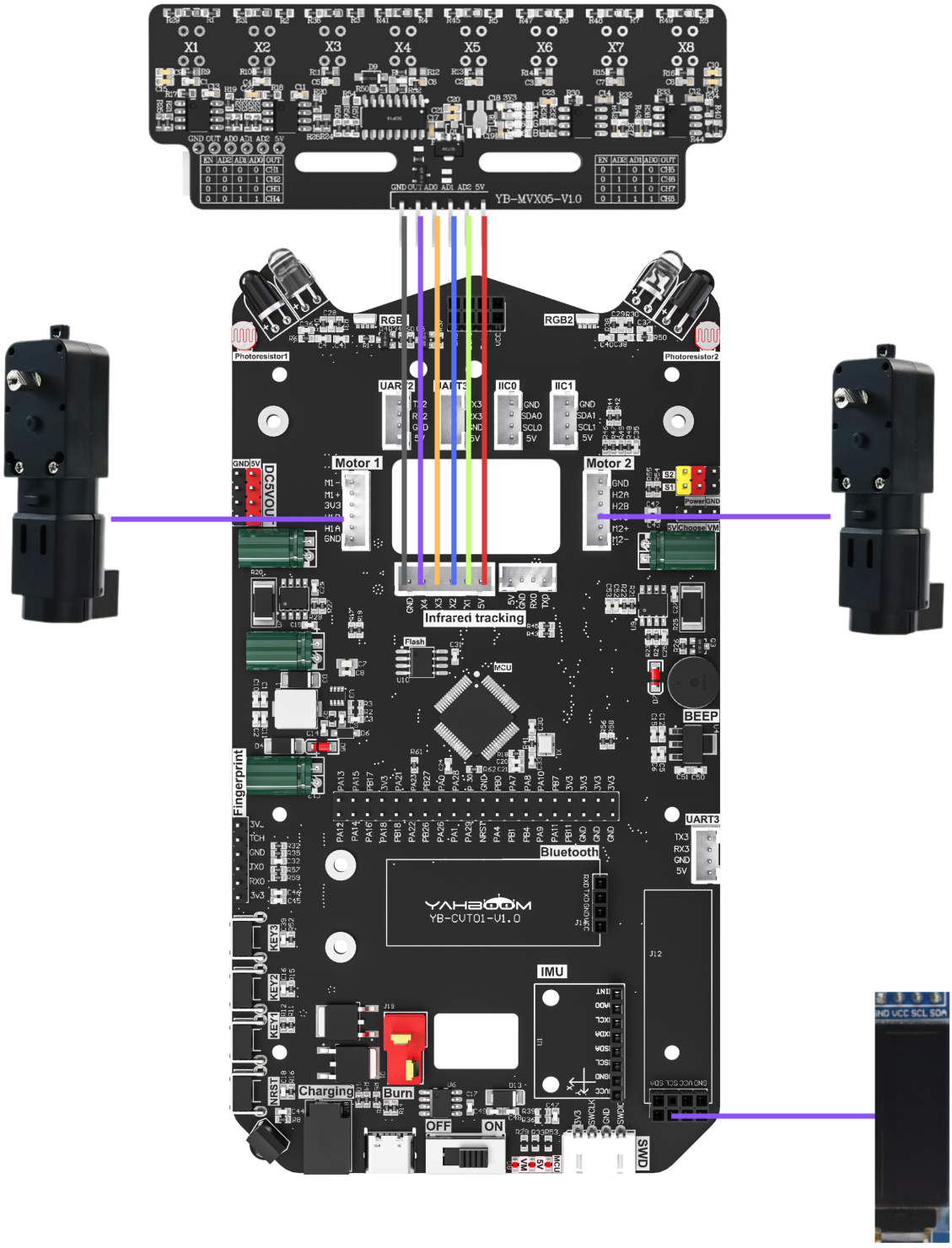
8-channel Grayscale Color Line Tracking

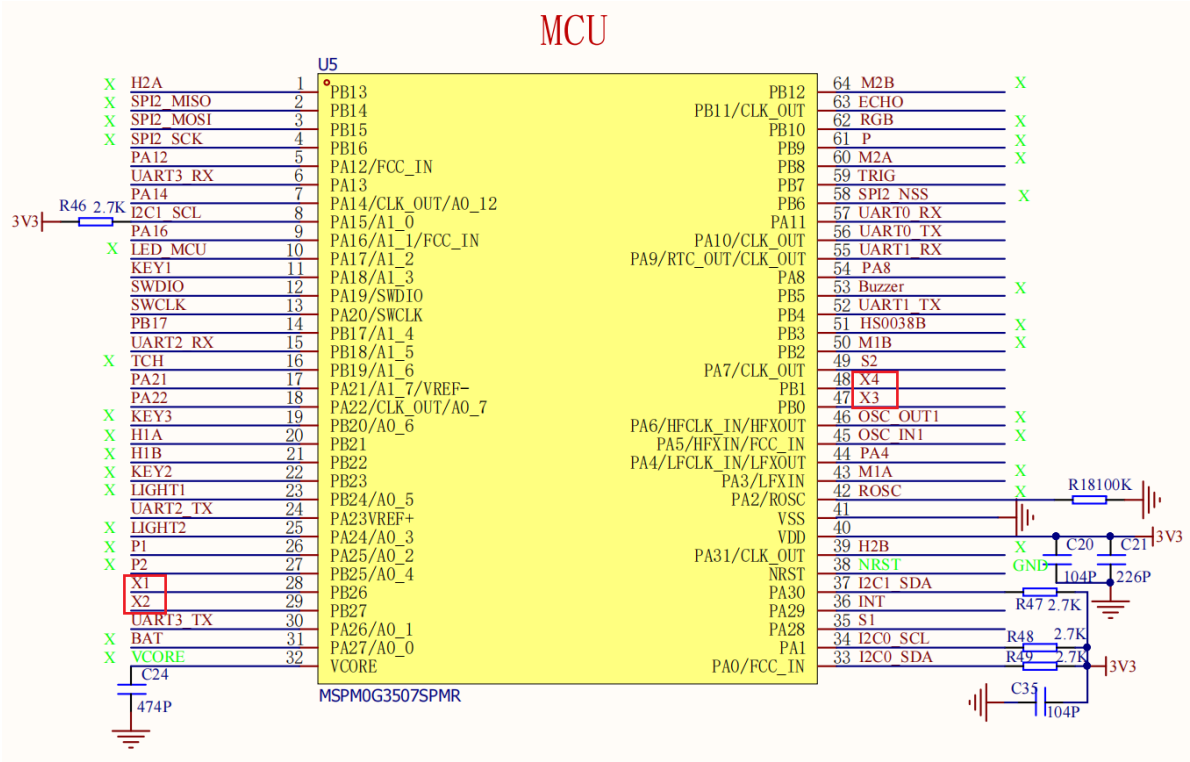
Our store offers four types of 8-channel Grayscale Tracking Modules, distinguished by light color: **Red, Green, Blue, White**. Different colored grayscale modules are suitable for different line inspection scenarios. The provided source code is compatible with all sensors.

