

K230 Vision Recognition Module

K230 Vision Recognition Module

- 1. Software and Hardware
- 2. Basic Principles
 - 2.1 Hardware Schematic
 - 2.2 Physical Connection Diagram
 - 2.3 Control Principle
- 3. Project Configuration
 - 3.1 Description
 - 3.2 Pin Configuration
- 4. Main Functions
 - 4.1 User Functions
- 5. Experimental Phenomenon

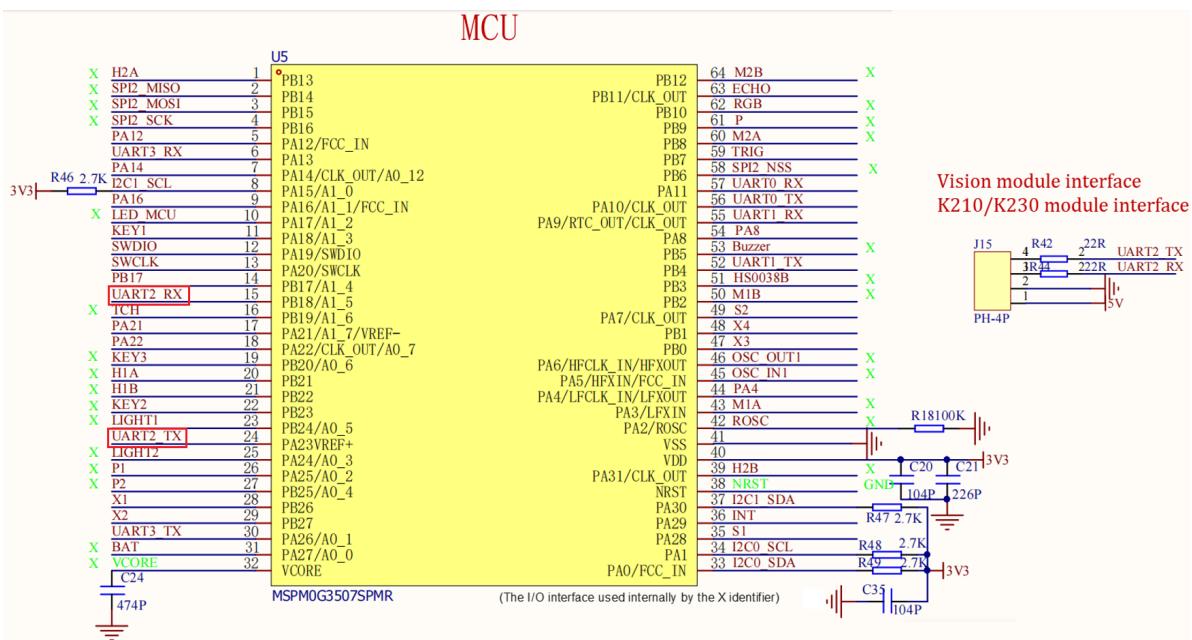
This tutorial demonstrates: how the MSPM0G3507 expansion board works with the K230 vision recognition module, using the function buttons on the expansion board to control the K230 module.

1. Software and Hardware

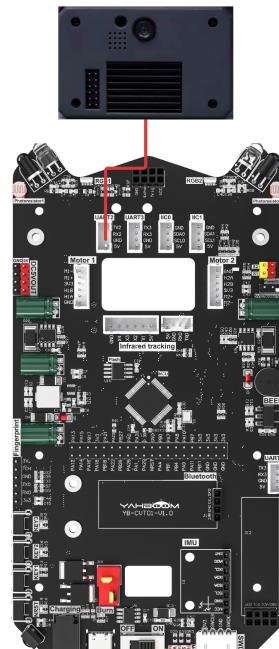
- **KEIL5**
- **MSPM0G3507 Development Board**
 - Function buttons integrated on the development board
- **K230 Vision Module**
- **Type-C data cable or ST-Link**
 - For programming download or simulation to the development board
- **Serial Port Assistant**
 - Receive and print serial port data

