Data Reading

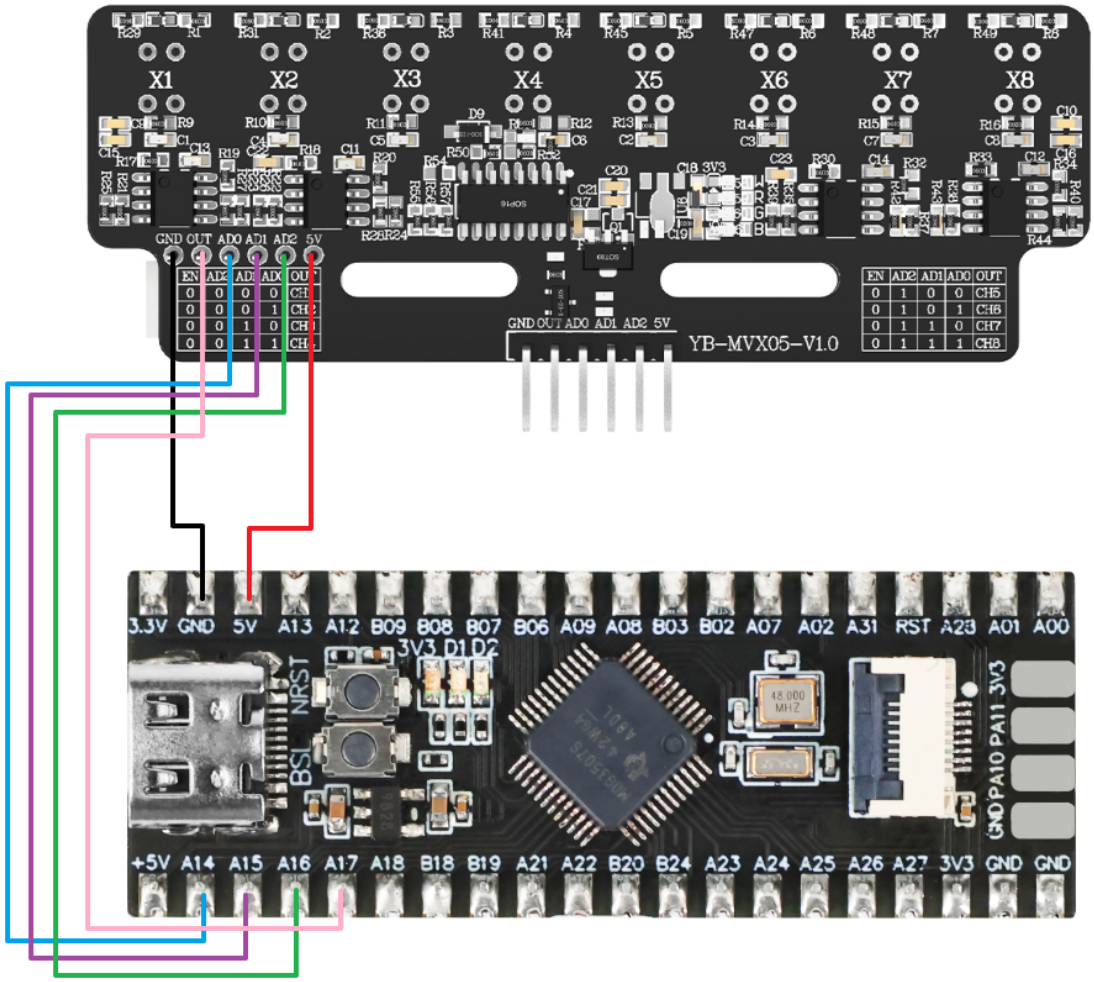
1. Quick Start

This tutorial explains how to use the MSPM0G3507 motherboard to read the digital information from an eight-channel grayscale tracking module and view the print results using a serial port assistant.

The hardware used in this tutorial is the MSPM0G3507 core board and eight-channel grayscale tracking module sold by Yabo. After connecting the wires according to the wiring diagram below, you can then program the motherboard to obtain the data. Modules other than Yabo's are for reference only.

