MSPM0_K210 mechanical code recognition

1. K210 communicates with MSPM0

1.1 Experimental prerequisites

This tutorial uses the MSPM0G3507 development board. K210 needs to run the program in **K210-AI(MSPM0G3507)** to start the experiment

MSPM0G *1

K210 visual module *1 (sd card (with AI model), camera)

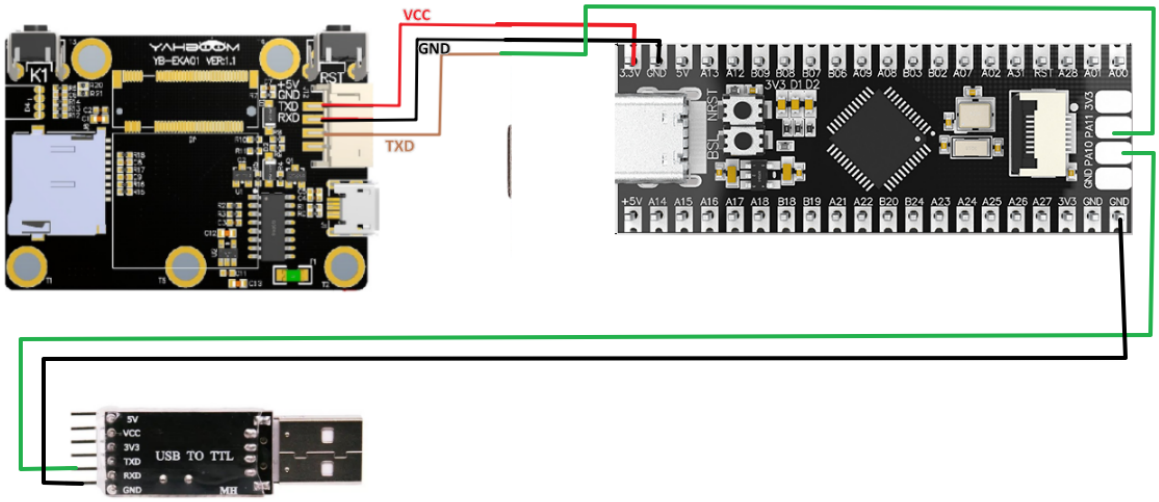
USB to TTL module *1

1.2 Experimental wiring

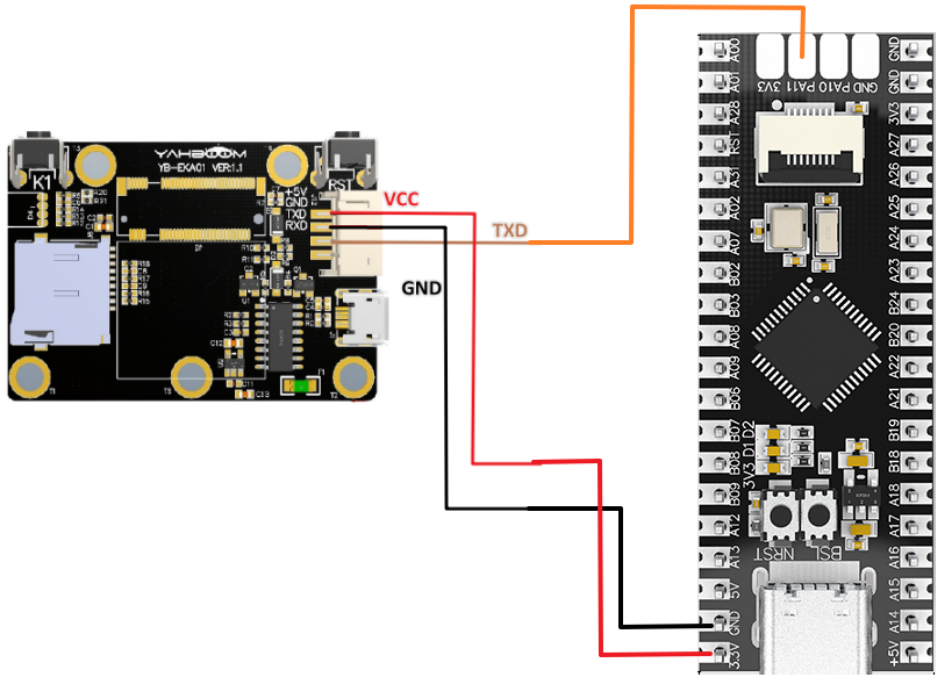
Method 1: Use USB to TTL module

MSPM0G	K210 visual module
RX	TXD
GND	GND
VCC	5V

MSPM0G	USB to TTL module
TX	RXD
GND	GND

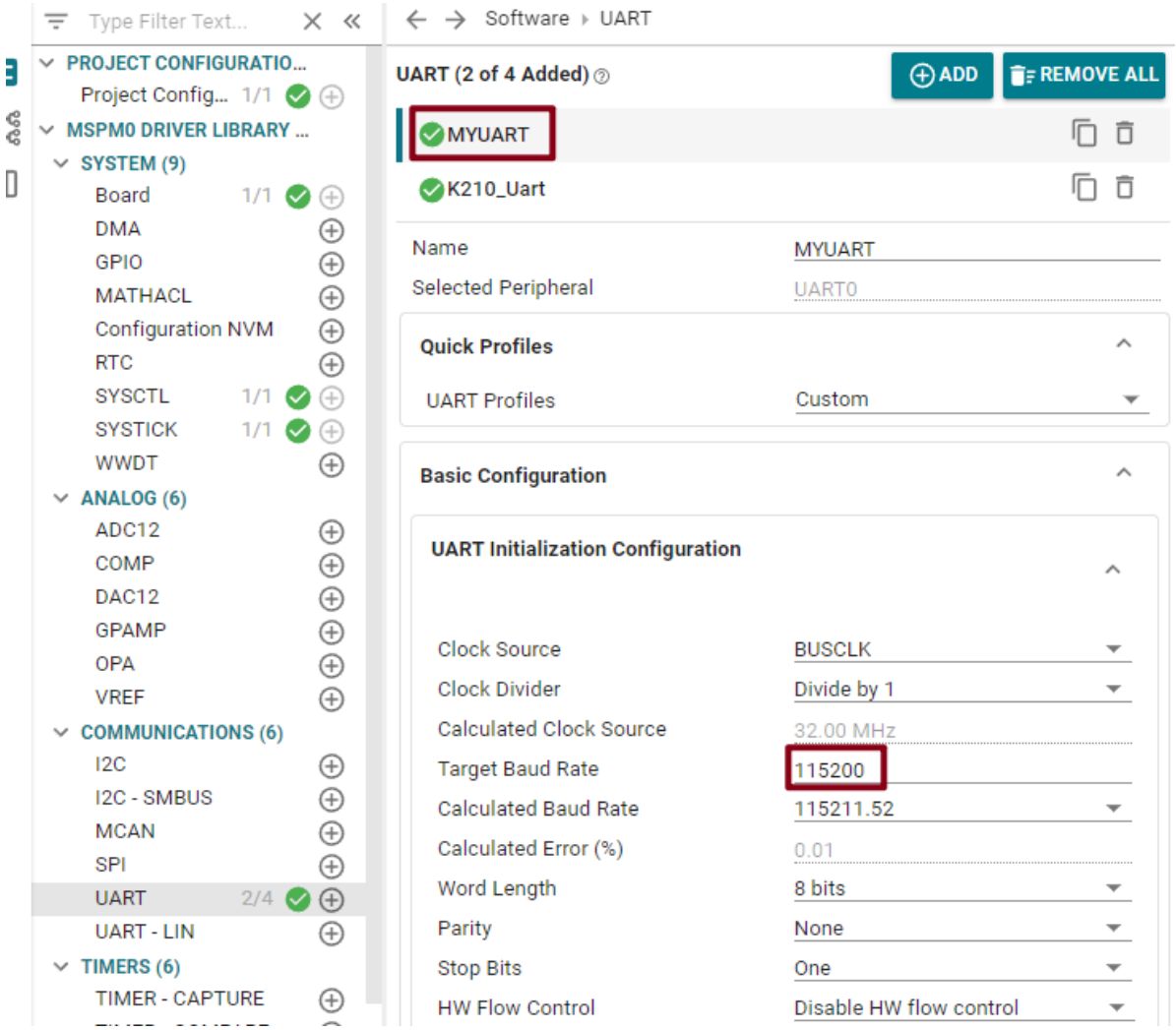


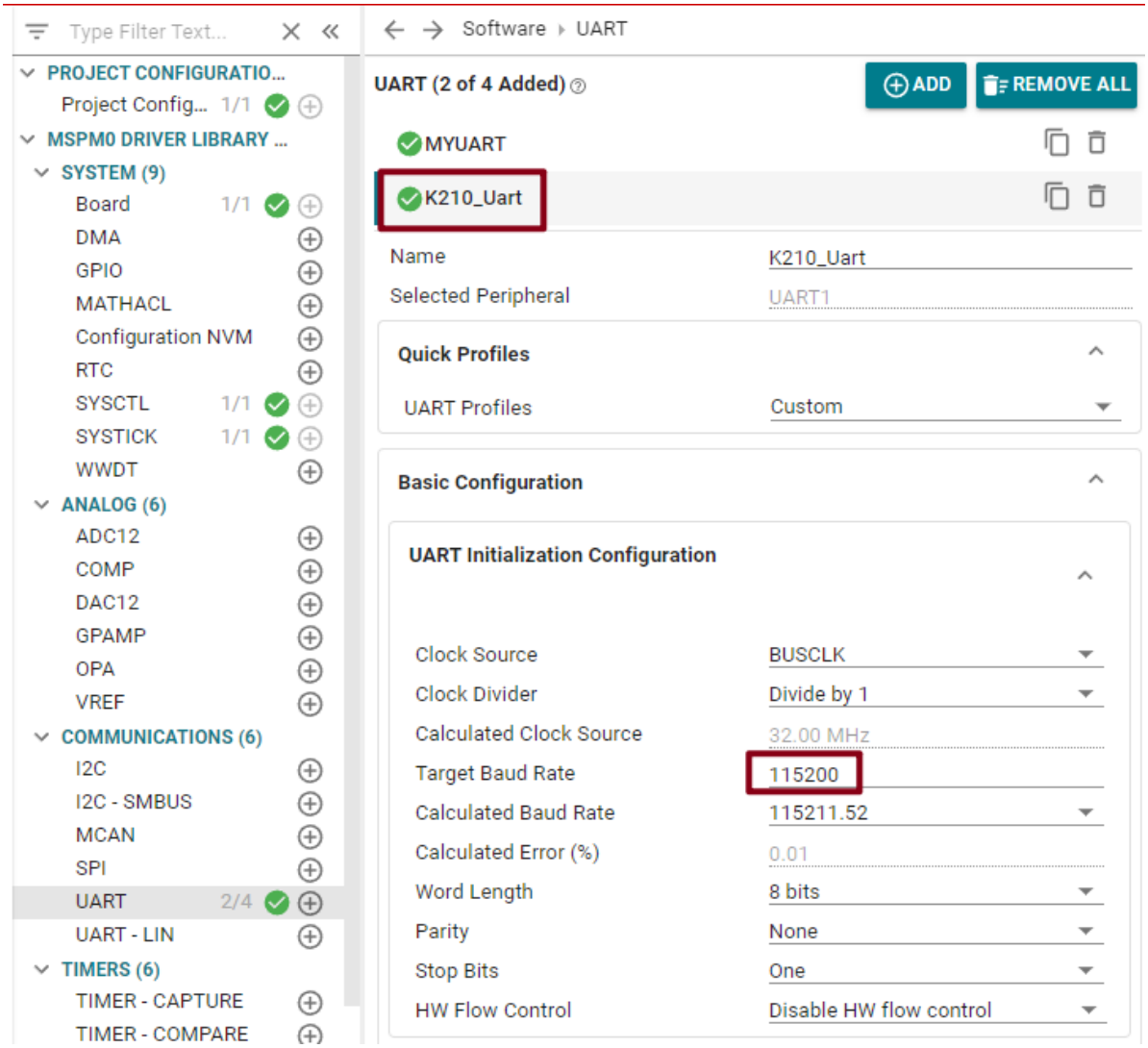
Method 2: Use the onboard Type-C port



1.3 Main code explanation

This example sets the baud rate of the serial port printing to 115200 bps, and the baud rate of the k210 module connection is set to 115200 bps.