2. Basic Principles

2.1 Hardware Schematic



2.2 Physical Connection Diagram



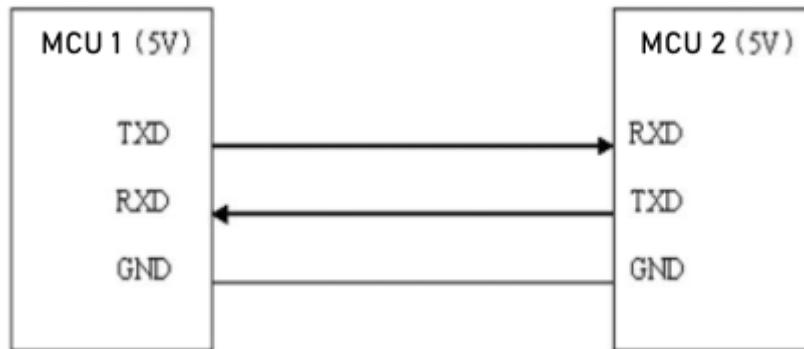
2.3 Control Principle

K230 Wiring (Note: The wiring diagram below is for position reference only. Our factory provides K230 dual-head PH2.0 4Pin all-black cables with anti-mistake design, so you don't need to worry about wiring issues)

K230	MSPM0G3507
RX	TX2
TX	RX2
GND	GND
5V	5V

Serial Communication:

USART: Universal Synchronous/Asynchronous Receiver/Transmitter. It is a hardware peripheral integrated inside MSPM0G3507 that can **automatically generate data frame timing based on one byte of data from the data register and send it out through the TX pin, or automatically receive data frame timing from the RX pin and concatenate it into one byte of data, stored in the data register.**

