

# Voice Interaction Control

## Voice Interaction Control

- I. Software-Hardware
- II. Basic Principle
  - 1. Hardware Schematic Diagram
  - 2. Physical Connection Diagram
  - 3. Control Principle
- III. Main Functions
  - 1. User Functions
- IV. Voice Module Commands
- V. Experimental Phenomenon

This tutorial is a comprehensive experiment combining multiple peripherals. You can first understand individual peripherals before proceeding with this experiment.

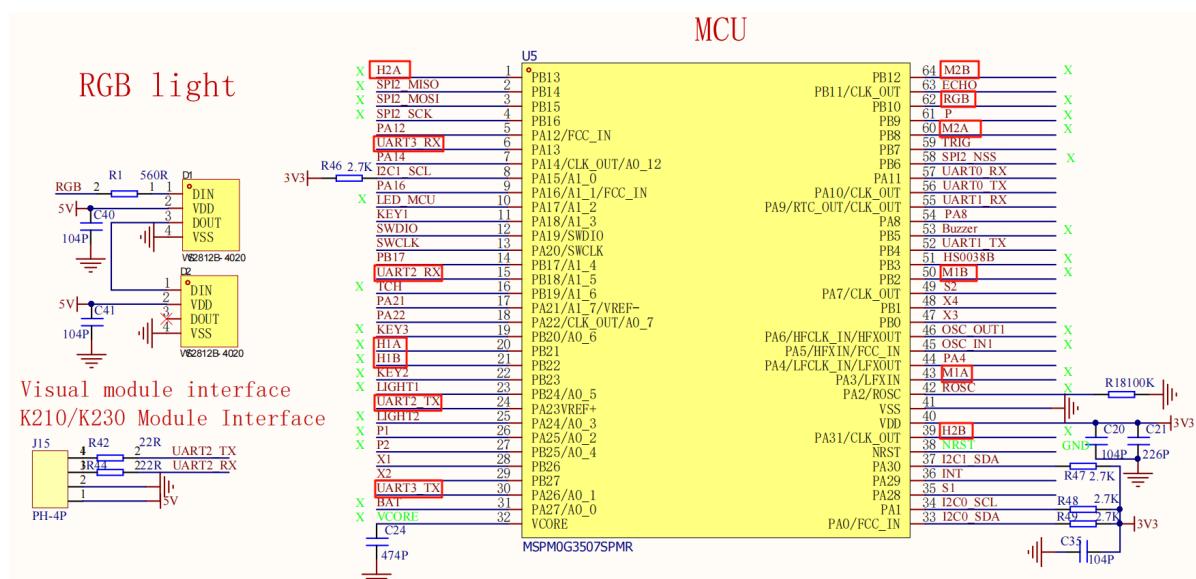
## I. Software-Hardware

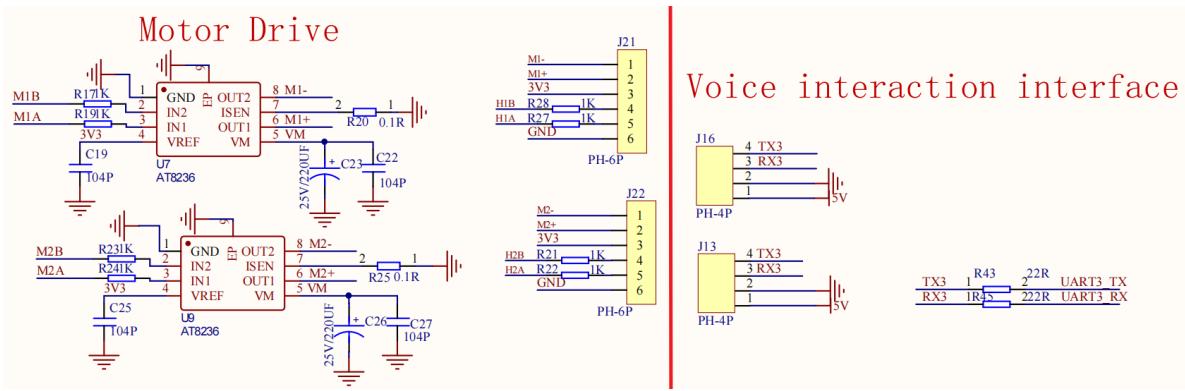
- KEIL
- MSPM0G3507 Development Board
- Type-C Data Cable or DAP-Link
- Voice Interaction Integrated Module
- K230 Vision Module

