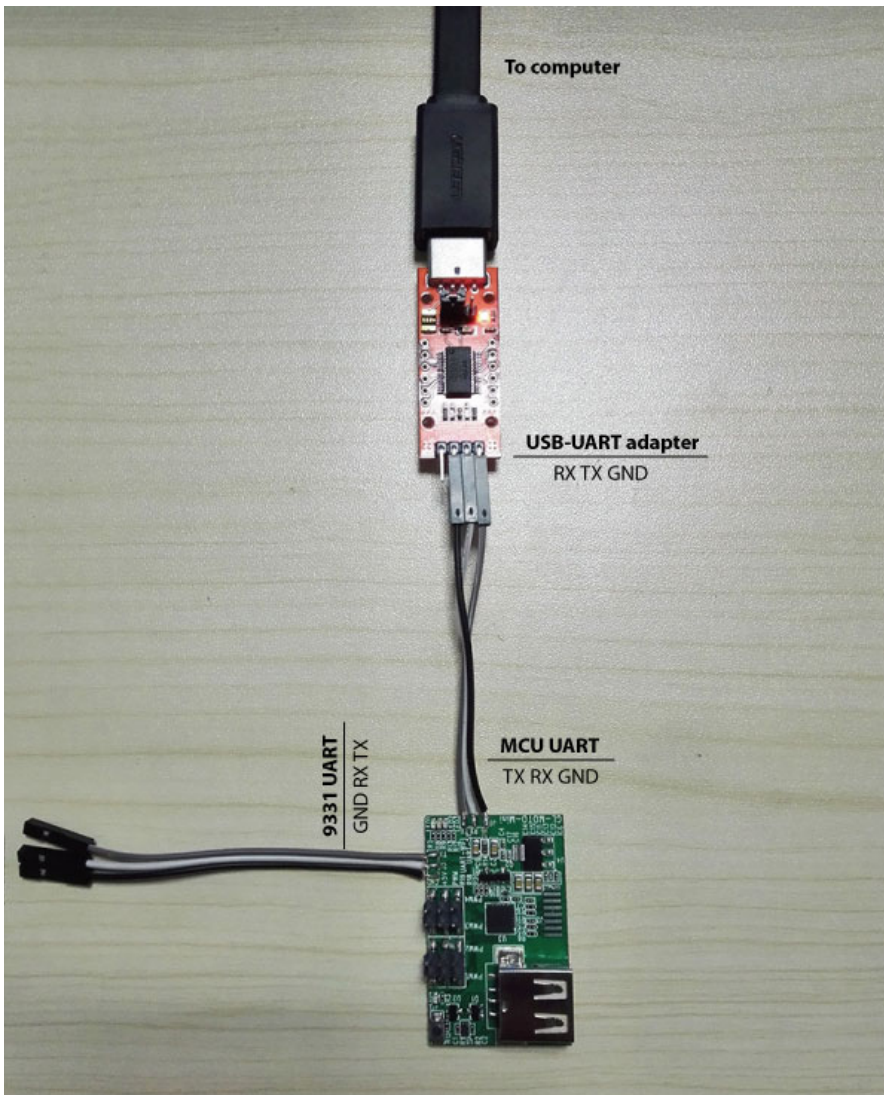

3. MCU Programming

You only need a USB-UART adapter to download the bin to the MCU.

3.1. Connection

You need a USB-UART adapter as this one or similar one.

You need to solder the wire to the MCU UART pins as illustrated below, then connect to your computer via the USB-UART adapter.

3.2. Compile the code

The code can be downloaded from our website. You can use Keil 4.0 to compile it. We are searching for an open source tool for compiling the source code.