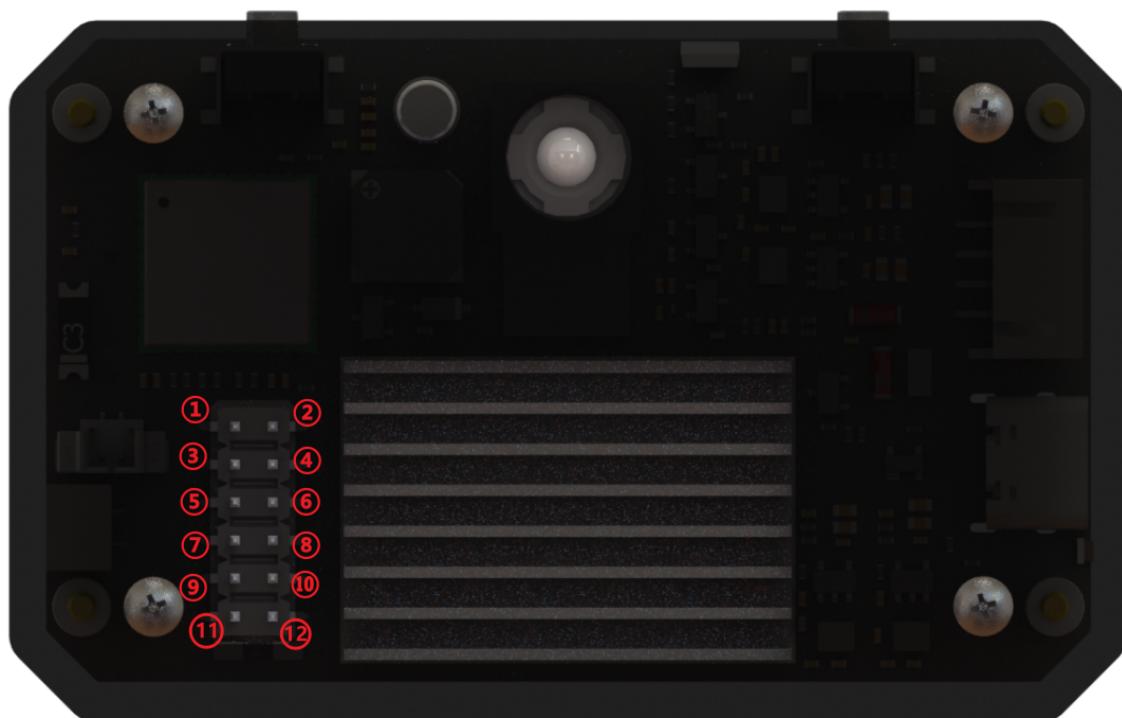


When using serial port for communication, both communication parties must be on the "same channel". "Same channel" means the same communication protocol. Serial port (USART) stipulates: data must be transmitted in the form of "frames" during communication. One frame of serial port data includes: start bit + data bit + parity bit + stop bit. Where: 1) Start bit: fixed is 1 cycle low level signal 2) Data bit: can be agreed by both parties as 5 ~ 9 bits 3) Parity bit: serial port uses parity check, can be agreed by both parties 4) Stop bit: optional 0.5 ~ 2 cycles high level. At the same time, to synchronize the transmission speed of both parties, it is also necessary to agree on the number of data frames transmitted per second, called **baud rate**. Typical baud rates include 9600, 115200, 57600...

K230 vision module function distribution is as follows:

To compress the module size, our K230 module does not bring out all pins supported by the chip. Let's look at some commonly used pins of YAHBOOM K230

1. Two rows of pins next to the heat sink



- 1 : GPIO 42, can be multiplexed as: GPIO42 / UART1_RTS / PWM0 / QSPI1_D2 / RESV /
- 2 : GPIO 43, can be multiplexed as: GPIO43 / UART1_CTS / PWM1 / QSPI1_D3 / RESV /
- 3 : GPIO 33, can be multiplexed as: GPIO33 / IIC0_SDA / IIS_WS / UART3_RXD / RESV /

4 : GND

5 : GPIO 32, can be multiplexed as: GPIO32 / IIC0_SCL / IIS_CLK / UART3_TXD / RESV /

6 : GPIO 26, can be multiplexed as: GPIO26 / MMC1_CLK / RESV / PDM_CLK /

7 : GND

8 : GPIO 34, can be multiplexed as: GPIO34 / IIC1_SCL / IIS_D_IN0 / PDM_IN3 / UART3_RTS /

