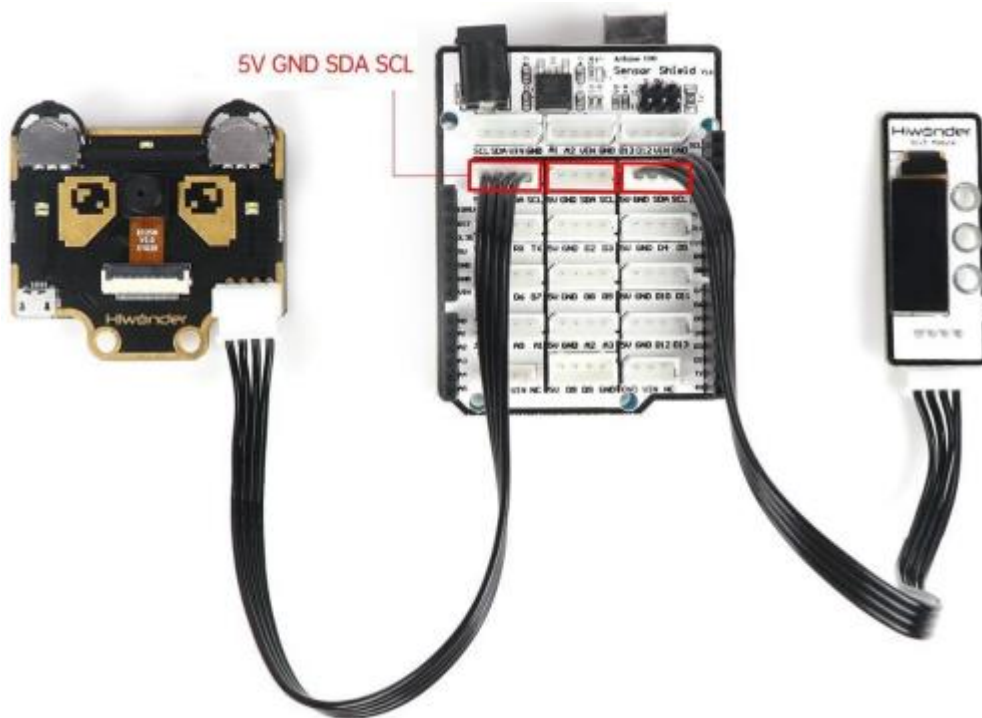


Lesson 7 Image Classification Integration with Arduino

1. Preparation

Connect WonderCam and OLED Display Module to Arduino Expansion Board using 4-Pin cable.

WonderCam and OLED Display Module can be connected to any IIC port (5V GND SDA SCL).



2. Learning Objectives

- ① To know the connectivity of WonderCam and OLED Display Module on Arduino Expansion Board
- ② To understand the program logic.