3.3. Download program to your MCU

Download the MCU programming tools from our website. Unzip and run stc-isp-15xx-v6.85H.exe

Step1: Execute the program

You may need to change the UI to English if it is in Chinese.
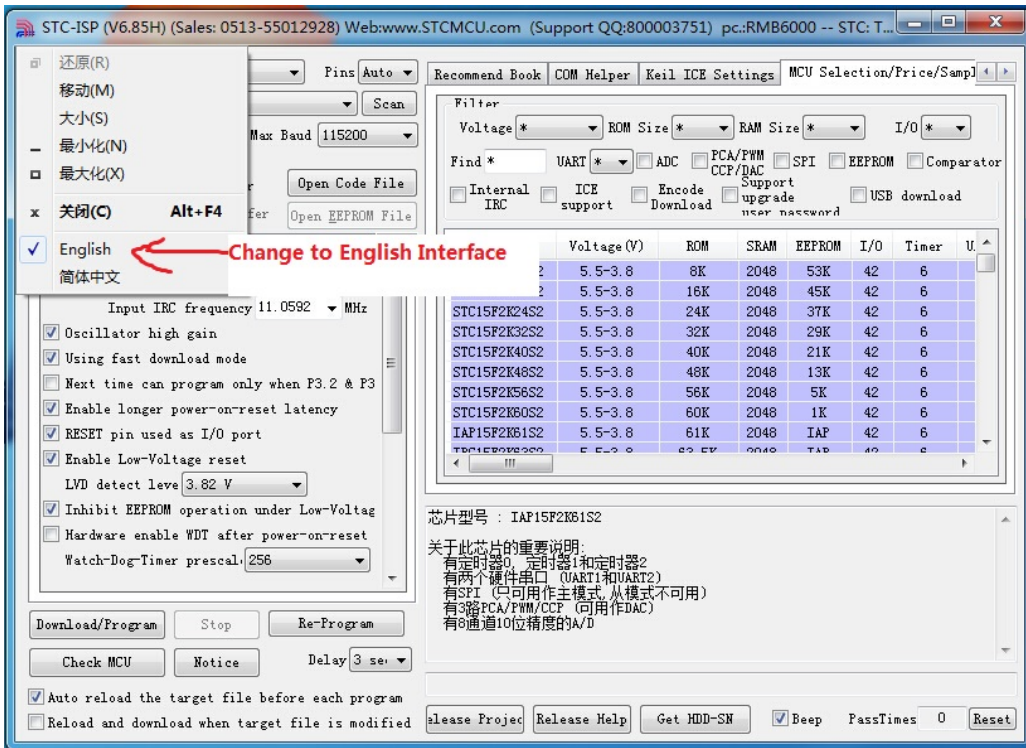
Step 2: Choose the correct MCU – IAP15W413AS

It is in the STC15W408AS series.
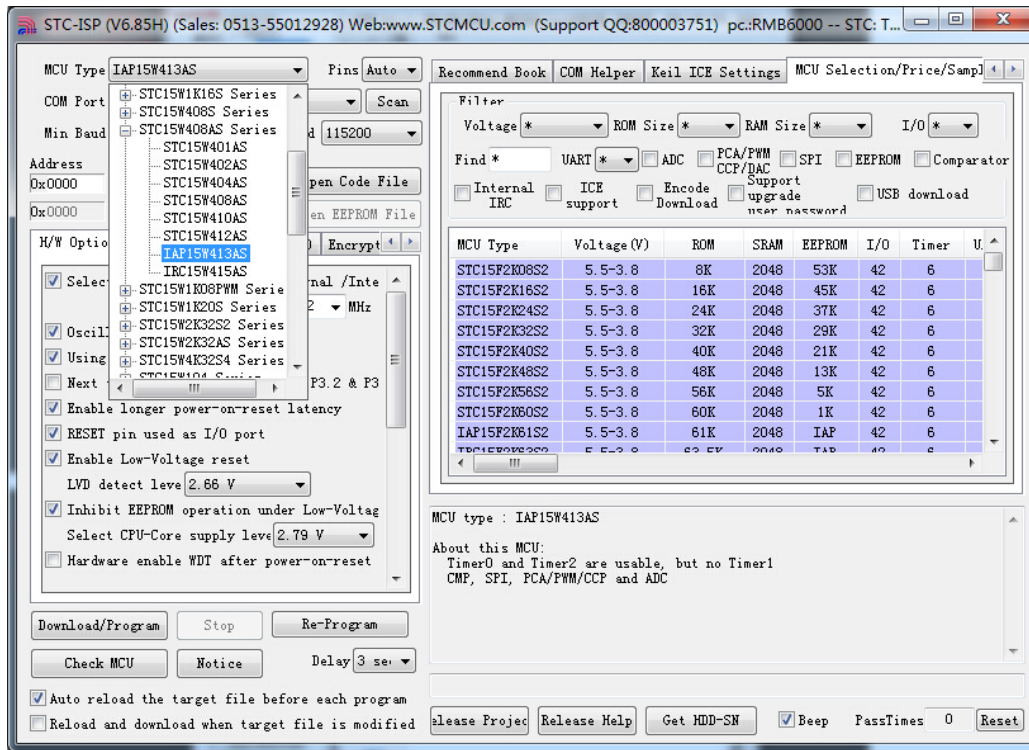
Step 3: Open the bin file