Sensor Classification	Red	Green	Blue	White
White Background	Green, Cyan, Blue, Black	Red, Blue, Purple, Black	Yellow, Red, Green, Black	Black
Black Background	Red, Orange, Yellow, White	Green, Yellow, Cyan, White	Blue, Purple, Cyan, White	White

Hardware Wiring

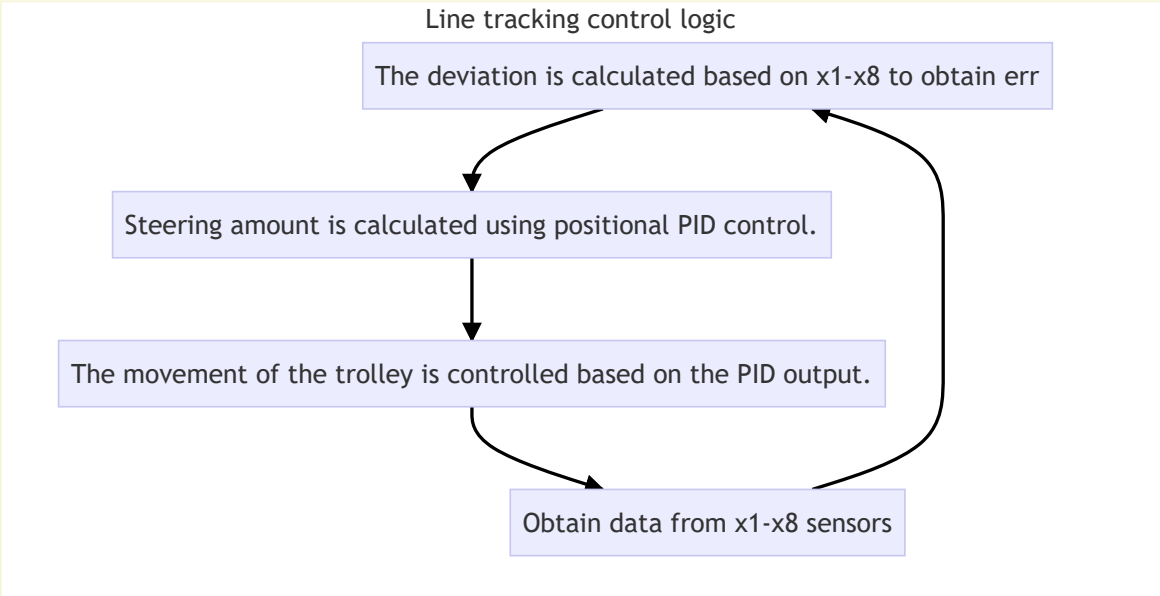
8-channel Grayscale Tracking Module	MSPM0G3507
5V	5V
GND	GND
AD0	X1
AD1	X2
AD2	X3
OUT	X4


