

Data Reading

Data Reading

1. Quick Start
2. Hardware Wiring
3. Usage Method
4. Observation and Results
5. Code Explanation

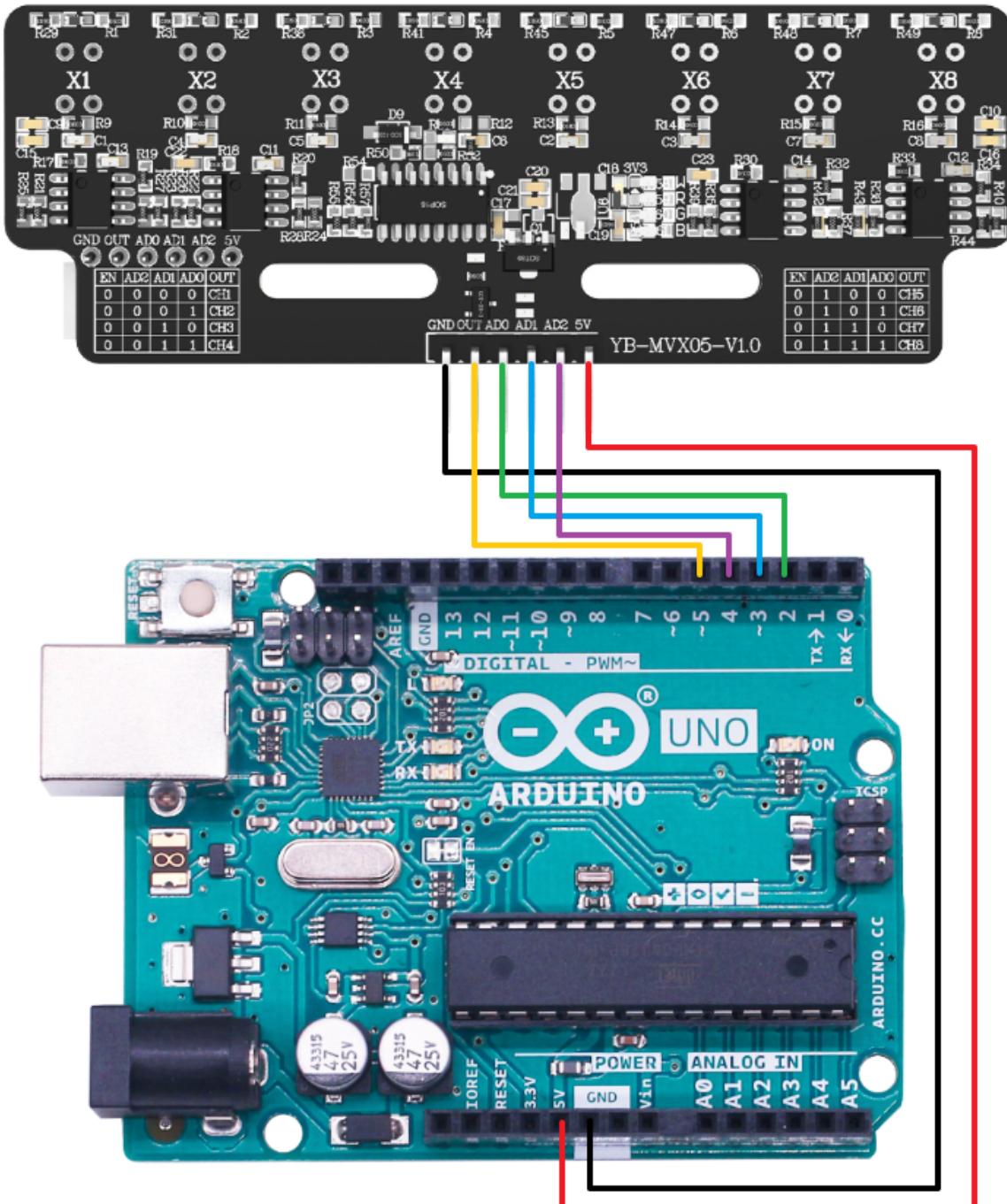
1. Quick Start

This tutorial explains how to use the Arduino UNO motherboard to read the digital information from an eight-channel grayscale line-following module and print the results via a serial monitor.

The hardware used in this tutorial is an Arduino UNO and an eight-channel grayscale line-following module sold by Yabo. After connecting the wires according to the wiring diagram below, you can obtain the data by programming the Arduino UNO. Modules other than Yabo's are for reference only.

2. Hardware Wiring

The Arduino UNO's USB-B interface needs to be connected to the computer's USB port using a data cable.



Eight-channel grayscale line-following module

Arduino UNO

5V		5V
GND		GND
AD0		2
AD1		3
AD2		4
OUT		5

The standard cable for the eight-channel grayscale tracking module sold by Yabo is an XH2.54 to 6-pin DuPont cable. One end of the XH2.54 ribbon cable should be connected to the eight-channel grayscale module, and the other end of the DuPont cable can be connected normally as shown in the picture above:

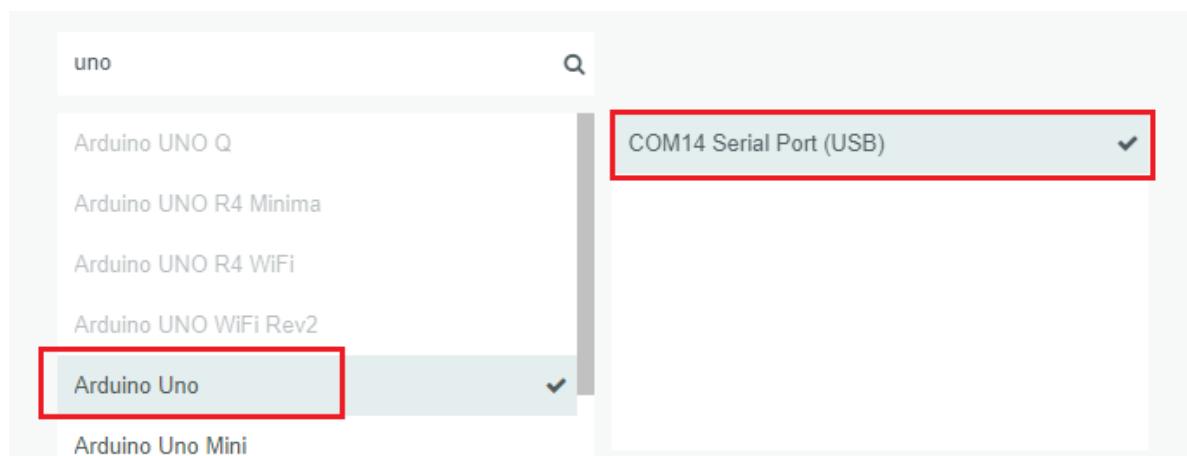


3. Usage Method

1. Open the `Grayscale_Read.ino` file in the `Grayscale_Read` folder using the Arduino IDE.
2. Connect the Arduino UNO to the computer using a data cable.
3. In the menu bar at the top left of the Arduino IDE, click the port selection box, and then select the serial port of the Arduino you are connecting.



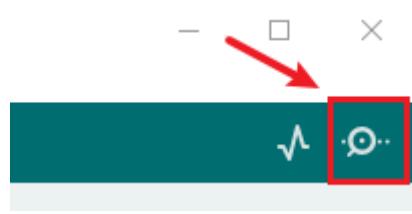
4. If this is the first connection and there is a COM port number, but no Arduino model is specified, manually select the model.



5. Next, click the "Upload" button on the toolbar. The program will begin compiling and burning to the Arduino UNO.



6. After the upload is complete, click the "Serial Monitor" icon in the upper right corner of the toolbar.



7. In the upper right corner of the serial monitor window, ensure the baud rate is set to **115200**.

4. Observation and Results

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

```
[ X1:0 ][ X2:0 ][ X3:0 ][ X4:0 ][ X5:0 ][ X6:0 ][ X7:0 ][ X8:0 ]
[ X1:0 ][ X2:0 ][ X3:0 ][ X4:0 ][ X5:0 ][ X6:0 ][ X7:0 ][ X8:0 ]
[ X1:0 ][ X2:0 ][ X3:0 ][ X4:0 ][ X5:0 ][ X6:0 ][ X7:0 ][ X8:0 ]
[ X1:0 ][ X2:0 ][ X3:0 ][ X4:0 ][ X5:0 ][ X6:0 ][ X7:0 ][ X8:0 ]
[ X1:0 ][ X2:0 ][ X3:0 ][ X4:0 ][ X5:0 ][ X6:0 ][ X7:0 ][ X8:0 ]
[ X1:0 ][ X2:0 ][ X3:0 ][ X4:0 ][ X5:0 ][ X6:0 ][ X7:0 ][ X8:0 ]
[ X1:0 ][ X2:0 ][ X3:0 ][ X4:0 ][ X5:0 ][ X6:0 ][ X7:0 ][ X8:0 ]
[ X1:0 ][ X2:0 ][ X3:0 ][ X4:0 ][ X5:0 ][ X6:0 ][ X7:0 ][ X8:0 ]
[ X1:0 ][ X2:0 ][ X3:0 ][ X4:0 ][ X5:0 ][ X6:0 ][ X7:0 ][ X8:0 ]
[ X1:0 ][ X2:0 ][ X3:0 ][ X4:0 ][ X5:0 ][ X6:0 ][ X7:0 ][ X8:0 ]
```

5. Code Explanation

```
//GPIO required for initializing the grayscale sensor
void Grayscale_Sensor_Init(void)
{
    pinMode(SENSOR_AD0_PIN, OUTPUT);
    pinMode(SENSOR_AD1_PIN, OUTPUT);
    pinMode(SENSOR_AD2_PIN, OUTPUT);
    pinMode(SENSOR_OUT_PIN, INPUT);
}

//Select sensor channel
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);
}

//Read the grayscale values of all 8 channels.
void Grayscale_Sensor_Read_All(uint16_t* sensor_values)
{
    uint8_t i;
    for (i = 0; i < GRAYSCALE_SENSOR_CHANNELS; i++)
    {
        _select_channel(i);
        delayMicroseconds(50);
        sensor_values[i] = Read_OUT_value();
    }
}
```

- `Grayscale_Sensor_Init`: Called at the start of the program, sets the control pins (AD0-AD2) to output mode and the data pin (OUT) to input mode.
- `_select_channel`: An internal function that selects one of the eight sensors by controlling the level combination of the three pins AD0, AD1, and AD2.
- `Grayscale_sensor_Read_A11`: An external function that iterates through all 8 sensors by repeatedly calling `_select_channel` and reads their values sequentially, storing them in an array.