Click the "Open Code File" button and choose your bin file for MCU.

Step 4: Choose MCU crystal frequency – 22.1184Mhz

Step 5: Choose the correct COM port

The software will automatically choose the correct COM port. If not, choose it manually now.
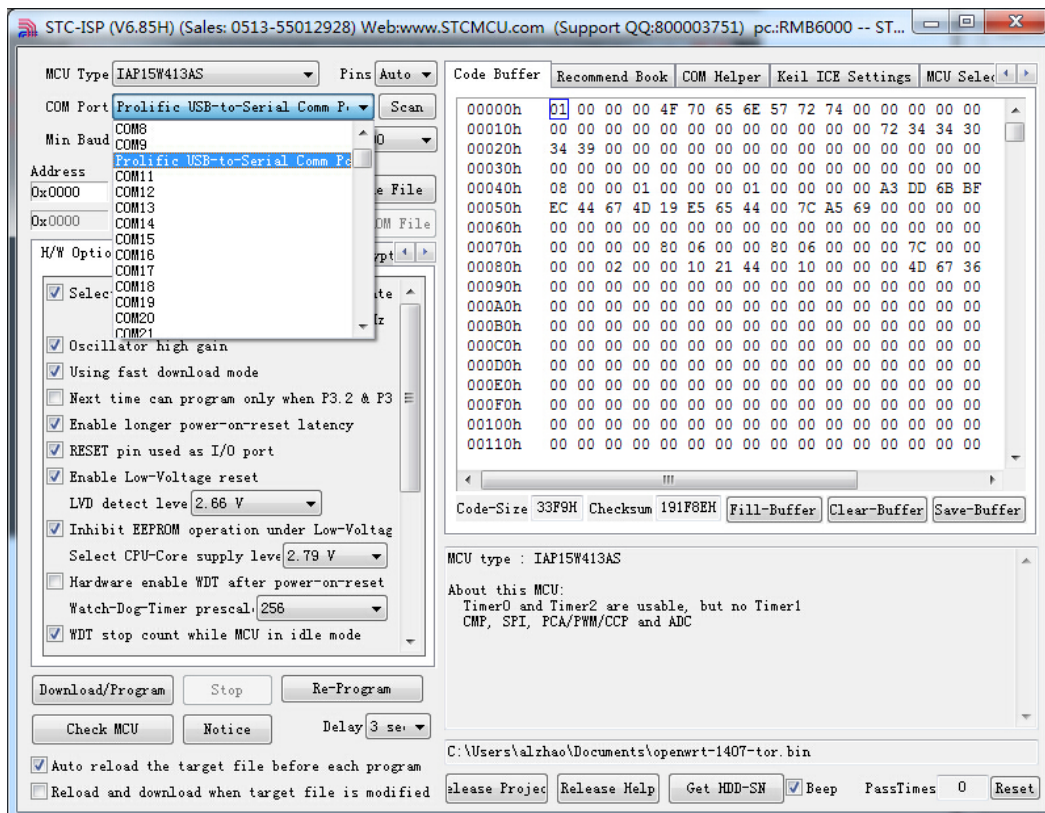
Step 6: Download/Program