For program download or simulation to the development board

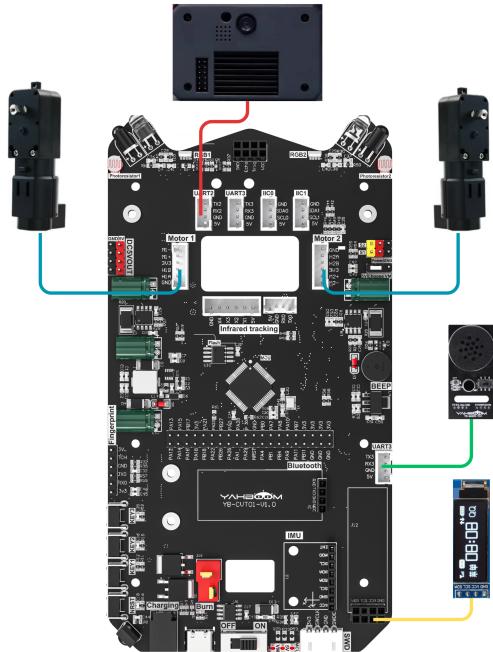
## II. Basic Principle

### 1. Hardware Schematic Diagram





## 2. Physical Connection Diagram



### Wiring Pins

**Motor Wiring** (Note: The wiring diagram below is for position reference only. We provide dual-head PH2.0 6Pin all-black cables with fool-proof design, so you don't need to worry about wiring issues)

TT Encoder Motor	MSPM0G3507
M-	M1-
M+	M1+
VCC	3V3
Encoder B	H1B
Encoder A	H1A
GND	GND

TT Encoder Motor	MSPM0G3507
GND	GND
Encoder A	H2A
Encoder B	H2B
VCC	3V3
M+	M2+
M-	M2-

**Voice Module Wiring** (Note: The wiring diagram below is for position reference only. We provide dual-head PH2.0 4Pin all-black cables for the voice module with fool-proof design, so you don't need to worry about wiring issues)