X1
X2
X3
X4
X5
X6
X7
X8
YB-MVX05-V1.0
RGB
RGB2
Photoresistor1
Photoresistor2
Motor 1
Motor 2
DC5V OUT
Infrared tracking
Flash
MCU
BEEP
Fingerprint
UART3
Bluetooth
YAHBOOM
YB-CUT01-V1.0
IMU
Charging
Burn
OFF
ON
SWD
KEY1
KEY2
KEY3



Partial Code Analysis

Control Principle



Modify the active level by changing ACTIVE_VALUE to 0/1.

bsp_ir_eight.c

```
/**
 * @brief Read the status of eight infrared sensors
```

```
 * @param ir_results An array of length 8 is used to store the state (0/1) of
each path.
 */
void ReadEightIR(uint8_t ir_results[8]) {

    for (int channel = 0; channel < 8; channel++) {
        // Calculate the C, B, and A values ••corresponding to the current
channel (C is the most significant bit, and A is the least significant bit).
        uint8_t c = (channel >> 2) & 0x01;
        uint8_t b = (channel >> 1) & 0x01;
        uint8_t a = channel & 0x01;
        SET_CHANNEL(c, b, a); // Set channel selection pin
        delay_ms(1);
        // Read the infrared status of the current channel and store it in the
result array.
        ir_results[channel] = READ_IR_OUT() ? 1 : 0;

    }
}
```

bsp_ir_eight.h

```
#define EIGHT_IR_PORT        Eight_IR_PORT   // port
#define EIGHT_IR_AD0_PIN     Eight_IR_AD0_X1_PIN    // Channel selection: A
(least significant bit)
#define EIGHT_IR_AD1_PIN     Eight_IR_AD1_X2_PIN    // Channel selection: B
#define EIGHT_IR_AD2_PIN     Eight_IR_AD2_X3_PIN    // Channel selection: C
(most significant bit)
#define EIGHT_IR_OUT_PIN     Eight_IR_OUT_X4_PIN    // Infrared detection output
pin


// Set the level (0 or 1) of the channel selection pins C, B, and A.
#define SET_CHANNEL(c_val, b_val, a_val) \
    do { \
        GPIO_setPins(EIGHT_IR_PORT, EIGHT_IR_AD2_PIN, c_val); \
        GPIO_setPins(EIGHT_IR_PORT, EIGHT_IR_AD1_PIN, b_val); \
        GPIO_setPins(EIGHT_IR_PORT, EIGHT_IR_AD0_PIN, a_val); \
    } while(0)
```

Main Functions

ReadEightIR

Function Prototype	**void ReadEightIR(uint8_t ir_results[8])**
Function Description	Reads the status of eight infrared sensors and stores the status (0 or 1) of each channel into a specified array. Channels are selected by calculating the corresponding C, B, and A values. The status is read and stored after a 1ms delay.
Input Parameters	ir_results: An array of type uint8_t with a length of 8, used to store the status of eight infrared sensors (0 indicates no detection, 1 indicates detection).
Return Value	None

Experimental Phenomena

After connecting the car to the OLED module and programming the MSPM0, place the car on a map with a white background and black lines. The car will automatically start tracking, and the sensor data will be displayed on the OLED.