Now click the "Download/Program" button and it will be down quickly.
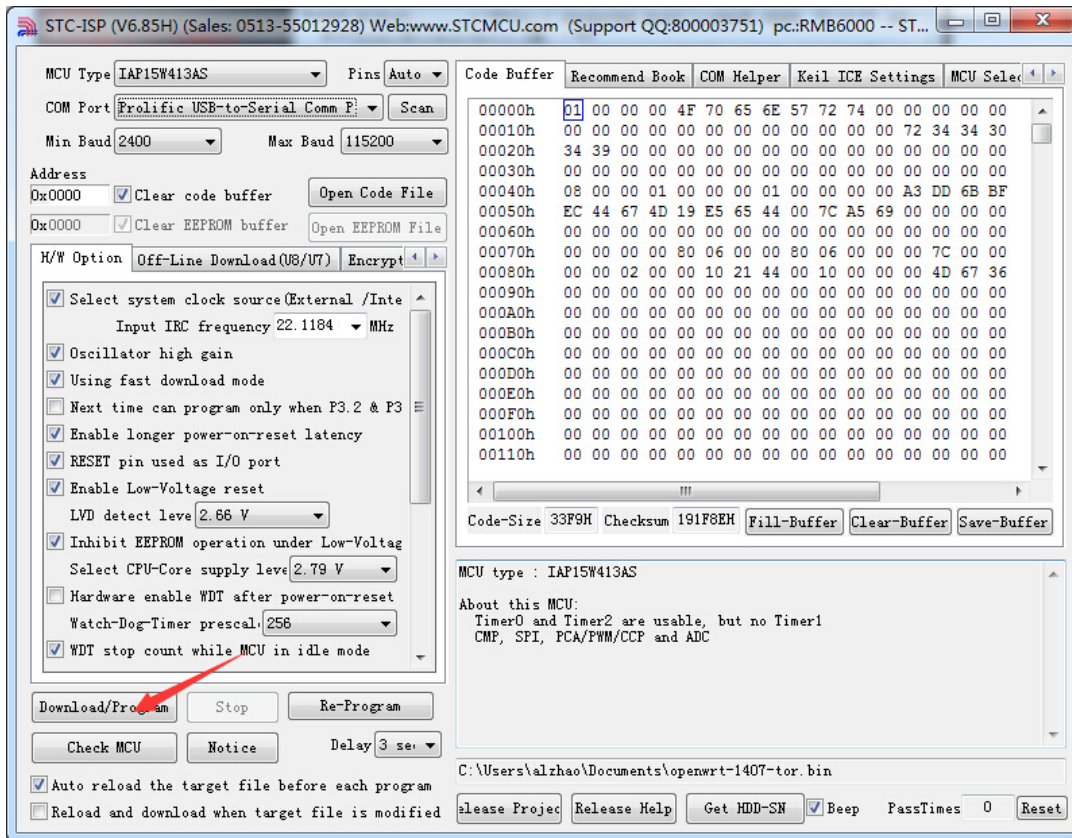
4. MCU code

The main.c is commented in both Chinese and English. Download the source from our website.