Smart Voice Interaction Module	MSPM0G3507
RX1	TX3
TX1	RX3
GND	GND
5V	5V

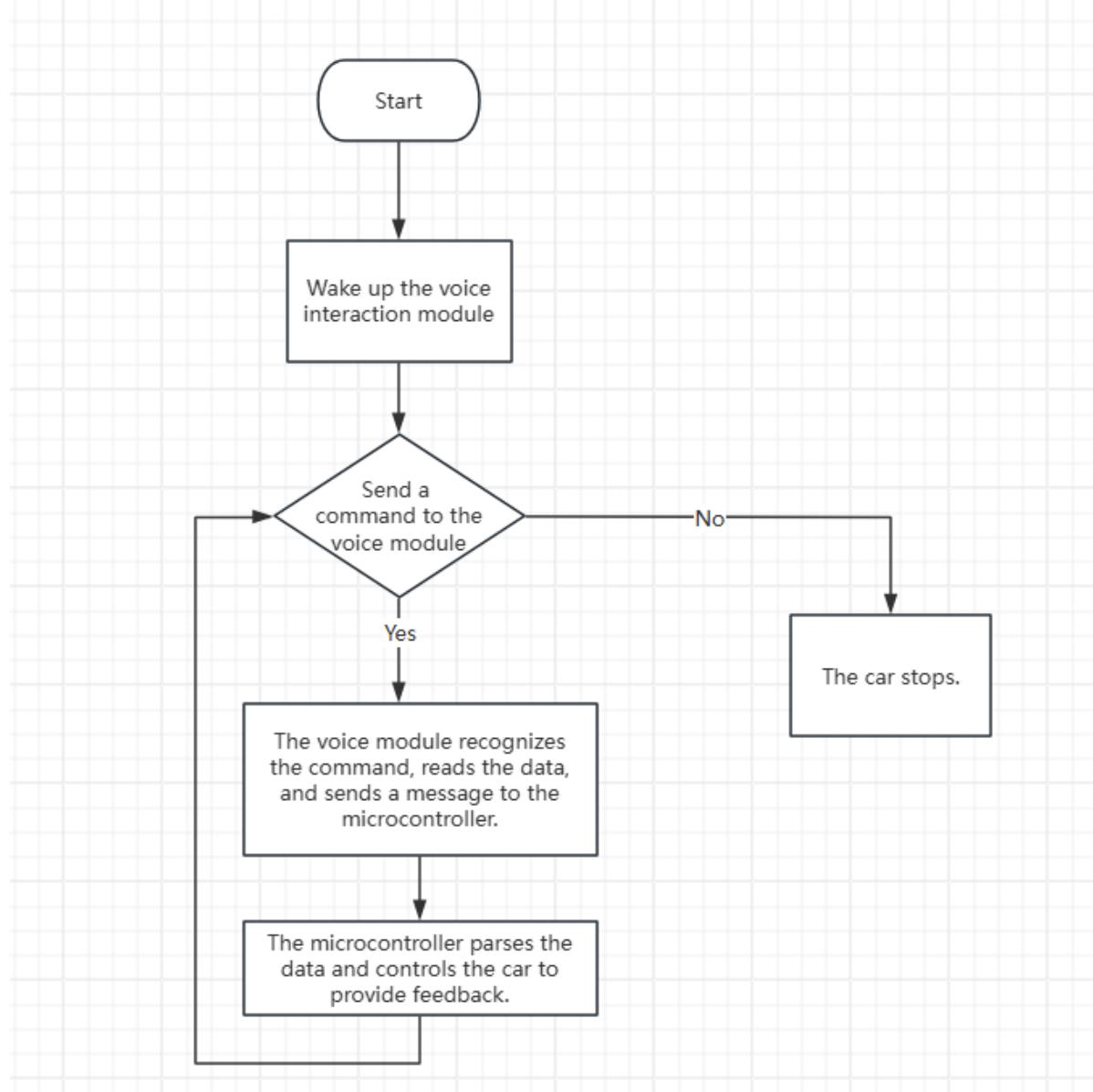
**K230 Wiring** (Note: The wiring diagram below is for position reference only. We provide dual-head PH2.0 4Pin all-black cables for K230 with fool-proof design, so you don't need to worry about wiring issues)

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

### 3. Control Principle

Module	Function
Voice Interaction Module	Communicates with microcontroller and performs voice broadcast

#### Program Flowchart



After power on, we need to first wake up the voice interaction module. When we say "Xiao Ya Xiao Ya" to the voice module, the voice module will send a data frame to the microcontroller, but the host does not need to send data back to the module. Then the voice module will broadcast "I'm here". After hearing the broadcast, we can issue voice commands to the voice module. The voice module and microcontroller will send data frames to each other for communication.

Voice module: After receiving voice commands, it sends data frames to the microcontroller's serial port.

Microcontroller part: Receives data frames and then judges each byte of the data frame. Our data frame protocol is hexadecimal byte stream, format: AA 55 00 XX FB (active broadcast), AA 55 FF XX FB (passive broadcast), where the frame header is the first two bytes AA 55, the frame end is a fixed byte FB, the third byte controls the active and passive broadcast of the voice module, the fourth byte is the command data we need. We return the fourth byte data, and then through processing the data, make the car give corresponding feedback. After the car completes the feedback, the microcontroller will send specified content to the voice module again, and the voice module will broadcast after receiving it.