9 : 5v output

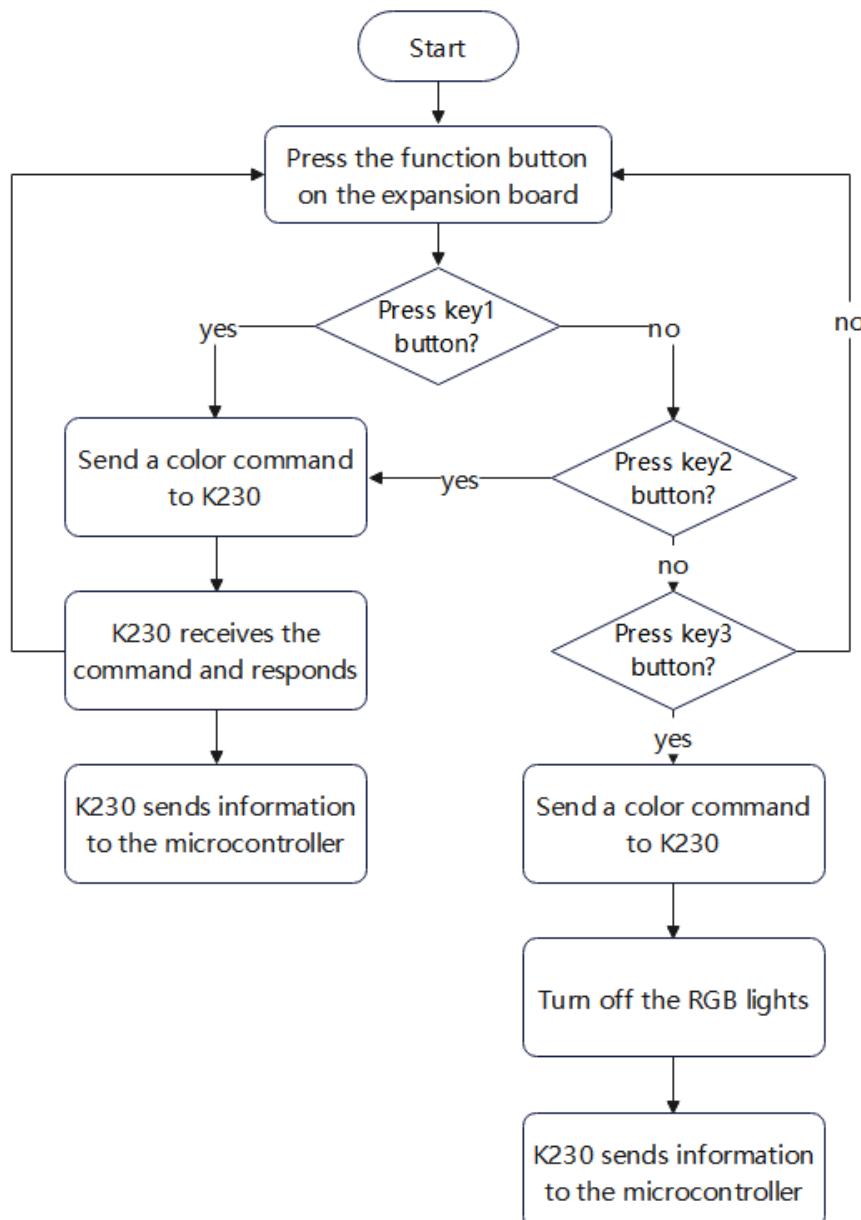
10 : GPIO 35, can be multiplexed as: GPIO35 / IIC1_SDA / IIS_D_OUT0 / PDM_IN1 / UART3_CTS /

11 : 5v output

12 : 3.3v output

!!! When connecting 12pin GPIO, please be sure to confirm the silkscreen markings to ensure correct connection. Artificial damage caused by short circuit, reverse connection, overvoltage or overcurrent will not be able to provide after-sales repair service. !!!!

Operation flow chart is as follows:



3. Project Configuration

3.1 Description

You can refer to the basic tutorial to complete the development environment setup.

3.2 Pin Configuration

The screenshot shows the MSP430 LaunchPad Software interface with the 'GPIO' tab selected. On the left, a tree view lists various peripheral components and their sub-components. The 'GPIO' node under 'SYSTEM' is expanded, showing pins 5, 18, and 27. The pin 5 node is selected. On the right, the pin configuration details are displayed:

- Pin Details:** Name: KEY, Port: Any, Port Segment: Any.
- Group Pins:** Three pins are grouped under 'button1', 'button2', and 'button3'. Each group has its own Name (button1, button2, button3), Direction (Input), and IO Structure (Any).
- Digital IOMUX Features:** Assigned Port: Any, Assigned Port Segment: Any, Assigned Pin: 18.
- Interrupts/Events:** LaunchPad-Specific Pin: No Shortcut Used.
- PinMux:** Peripheral and Pin Configuration.

Type Filter Text... X

Software > GPIO

RGB
SPI

Name: KEY
Port: Any
Port Segment: Any

Group Pins

3 added

button1
 button2
 button3

Digital IOMUX Features

Assigned Port: Any
Assigned Port Segment: Any
Assigned Pin: 23

Interrupts/Events

LaunchPad-Specific Pin: No Shortcut Used

PinMux Peripheral and Pin Configuration

Other Dependencies

PROJECT CONFIGURATION...
Project Config... 1/1
MSP400 DRIVER LIBRARY...
SYSTEM (9)
Board 1/1
Configuration NVM
DMA
GPIO 5
MATHACL
RTC
SYSCTL 1/1
SYSTICK 1/1
WWDT
ANALOG (6)
ADC12 1/2
COMP
DAC12
GPAMP
OPA
VREF
COMMUNICATIONS (6)
I2C
I2C - SMBUS
MCAN
SPI 1/2
UART 3/4
UART - LIN
TIMERS (6)
TIMER 2/7
TIMER - CAPTURE
TIMER - COMPARE
TIMER - PWM 1/7
TIMER - QEI
Timer Fault