The code is quite straightforward. If you have any questions, please as in our forum.
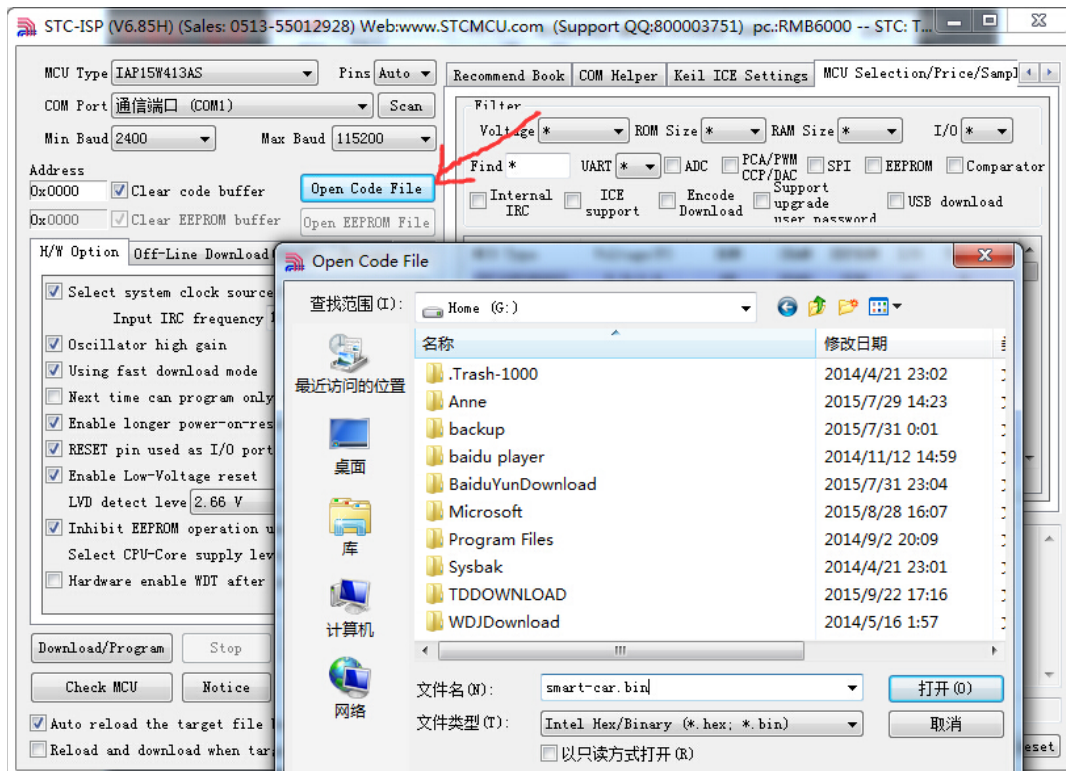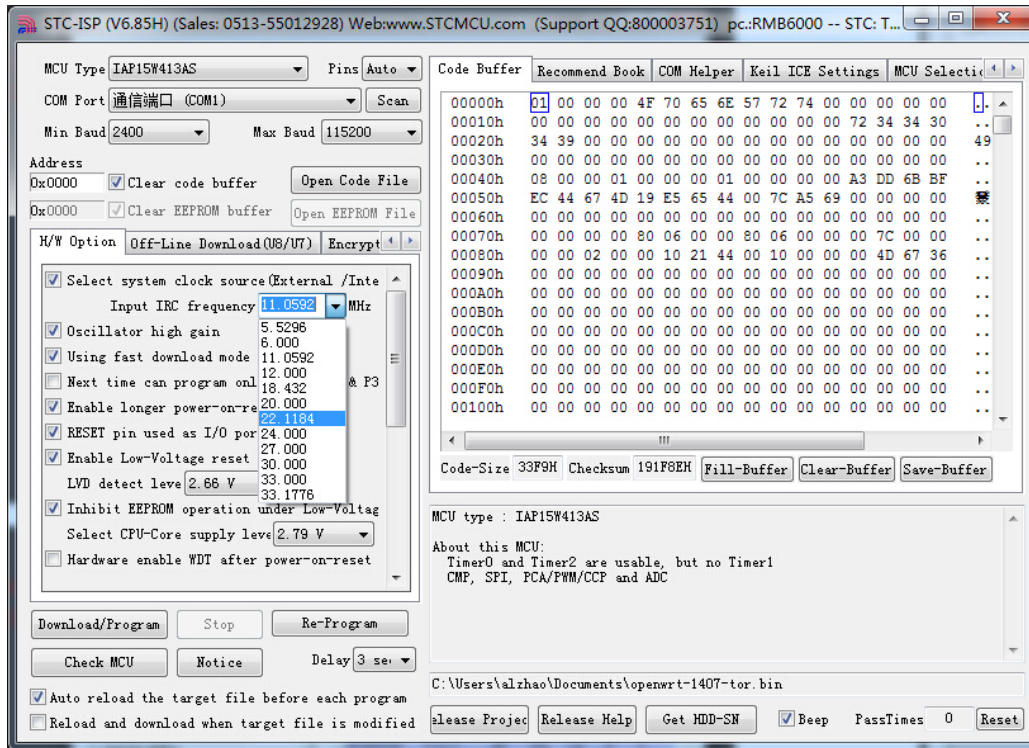
```c
void main(void)
{
        led_init();                          //LED set to output
        led_off();                  //LED turn off

        router_init();          //router set to output
        router_close();             //close router
        router_open();              //open router

        UART_config();              //UART init

        PCA_config();               //PWM init
        pca_pwm_init();             //PWM set to output

        //TIME_PWM test
        P_PWM = 0;
        P3M1 &= ~(1 << 5); //P3.5 set to output
```

```
                P3M0 |= (1 << 5);

                Timer_config();

                pwm = 2768;           //set a init value for PWM, %10 duty cycle
                LoadPWM(pwm);

                EA = 1;                                       //init interrupt

        while (1)
        {
                delay_ms(1);
                if(++communicate_time > 500)  //Increase each loop, when bigger than
500, it will regard as communication interrupt
                {
                        //failsafe
                        message_data[0]=1500;
                        message_data[1]=1500;
                        message_data[2]=1500;
                        message_data[3]=1500;
                        if(communicate_time==60000)//lost control
                        {
                                led_on();//turn on LED
                        }
                        else
                        {
                                led_off();//turn off LED
                        }
                }
                get_message();  //get messages

                /*
                *   PWM calculation:
                *
                *   924+3.68*(1000-1000)            924           0.5
                * ----------------------   = ------------  = ---------
                *          36874                   36874          20
                *
                *   924+3.68*(1500-1000)            2764           1.5
                * ----------------------   = ------------  = ---------
                *          36874                   36874          20
                *
                *   924+3.68*(2000-1000)            4604           2.5