Type Filter Text...
PROJECT CONFIGURATIO...
Project Config... 1/1
MSPM0 DRIVER LIBRARY ...
SYSTEM (9)
Board 1/1
DMA
GPIO
MATHACL
Configuration NVM
RTC
SYSCTL 1/1
SYSTICK 1/1
WWDT
ANALOG (6)
ADC12
COMP
DAC12
GPAMP
OPA
VREF
COMMUNICATIONS (6)
I2C
I2C - SMBUS
MCAN
SPI
UART 2/4
UART - LIN
TIMERS (6)
TIMER - CAPTURE
Software ▸ UART
UART (2 of 4 Added)
ADD
REMOVE ALL
MYUART
K210_Uart
Name
MYUART
Selected Peripheral
UART0
Quick Profiles
UART Profiles
Custom
Basic Configuration
UART Initialization Configuration
Clock Source
BUSCLK
Clock Divider
Divide by 1
Calculated Clock Source
32.00 MHz
Target Baud Rate
115200
Calculated Baud Rate
115211.52
Calculated Error (%)
0.01
Word Length
8 bits
Parity
None
Stop Bits
One
HW Flow Control
Disable HW flow control




```
int main(void)
{
    SYSCFG_DL_init();

    NVIC_ClearPendingIRQ(MYUART_INST_INT_IRQN);//清除串口中断标志 Clear the serial port interrupt flag
    NVIC_EnableIRQ(MYUART_INST_INT_IRQN);//使能串口中断 Enable serial port interrupt



    while (1)
    {
         if (k210_msg.class_n != 0)//例程号不为空 Routine number is not empty
        {
            if(k210_msg.class_n == 4)
            {

sprintf(buff_com,"x=%d,y=%d,w=%d,h=%d\r\n",k210_msg.x,k210_msg.y,k210_msg.w,k210_msg.h);
                uart0_send_string(buff_com);

                sprintf(buff_com,"id = %c%c,str = %s\r\n",(k210_msg.id>>8),k210_msg.id,k210_msg.msg_msg);
                uart0_send_string(buff_com);
```

```
                k210_msg.class_n = 0;//清除例程号 Clear Routine Number
            }

        }
        delay_ms(500);

    }
}
```