Type Filter Text... X

PROJECT CONFIGURATION... Project Config... 1/1 ✓ +

MSP40M DRIVER LIBRARY... SYSTEM (9)

- Board 1/1 ✓ +
- Configuration NVM +
- DMA +
- GPIO 5** ✓ +
- MATHACL +
- RTC +
- SYSCTL 1/1 ✓ +
- SYSTICK 1/1 ✓ +
- WWDT +

ANALOG (6)

- ADC12 1/2 ✓ +
- COMP +
- DAC12 +
- GPAMP +
- OPA +
- VREF +

COMMUNICATIONS (6)

- I2C +
- I2C - SMBUS +
- MCAN +
- SPI 1/2 ✓ +
- UART 3/4 ✓ +
- UART - LIN +

TIMERS (6)

- TIMER 2/7 ✓ +
- TIMER - CAPTURE +
- TIMER - COMPARE +
- TIMER - PWM 1/7 ✓ +
- TIMER - QEI +
- Timer Fault +

RGB ✓

SPI ✓

Name KEY
Port Any
Port Segment Any

Group Pins

3 added

+ ADD REMOVE ALL

- ✓ button1
- ✓ button2
- ✓ button3

Name button3
Direction Input
IO Structure Any

Digital IOMUX Features

Assigned Port Any
Assigned Port Segment Any
Assigned Pin 20

Interrupts/Events

LaunchPad-Specific Pin No Shortcut Used

PinMux Peripheral and Pin Configuration

Other Dependencies

Type Filter Text... X

Software > GPIO

PROJECT CONFIGURATION... Project Config... 1/1 ✓ +

MSP400 DRIVER LIBRARY ... SYSTEM (9)

- Board 1/1 ✓ +
- Configuration NVM +
- DMA +
- GPIO 5 ✓ +
- MATHACL +
- RTC +
- SYSCTL 1/1 ✓ +
- SYSTICK 1/1 ✓ +
- WWDT +

ANALOG (6)

- ADC12 1/2 ✓ +
- COMP +
- DAC12 +
- GPAMP +
- OPA +
- VREF +

COMMUNICATIONS (6)

- I2C +
- I2C - SMBUS +
- MCAN +
- SPI 1/2 ✓ +
- UART 3/4 ✓ +
- UART - LIN +

TIMERS (6)

- TIMER 2/7 ✓ +
- TIMER - CAPTURE +
- TIMER - COMPARE +
- TIMER - PWM 1/7 ✓ +
- TIMER - QEI +
- Timer Fault +

KEY

RGB

SPI

Name RGB

Port Any

Port Segment Any

Group Pins

1 added

+ ADD REMOVE ALL

WQ2812

Name WQ2812

Direction Output

Initial Value Cleared

IO Structure Standard

Digital IOMUX Features

Assigned Port PORTB

Assigned Port Segment Any

Assigned Pin 10

Interrupts/Events

LaunchPad-Specific Pin No Shortcut Used

PinMux Peripheral and Pin Configuration

Other Dependencies

Type Filter Text... X

PROJECT CONFIGURATION... Project Config... 1/1 ✓ +

MSPM0 DRIVER LIBRARY ...