2. Hardware Wiring

The MSPM0G3507's USB port needs to be connected to the computer's USB port using a Type-C port.

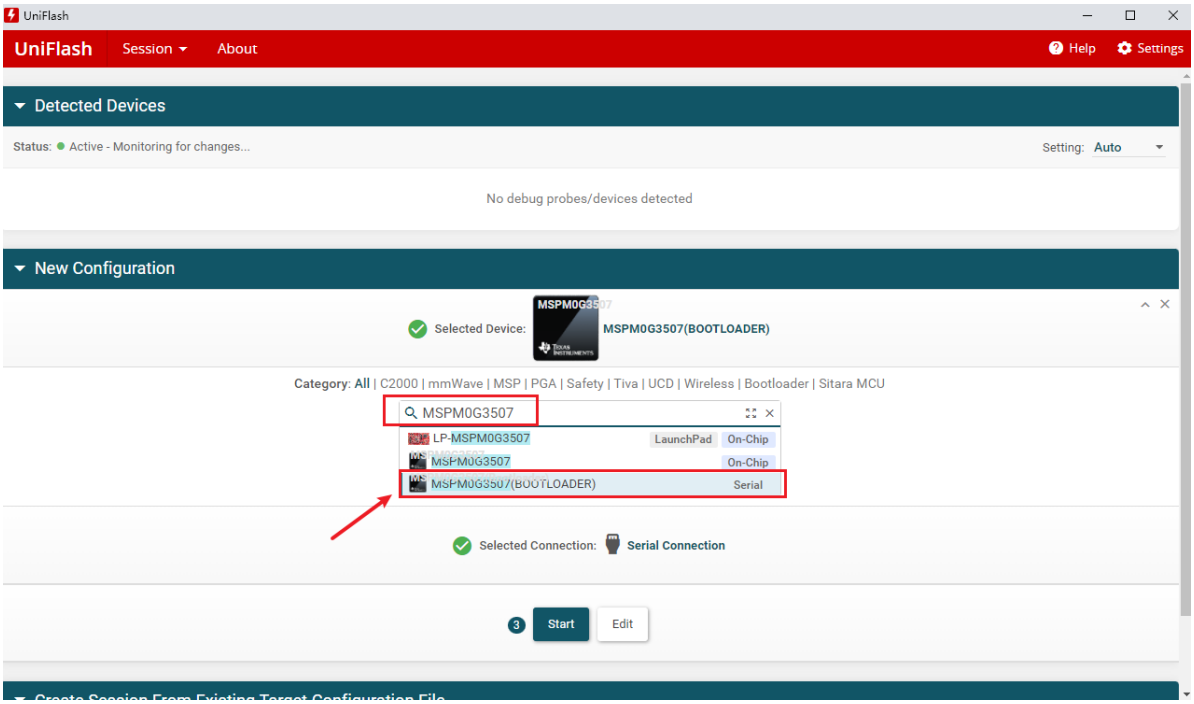


Eight-channel grayscale line-following module	MSPM0G3507
5V	5V
GND	GND
AD0	PA14
AD1	PA15
AD2	PA16
OUT	PA17