After the above program, if this routine is run, the members of the k210_msg structure will have corresponding values and will be processed through the serial port printing

k210_msg: is the structure for receiving information, and its main members are

- x: is the horizontal coordinate of the upper left corner of the identified box (range: 0-240)
- y: is the vertical coordinate of the upper left corner of the identified box (range: 0-320)
- w: is the width of the identified box (range: 0-240)
- h: is the length of the identified box (range: 0-320)
- id: is the identified label
- class_n: routine number
- msg_msg[20]: valid data

After the data is received and processed by the function, each member of k210_msg will store valid information. If you want to perform secondary development, you can directly call the members of k210_msg

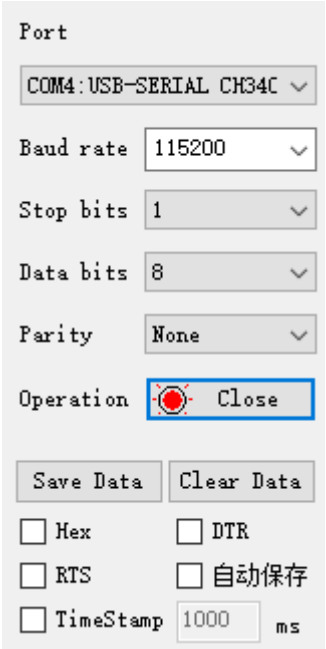
Note: The keil project source code must be compiled under the SDK path.

1.4 Experimental Phenomenon

1. After connecting the cable, the K210 perspective module runs offline. Please check 【6.2 K210 as coprocessor】--【ReadMe】

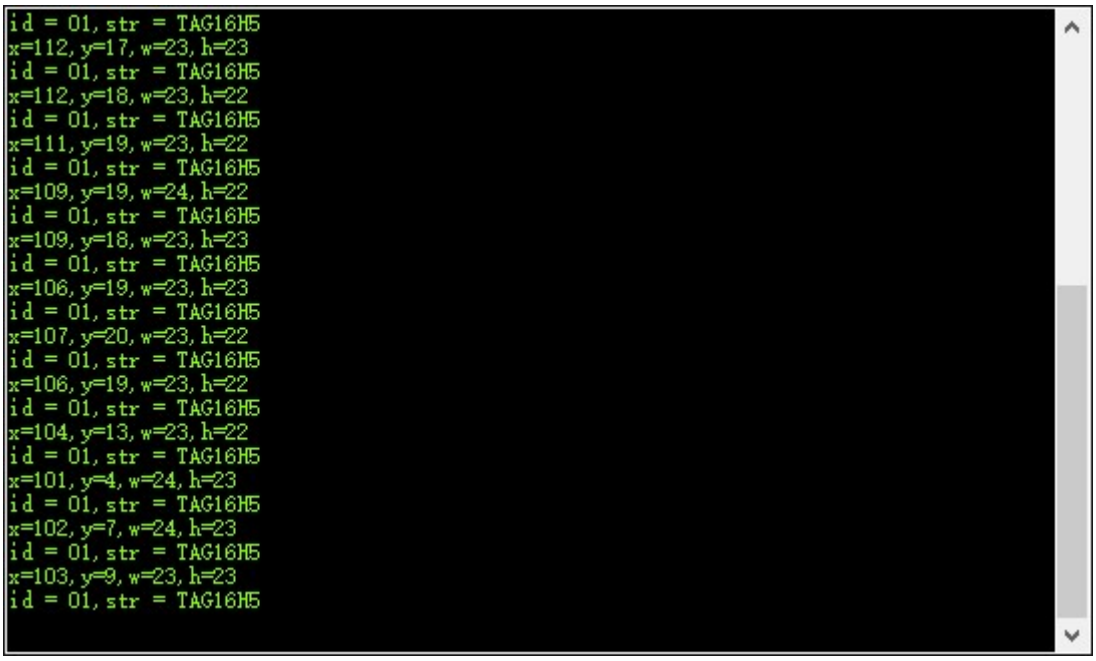
Link:http://www.yahboom.net/study/K210-AI-Camera

2. The serial port assistant is set to the interface as shown in the figure



3. Then run other routines of mechanical code recognition, and the serial port assistant will print out the important information transmitted by k210 to MSPM0G, as shown in the figure

below



Mechanical code recognition only transmits the six member variables x, y, w, h, msg, and id of k210_msg.