- SYSTEM (9)
 - Board 1/1 ✓ +
 - Configuration NVM +
 - DMA +
 - GPIO 5 ✓ +
 - MATHACL +
 - RTC +
 - SYSCTL 1/1 ✓ +
 - SYSTICK 1/1 ✓ +
 - WWDT +
- ANALOG (6)
 - ADC12 1/2 ✓ +
 - COMP +
 - DAC12 +
 - GPAMP +
 - OPA +
 - VREF +
- COMMUNICATIONS (6)
 - I2C +
 - I2C - SMBUS +
 - MCAN +
 - SPI 1/2 ✓ +
 - UART 3/4 ✓ +
 - UART - LIN +
- TIMERS (6)
 - TIMER 2/7 ✓ +
 - TIMER - CAPTURE +
 - TIMER - COMPARE +
 - TIMER - PWM 1/7 ✓ +
 - TIMER - QEI +
 - Timer Fault +

Software > GPIO

KEY

RGB

SPI

Name SPI
Port Any
Port Segment Any

Group Pins

1 added + ADD REMOVE ALL

CS

Name CS
Direction Output
Initial Value Set
IO Structure Any

Digital IOMUX Features

Assigned Port Any
Assigned Port Segment Any
Assigned Pin 6

Interrupts/Events

LaunchPad-Specific Pin No Shortcut Used

PinMux Peripheral and Pin Configuration

Other Dependencies

Type Filter Text... X

Software > SPI

SPI (1 of 2 Added) ⓘ

+ ADD REMOVE ALL

SPI_W25Q64

Peripheral does not retain register contents in STOP or STANDBY modes. User should take care to save and restore register configuration in application. See Retention Configuration section for more details.

Name: SPI_W25Q64
Selected Peripheral: SPI1

Quick Profiles: SPI Profiles: Custom

Basic Configuration

SPI Initialization Configuration

Mode Select: Controller

Clock Configuration

Target Bit Rate (Hz): 10000000
Calculated Bit Rate: 10000000.00
Calculated Error (%): 0
Frame Format: Motorola 3-wire
Clock Polarity: Low
Phase: Data captured on first clock ...
Frame Size (bits): 8
Bit Order: MSB

PROJECT CONFIGURATION... Project Config... 1/1 ✓ +

MSPM0 DRIVER LIBRARY... SYSTEM (9)

- Board 1/1 ✓ +
- Configuration NVM +
- DMA +
- GPIO 5 ✓ +
- MATHACL +
- RTC +
- SYSCTL 1/1 ✓ +
- SYSTICK 1/1 ✓ +
- WWDT +

ANALOG (6)

- ADC12 1/2 ✓ +
- COMP +
- DAC12 +
- GPAMP +
- OPA +
- VREF +

COMMUNICATIONS (6)

- I2C +
- I2C - SMBUS +
- MCAN +
- SPI 1/2 ✓ +
- UART 2/4 ✓ +
- UART - LIN +

TIMERS (6)

- TIMER 2/7 ✓ +
- TIMER - CAPTURE +
- TIMER - COMPARE +
- TIMER - PWM 1/7 ✓ +
- TIMER - QEI +
- Timer Fault +

Type Filter Text... X ← → Software > UART

PROJECT CONFIGURATION... Project Config... 1/1 ✓ +

MSP40M DRIVER LIBRARY ... SYSTEM (9)

- Board 1/1 ✓ +
- Configuration NVM +
- DMA +
- GPIO 5 ✓ +
- MATHACL +
- RTC +
- SYSCTL 1/1 ✓ +
- SYSTICK 1/1 ✓ +
- WWDT +

ANALOG (6)

- ADC12 1/2 ✓ +
- COMP +
- DAC12 +
- GPAMP +
- OPA +
- VREF +

COMMUNICATIONS (6)

- I2C +
- I2C - SMBUS +
- MCAN +
- SPI 1/2 ✓ +
- UART 2/4 ✓ +
- UART - LIN +

TIMERS (6)

- TIMER 2/7 ✓ +
- TIMER - CAPTURE +
- TIMER - COMPARE +
- TIMER - PWM 1/7 ✓ +
- TIMER - QEI +
- Timer Fault +

UART (2 of 4 Added) ✓ ADD REMOVE ALL

✓ UART_0 ✓ UART_2

Name: UART_2
Selected Peripheral: UART2

Quick Profiles

UART Profiles: Custom

Basic Configuration

UART Initialization Configuration

Clock Source	MFCLK
Clock Divider	Divide by 1
Calculated Clock Source	4.00 MHz
Target Baud Rate	115200
Calculated Baud Rate	115107.91
Calculated Error (%)	0.0799
Word Length	8 bits
Parity	None
Stop Bits	One
HW Flow Control	Disable HW flow control