                * ----------------------   = ------------  = ---------
                *          36874                   36874          20
                *
                */
                //message_data[0] is steering,message_data[1] is throttle

                //pwm2 connected to GL-MOTO-Mini's PWM1,the timmer is pwm2, pin is P2.7
                //PWM1 full cycle
                pwm2 = 924+3.68*(message_data[1]-1000);//20ms duty cycle 0.5~2.5
                if(pwm2 >= PWM_HIGH_MAX_PCA) pwm2 = PWM_HIGH_MIN_PCA;
                PWMn_Update(PCA2,pwm2);
```

```
                  //PWM1 half cycle
                  //pwm2 = 1843+1.85*(message_data[1]-1000);//20ms duty cycle 1~2
                  //if(pwm2 >= PWM_HIGH_MAX_PCA)        pwm2 = PWM_HIGH_MIN_PCA;
                  //PWMn_Update(PCA2,pwm2);

                  //pwm1 connected to GL-MOTO-Mini's PWM2,timer is pwm1, pin is P2.6
                  //PWM1 full cycle
                  pwm1 = 924+3.68*(2000-(message_data[0]));        //20ms duty cycle
0.5~2.5
                  if(pwm1 >= PWM_HIGH_MAX_PCA) pwm1 = PWM_HIGH_MIN_PCA;
                  PWMn_Update(PCA1,pwm1);

                  //PWM2 half cycle
                  //pwm1 = 1843+1.85*(message_data[0]-1000);//20ms duty cycle 1~2
                  //if(pwm1 >= PWM_HIGH_MAX_PCA)        pwm1 = PWM_HIGH_MIN_PCA;
                  //PWMn_Update(PCA1,pwm1);

                  //pwm0 is connected to GL-MOTO-Mini's PWM3, timer is pwm0 pin is P2.5
                  //PWM3 full cycle
                  pwm0 = 924+3.68*(message_data[0]-1000);//20ms duty cycle 0.5~2.5
                  if(pwm0 >= PWM_HIGH_MAX_PCA) pwm0 = PWM_HIGH_MIN_PCA;
                  PWMn_Update(PCA0,pwm0);

                  //PWM3 half cycle
                  //pwm0 = 1843+1.85*(message_data[0]-1000);//20ms duty cycle 1~2
                  //if(pwm0 >= PWM_HIGH_MAX_PCA)        pwm0 = PWM_HIGH_MIN_PCA;
                  //PWMn_Update(PCA0,pwm0);

                  pwm = 924+3.68*(message_data[0]-1000);//20ms duty cycle 1~2
                  if(pwm >= PWM_HIGH_MAX_TIME) pwm = PWM_HIGH_MAX_TIME;
                  LoadPWM(pwm);

                  //pwm = 1843+1.85*(message_data[0]-1000);//20ms duty cycle 1~2
                  //if(pwm >= PWM_HIGH_MAX_TIME)        pwm = PWM_HIGH_MAX_TIME;
                  //LoadPWM(pwm);
          }
}
```