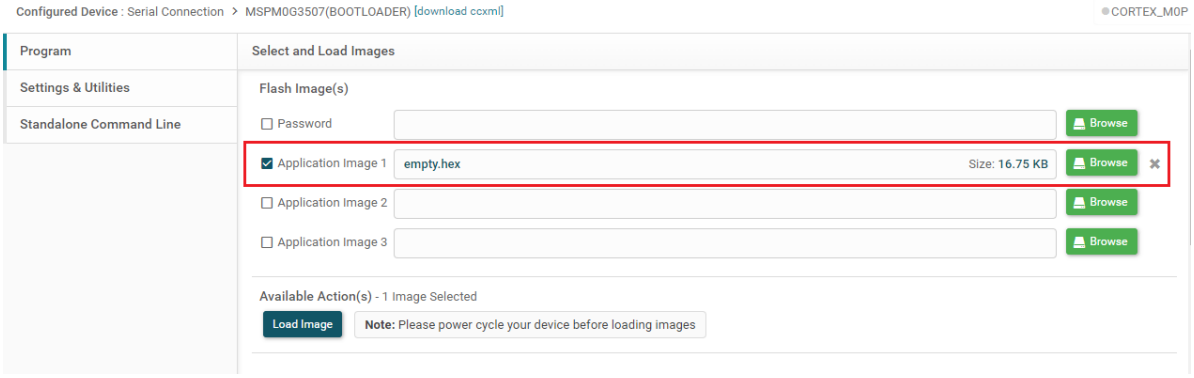
The 8-channel grayscale module sold by Yabo uses an XH2.54 to 6-pin DuPont cable for connection:

3. Usage Instructions

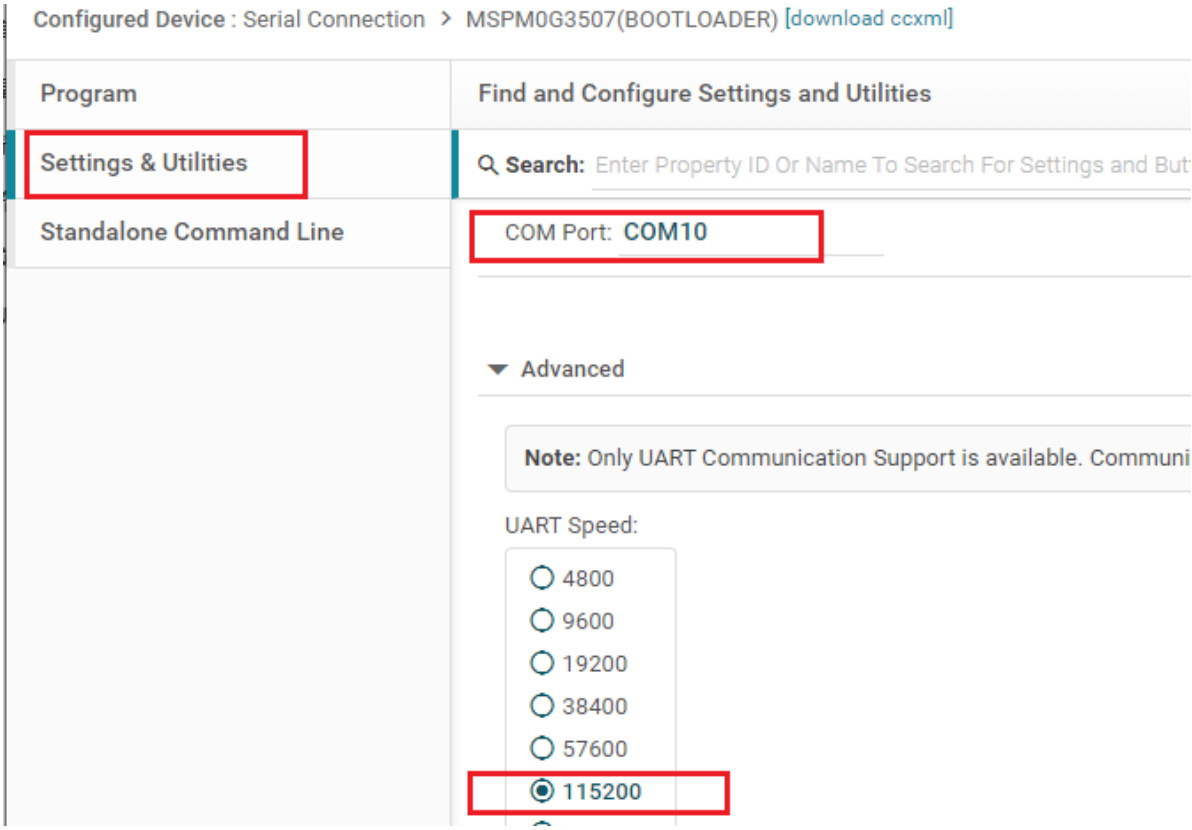
1. Open the UniFlash programming tool, search for MSPM0G3507, and click the third result. If this software is not available, you need to refer to the MSPM0G3507 core board documentation to learn the basic usage and programming of the motherboard.