3. Programming Plan

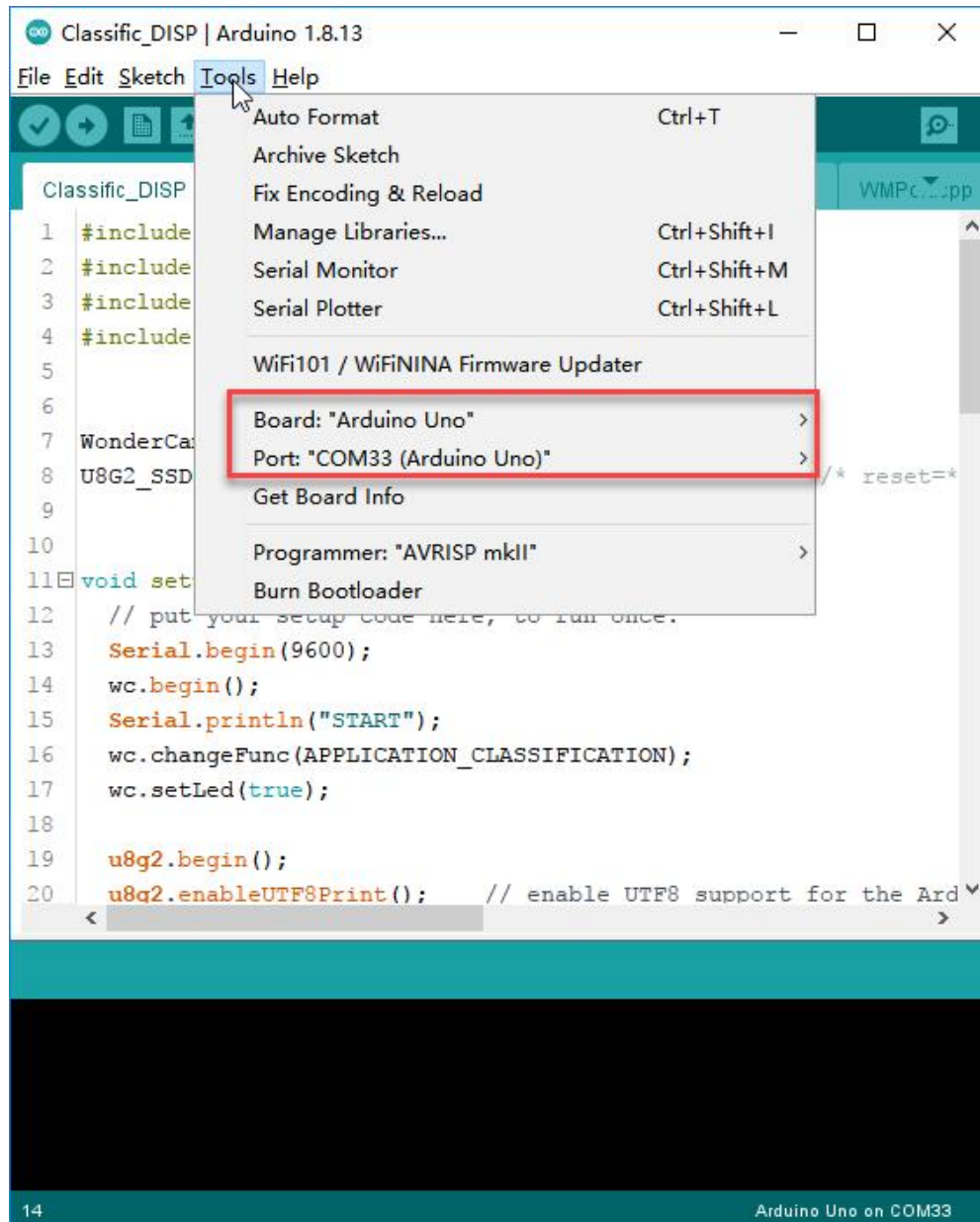
In this lesson, we are using the Image Classification function to integrate with Arduino. The program will initialize the Vision module followed by classifying garbage cans. When a mark is detected, it will proceed to identify the trust value and display the corresponding name on the OLED screen.

4. Compiling Program and Upload



This full program code can be found in folder "Image Classification Program" in "03_Classific_DISP".

- 1) Connect Arduino UNO board to computer.
- 2) In "Image Classification Program" in "03_Classific_DISP", double click into program.
- 3) In Arduino IDE program, open Tools in menu and select corresponding Development Board and port. (Port number shown in this lesson may differ in individual computer environment).