Advanced Configuration

Extend Configuration

The screenshot shows the 'TI LaunchPad Peripheral Configuration' software interface. On the left is a tree view of peripheral components under 'PROJECT CONFIGURATION' and 'MSP430 DRIVER LIBRARY'. The 'TIMERS (6)' section is expanded, and 'TIMER - PWM' is selected, highlighted with a grey background. The main right panel displays the configuration for 'motor_PWM'. At the top, it says 'TIMER - PWM (1 of 7 Added)'. Below that is a note: 'Peripheral does not retain register contents in STOP or STANDBY modes. User should take care to save and restore register configuration in application. See Retention Configuration section for more details.' A 'Name' field is set to 'motor_PWM' and a 'Selected Peripheral' dropdown is set to 'TIMA0'. Under 'Quick Profiles', a 'PWM Profiles' dropdown is set to 'Custom'. The 'Basic Configuration' section includes 'Clock Configuration' where 'Timer Clock Source' is 'BUSCLK', 'Timer Clock Divider' is 'Divided by 2', 'Calculated Timer Clock Source' is '40000000', and 'Timer Clock Prescale' is '1'. It also includes 'Calculated Timer Clock Values' with 'Calculated Clock Frequency (Hz)' at '40000000' and 'Timer Clock Information' showing '610.35 Hz to 20.00 MHz w/ resolution'. The 'PWM Configuration' section shows 'PWM Period Count' as '1000' and 'Calculated PWM Frequency (Hz)' as '40000'. A 'Start Timer' checkbox is checked.

4. Main Functions

4.1 User Functions

Function: Send_k230

Function Prototype	<code>void Send_k230(uint8_t *data_str, uint16_t datasize)</code>
Function Description	Control the state of RGB light on k230
Input Parameter 1	Data
Input Parameter 2	Data length
Output Parameters	None

Function: Deal_k230

Function Prototype	void Deal_k230(uint8_t recv_msg)
Function Description	Retain k230 information
Input Parameters	Information from serial port
Output Parameters	None

Function: BSP_Loop

Function Prototype	void BSP_Loop(void)
Function Description	Use button, press once to send data to k230
Input Parameters	None
Output Parameters	None

Communication Protocol:

K230 part:

Receiving part: When receiving MCU data from serial port, parse it, then RGB light will give corresponding color feedback.

Sending part: k230 will send color data through serial port, format is **"\$" + color name + "#"**.

MCU part:

Sending part: When pressing expansion board button, will send color data to k230 through serial port.

Receiving part: Receive data sent by k230 through serial port, and process each bit of data.

Data frame judgment: If it's "\$", set flag to 1, then initialize string array; then process next bit of data, if it's not "#", store content in data. Until "#" is recognized, set flag to 0, then process array data. When array content comparison matches specific content, RGB light on car expansion board will give corresponding color feedback.

5. Experimental Phenomenon

After downloading the program, we connect the wiring between k230 and expansion board. Then we control the RGB light of k230 through function buttons on the expansion board.

When we press key1 button, k230's RGB light will light up one color. Pressing key1 once will light up one color;

When we press key2 button, k230's RGB light will light up other three colors (different from the three colors when pressing key1), pressing key2 once will light up one color;

When we press key3 button, k230's RGB light will turn off.

For program download, refer to **【3. Development Environment Setup and Usage: 3. Uniflash burning】**

How to download k230 code:

Required tools: K230, SD card, CanMV IDE K230

We find the K230uart_car.py file in the project folder, then drag it to CanMV IDE K230, then select to save the opened script as main.py to CanMV. For detailed steps, please refer to the [K230 Vision Module](#) tutorial's 【Quick Start】 chapter.