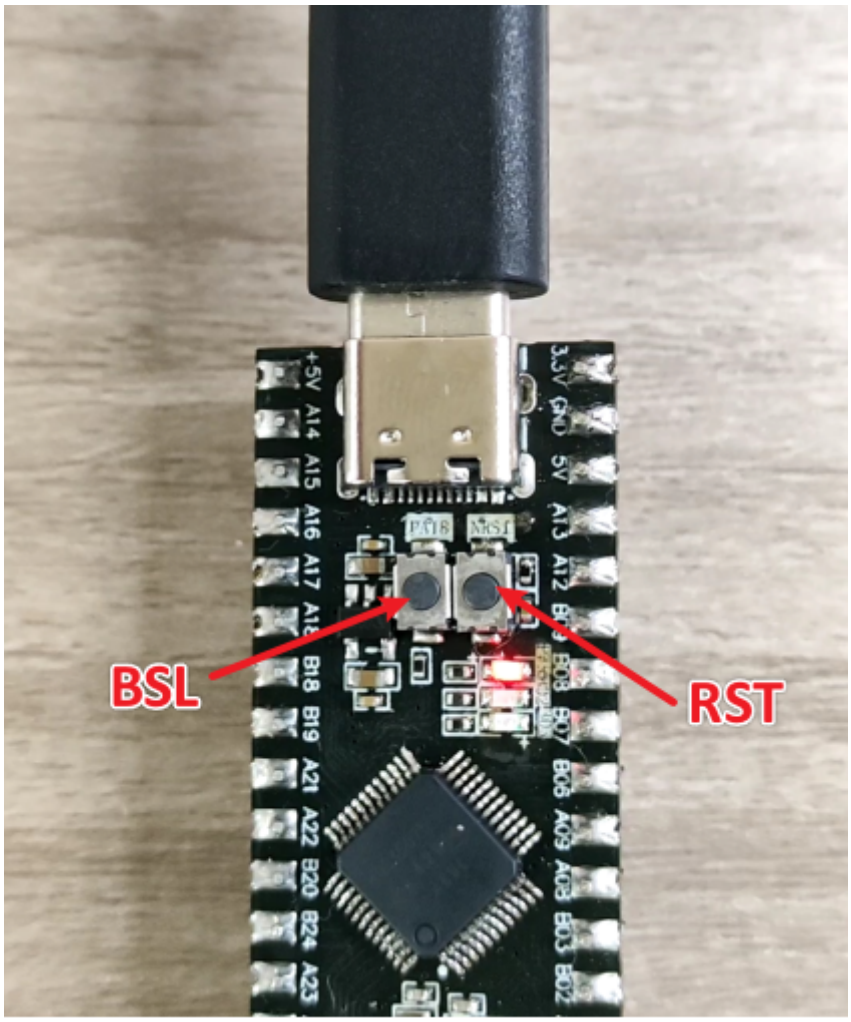
2. In the Program navigation page, select the hex file in the source code.