4) In Arduino IDE Program, click  button on the menu and wait for compilation process to complete.

```


1  #include <Arduino.h>
2  #include "WonderCam.h"
3  #include <U8g2lib.h>
4  #include <Wire.h>
5
6
7  WonderCam wc;
8  U8G2_SSD1306_128X32_UNIVISION_1_HW_I2C u8g2(U8G2_R0, /* reset=*/
9
10
11 void setup() {
12     // put your setup code here, to run once:
13     Serial.begin(9600);
14     wc.begin();
15     Serial.println("START");
16     wc.changeFunc(APPLICATION_CLASSIFICATION);
17     wc.setLed(true);
18
19     u8g2.begin();
20     u8g2.enableUTF8Print(); // enable UTF8 support for the Ard

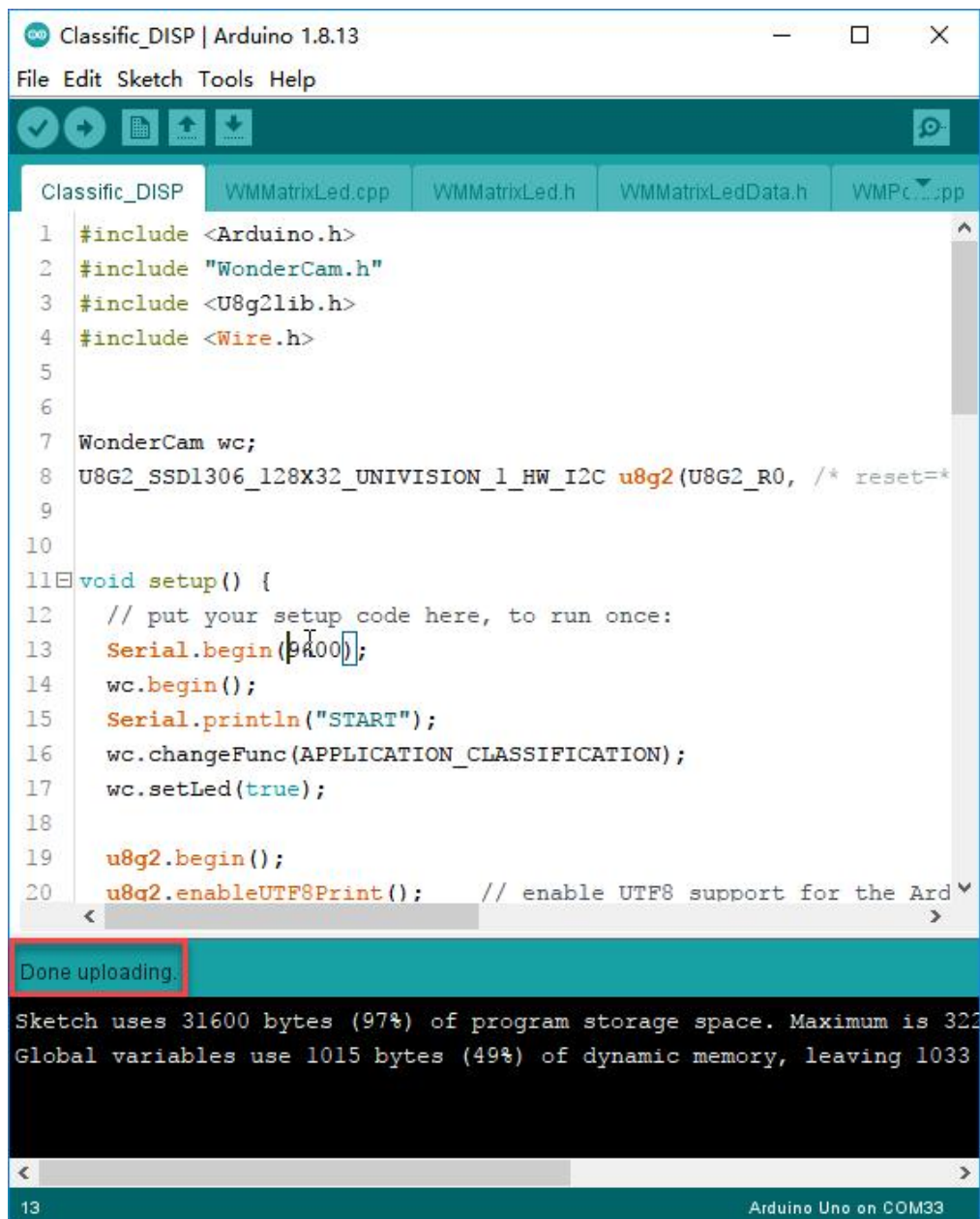
```

Done compiling.

Sketch uses 31600 bytes (97%) of program storage space. Maximum is 32768 bytes.
Global variables use 1015 bytes (49%) of dynamic memory, leaving 1043 bytes for the global variable heap.

14 Arduino Uno on COM33

- 5) Click  button to upload program to UNO Development Board. Wait for uploading process to complete.
- 6) During Uploading process, do not unplug or move the USB Cable to prevent transmission failure.



```

Classific_DISP | Arduino 1.8.13
File Edit Sketch Tools Help

Classific_DISP WMMatrixLed.cpp WMMatrixLed.h WMMatrixLedData.h WMPc...pp
1 #include <Arduino.h>
2 #include "WonderCam.h"
3 #include <U8g2lib.h>
4 #include <Wire.h>
5
6
7 WonderCam wc;
8 U8G2_SSD1306_128X32_UNIVISION_1_HW_I2C u8g2(U8G2_R0, /* reset=*
9
10
11 void setup() {
12     // put your setup code here, to run once:
13     Serial.begin(9600);
14     wc.begin();
15     Serial.println("START");
16     wc.changeFunc(APPLICATION_CLASSIFICATION);
17     wc.setLed(true);
18
19     u8g2.begin();
20     u8g2.enableUTF8Print(); // enable UTF8 support for the Ard

```

Done uploading.

Sketch uses 31600 bytes (97%) of program storage space. Maximum is 32256 bytes. Global variables use 1015 bytes (49%) of dynamic memory, leaving 1033 bytes free.

13 Arduino Uno on COM33

5. Result

Note: WonderCam module has a built-in 12 card sorting model at factory default setting, which can be used with the garbage template provided.

Once program had been uploaded, WonderCam will automatically switch to Image Classification interface. When a number is recognized, it will present the corresponding name presented on OLED display module.