The control message is parsed in getmessage(), the message format is

 #,1500,1500,1500,1500#

The message format is used to determine if this is a control message send from CPU. The values (1500) will be real values send by CPU.

5. The OpenWrt firmware

The OpenWrt firmware can be downloaded from our website. OpenWrt uses UART to communicate with the MCU. A daemon program is running to get message from smartphone client and then send to MCU.

6. Android Client

Step 1: Connect to the Wi-Fi controller

First, turn on your Wi-Fi to scan a list of wireless Aps. You will find **GL-iNet-smartcar**, click and connect to it.
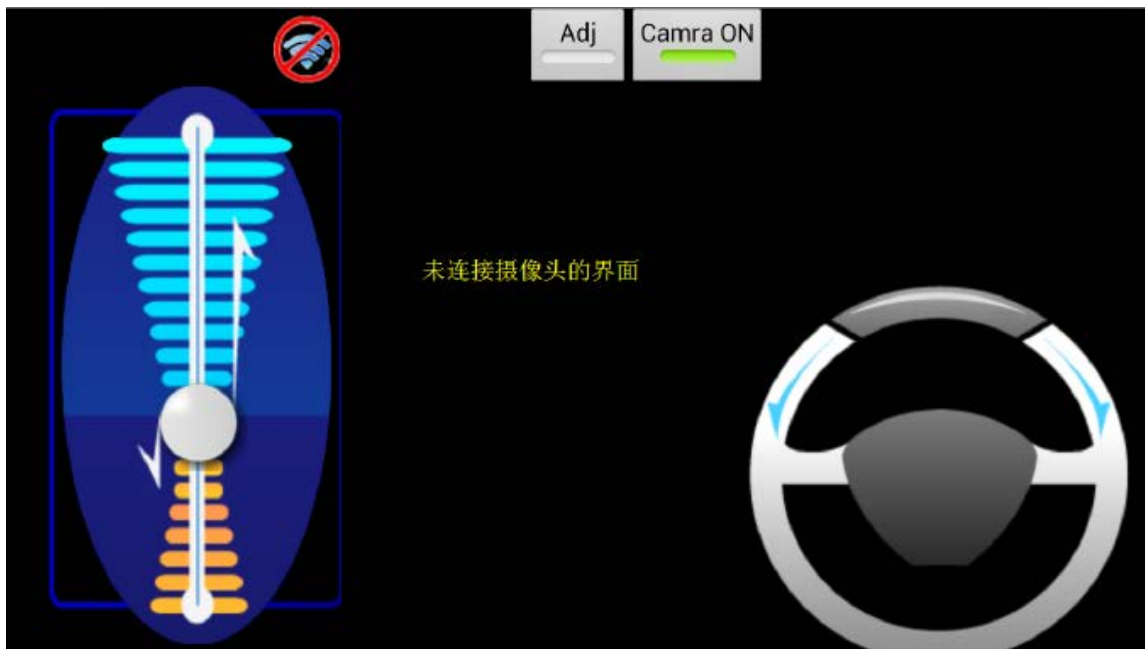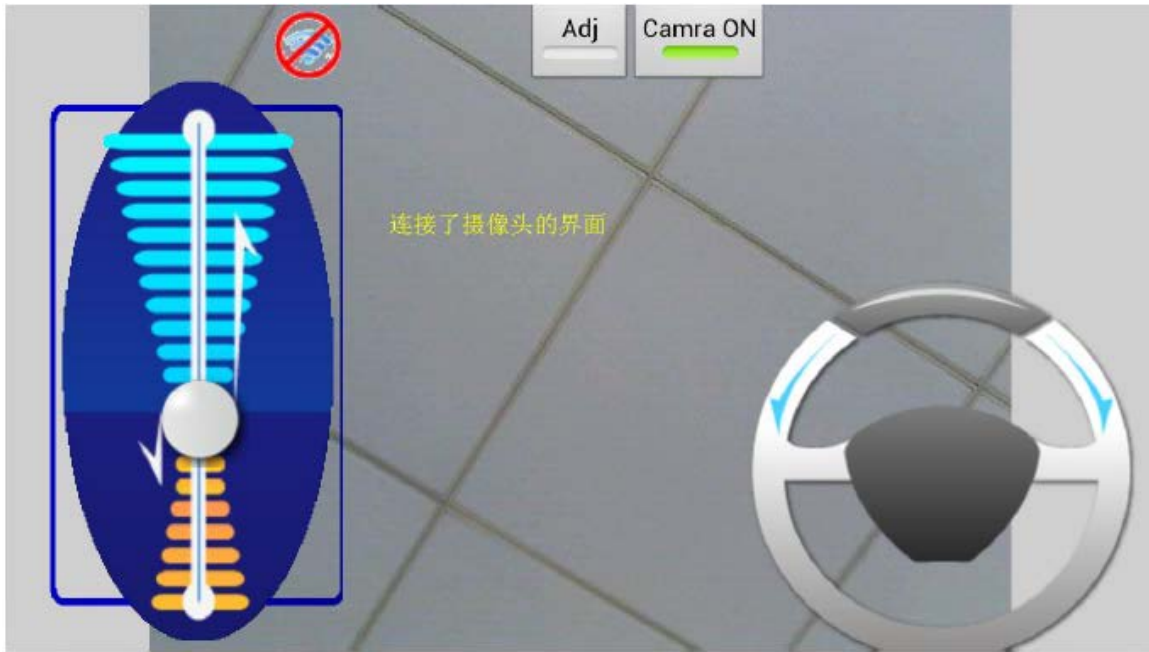
Step 2: Run the APP
Run the app. It will find the controller and automatically connect to it. Click **PLAY** to continue.

Step 3, turn ON/OFF live video streaming
In the interface, you can turn on/off live video streaming by clicking "Camera ON" button. "Adj" button is used for advanced adjustment.

连接了摄像头的界面

7. IOS APP

The IOS app is available in apple store.

(This part is to be finished)