3. Select the COM port and baud rate.

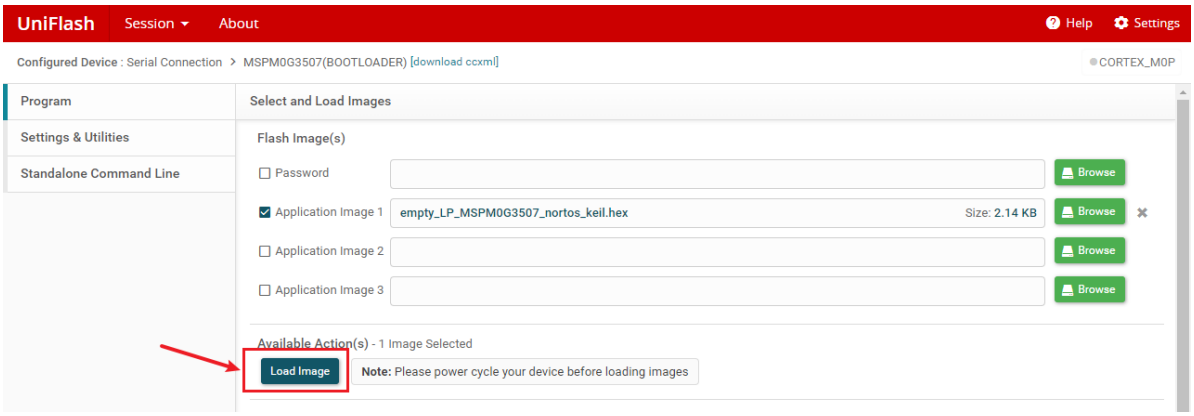


4. Press and hold the BSL button, then press and hold the RST button for about 1 second. Release the RST button, then release the BSL button. You are now in burning mode.

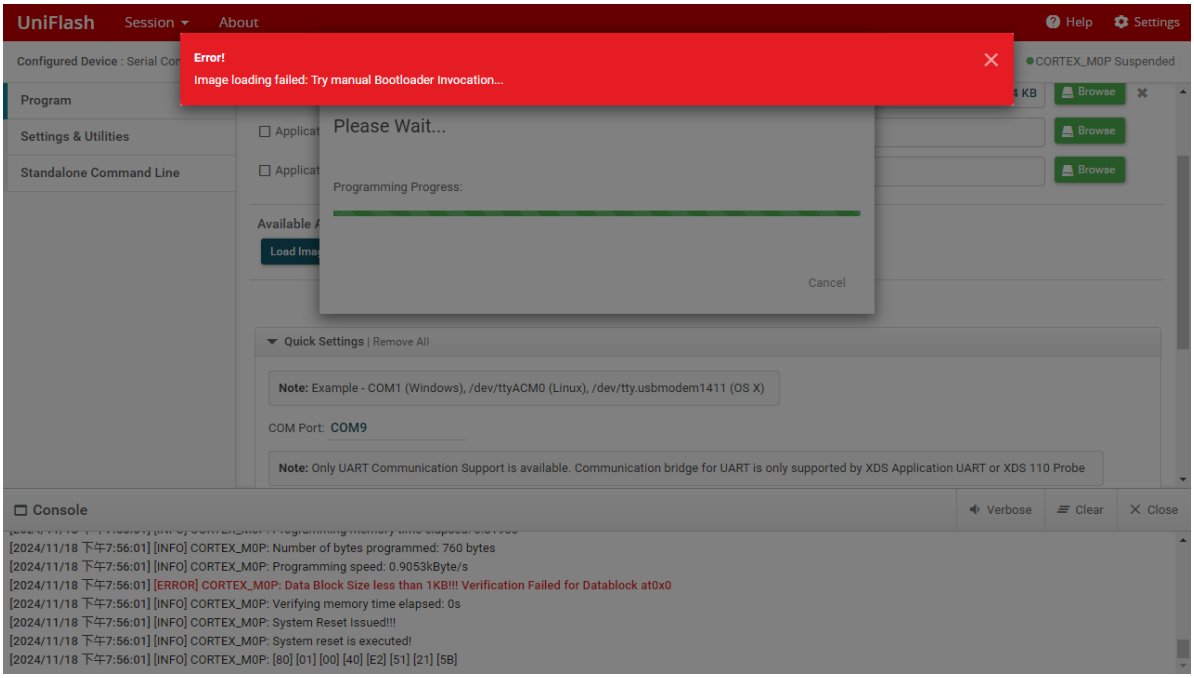


After entering the burning mode, click "Load image" to download.