### III. Main Functions

Mainly introduces user-written functional code, **for detailed code, you can open the project file we provided and enter the Bsp folder to view the source code.**

# 1. User Functions

## Function: `rgb_SetColor`

<b>Function Prototype</b>	<code>void rgb_SetColor(unsigned char LedId, unsigned long color)</code>
Function Description	Set RGB LED color
Input Parameters	LedId - which LED to control, color - color data
Output Parameters	None

## Function: `rgb_SetRGB`

<b>Function Prototype</b>	<code>void rgb_SetRGB(unsigned char LedId, unsigned long red, unsigned long green, unsigned long blue)</code>
Function Description	Set RGB LED color (RGB primary color setting)
Input Parameters	LedId - which LED to control, red - red data, green - green data, blue - blue data
Output Parameters	None

## Function: `get_voice`

<b>Function Prototype</b>	<code>void get_voice(void)</code>
Function Description	Receive data from voice module and control based on this data
Input Parameters	LedId - which LED to control, red - red data, green - green data, blue - blue data
Output Parameters	None

## Function: `Pto_Loop`

<b>Function Prototype</b>	<code>void Pto_Loop(void)</code>
Function Description	User receives and processes protocol transmitted from K230
Input Parameters	None
Output Parameters	None

## Function: `Human_Face_Track`

<b>Function Prototype</b>	<b>void Human_Face_Track(x,w,h)</b>
Function Description	Face tracking case
Input Parameters	Data transmitted from K230
Output Parameters	None

#### Function: Color\_Trace

<b>Function Prototype</b>	<b>void Color_Trace(x,w,h);</b>
Function Description	Color tracking case
Input Parameters	Data transmitted from K230
Output Parameters	None

#### Function: Control\_RGB\_ALL

<b>Function Prototype</b>	<b>void Control_RGB_ALL(RGB_Color_t color)</b>
Function Description	Control RGB color
Input Parameters	Structure-defined parameters representing RGB
Output Parameters	None

## IV. Voice Module Commands

---

命令词 (Command Word)	功能类型 (Function Type)	播报语句 (Broadcast Statement)	播报模式 (Broadcast Mode)	发送协议 (Send Protocol)	接收协议 (Receive Protocol)
WELCOME	欢迎语 (Welcome)	welcome	被 (Passive)	AA 55 01 00 FB	AA 55 01 00 FB
BYE	休息语 (Rest)	bye	主 (Active)	AA 55 02 6F FB	AA 55 02 00 FB
HI-YAHBOOM	唤醒词 (Wake Word)	I am here	主 (Active)	AA 55 03 00 FB	AA 55 03 00 FB
VOLUME-UP	增大音量 (Increase Vol)	Volume Up	主 (Active)	AA 55 04 00 FB	AA 55 04 00 FB
VOLUME-DOWN	减小音量 (Decrease Vol)	Volume down	主 (Active)	AA 55 05 00 FB	AA 55 05 00 FB
MAX-VOLUME	最大音量 (Max Volume)	Maximum volume	主 (Active)	AA 55 06 00 FB	AA 55 06 00 FB
MID-VOLUME	中等音量 (Medium Volume)	Medium volume	主 (Active)	AA 55 07 00 FB	AA 55 07 00 FB
MINI-VOLUME	最小音量 (Min Volume)	Minimum Volume	主 (Active)	AA 55 08 00 FB	AA 55 08 00 FB
START-REPORT	开播报 (Enable Broadcast)	Start broadcasting	主 (Active)	AA 55 09 00 FB	AA 55 09 00 FB
STOP-REPORT	关播报 (Disable Broadcast)	Stop Broadcast	主 (Active)	AA 55 0A 00 FB	AA 55 0A 00 FB
THE-CAR-STOPPED	命令词 (Command)	OK the car has stopped	主 (Active)	AA 55 00 11 FB	AA 55 00 11 FB
THE-CAR-MOVES-FORWARD	命令词 (Command)	OK on track	主 (Active)	AA 55 00 03 FB	AA 55 00 03 FB
THE-CAR-REVERSED	命令词 (Command)	OK backing up	主 (Active)	AA 55 00 05 FB	AA 55 00 05 FB
THE-CAR-TURNED-LEFT	命令词 (Command)	OK the car turns left	主 (Active)	AA 55 00 0F FB	AA 55 00 0F FB