Note: The download must be performed within 10 seconds of the development board entering burning mode; otherwise, the download will fail.



If you encounter the same error message as shown in the image below, "Image loading falled: Try manual Bootloader invocation...", the download has actually been successful. This is a bug in the download software.



5. Press the reset button (NRST) to restart the motherboard and run the program. Open the serial port assistant and view the printed information.

4. Phenomenon and Results

In the serial port assistant, you can see the continuously printed digital values of the eight-channel grayscale tracking module. X1 corresponds to the X1 indicator on the module; when the X1 indicator is lit, the corresponding value is 1.

```
[2025-11-08 17:18:48.215]# RECV ASCII>
[ X1:1 ][ X2:1 ][ X3:1 ][ X4:1 ][ X5:1 ][ X6:1 ][ X7:1 ][ X8:1 ]


[2025-11-08 17:18:48.312]# RECV ASCII>
[ X1:1 ][ X2:1 ][ X3:1 ][ X4:1 ][ X5:1 ][ X6:1 ][ X7:1 ][ X8:1 ]


[2025-11-08 17:18:48.407]# RECV ASCII>
[ X1:1 ][ X2:1 ][ X3:1 ][ X4:1 ][ X5:1 ][ X6:1 ][ X7:1 ][ X8:1 ]


[2025-11-08 17:18:48.502]# RECV ASCII>
[ X1:1 ][ X2:1 ][ X3:1 ][ X4:1 ][ X5:1 ][ X6:1 ][ X7:1 ][ X8:1 ]


[2025-11-08 17:18:48.613]# RECV ASCII>
[ X1:1 ][ X2:1 ][ X3:1 ][ X4:1 ][ X5:1 ][ X6:1 ][ X7:1 ][ X8:1 ]


[2025-11-08 17:18:48.709]# RECV ASCII>
[ X1:1 ][ X2:1 ][ X3:1 ][ X4:1 ][ X5:1 ][ X6:1 ][ X7:1 ][ X8:1 ]
```

5. Code Explanation

```c
// empty.c
int main(void)
{
    SYSCFG_DL_init();
    USART_Init();

    while(1)
    {
        Grayscale_Sensor_Read_All(g_sensor_data);

        for (i = 0; i < GRAYSCALE_SENSOR_CHANNELS; i++)
        {
            printf("[ %s:%d ]", sensor_labels[i], g_sensor_data[i]);
        }
        printf("\n");
        delay_ms(30);
    }
}
```

- `main`: The main entry point of the program. First, `SYSCFG_DL_init()` is called. This function, generated by TI's SysConfig tool, initializes all hardware peripherals, including GPIO and UART. Then, `USART_Init()` is called to enable serial port interrupts. Within an infinite loop, the program continuously calls `Grayscale_Sensor_Read_All()` to acquire sensor data and prints it out using `printf`. The `printf` function has been redirected to the UART, so the data is sent via the serial port.

```c
// grayscale_sensor.c
void Grayscale_Sensor_Read_All(uint16_t* sensor_values)
{
```

```
    uint8_t i;
    for (i = 0; i < GRAYSCALE_SENSOR_CHANNELS; i++)
    {
        _select_channel(i);
        _delay_us(50);
        sensor_values[i] = Read_OUT_value();
    }
}
static void _select_channel(uint8_t channel)
{
    SENSOR_AD0_WRITE((channel >> 0) & 0x01);
    SENSOR_AD1_WRITE((channel >> 1) & 0x01);
    SENSOR_AD2_WRITE((channel >> 2) & 0x01);
}
```

- `Grayscale_Sensor_Read_All`: Traverse all 8 sensor channels, passing The `_select_channel` function selects a channel, reads its value, and stores it in an array.
- `_select_channel`: The `SENSOR_ADx_WRITE` macro controls the GPIO level to select the corresponding sensor. These macros ultimately call the `DL_GPIO_setPins` and `DL_GPIO_clearPins` functions in the TI DriverLib library.