命令词 (Command Word)	功能类型 (Function Type)	播报语句 (Broadcast Statement)	播报模式 (Broadcast Mode)	发送协议 (Send Protocol)	接收协议 (Receive Protocol)
THE-CAR-TURNED-RIGHT	命令词 (Command)	OK the car turns right	主 (Active)	AA 55 00 10 FB	AA 55 00 10 FB
THE-CAR-TURNED-LEFT-IN-PLACE	命令词 (Command)	OK the car is turning left in place	主 (Active)	AA 55 00 08 FB	AA 55 00 08 FB
THE-CAR-TURNED-RIGHT-IN-PLACE	命令词 (Command)	OK the car is turning right in place	主 (Active)	AA 55 00 09 FB	AA 55 00 09 FB
TURN-OFF-THE-LIGHTS	命令词 (Command)	OK turn off the lights	主 (Active)	AA 55 00 4C FB	AA 55 00 4C FB
RED-LIGHT	命令词 (Command)	OK red light on	主 (Active)	AA 55 00 49 FB	AA 55 00 49 FB
GREEN-LIGHT	命令词 (Command)	OK green light	主 (Active)	AA 55 00 4A FB	AA 55 00 4A FB
BLUE-LIGHT	命令词 (Command)	OK blue light on	主 (Active)	AA 55 00 4B FB	AA 55 00 4B FB
YELLOW-LIGHT	命令词 (Command)	OK yellow light on	主 (Active)	AA 55 00 48 FB	AA 55 00 48 FB
START-TRACKING-FACES	命令词 (Command)	OK let us start tracking faces	主 (Active)	AA 55 00 47 FB	AA 55 00 47 FB
PATROL-THE-RED-LINE	命令词 (Command)	OK the red line patrol function is now enabled	主 (Active)	AA 55 00 17 FB	AA 55 00 17 FB
PATROL-GREEN-LINE	命令词 (Command)	OK the green line patrol function is now enabled	主 (Active)	AA 55 00 18 FB	AA 55 00 18 FB
PATROL-BLUE-LINE	命令词 (Command)	OK the blue line patrol function is now enabled	主 (Active)	AA 55 00 19 FB	AA 55 00 19 FB
PATROL-THE-BLACK-LINE	命令词 (Command)	OK the black line patrol has started	主 (Active)	AA 55 00 1B FB	AA 55 00 1B FB
START-TRACKING-COLOR-BLOCKS	命令词 (Command)	OK let us start tracking the color blocks	主 (Active)	AA 55 00 3D FB	AA 55 00 3D FB

## V. Experimental Phenomenon

---

After successfully downloading the program and connecting the voice interaction module wiring, we need to wake up the voice interaction module. The wake-up method is to say "Hi, yahboom" to the voice interaction module, and the module will reply "I'm here".

After hearing the reply, we can issue commands to the car. **This tutorial mainly controls the indicator lights on the expansion board. Car motion control commands are not available (Command issuance and the interval between commands need to be within 20 seconds, otherwise you need to re-wake the voice interaction module.)**

After each command operation, the voice interaction module will have corresponding broadcast replies.

For specific commands, please refer to the attached voice module protocol table.

For program download, refer to [Chapter 3: Development environment setup and usage → 3.Uniflash burning]

The voice interaction module needs to be flashed with firmware compatible with MSPM0 car. For firmware programming, refer to [Chapter 8: Extended application course (Sold separately) → 1. Voice interaction module communication]. If already programmed, no need to repeat programming.