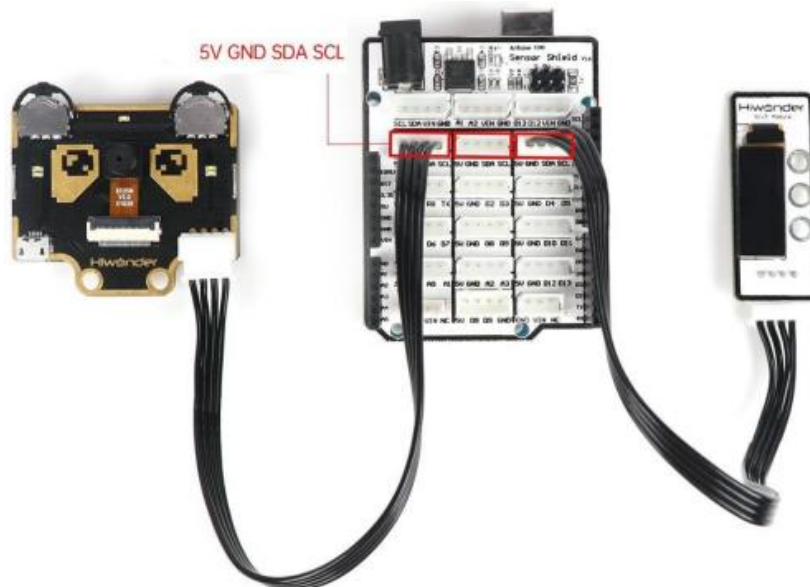


Lesson 8 Feature Learning Integration with Arduino

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

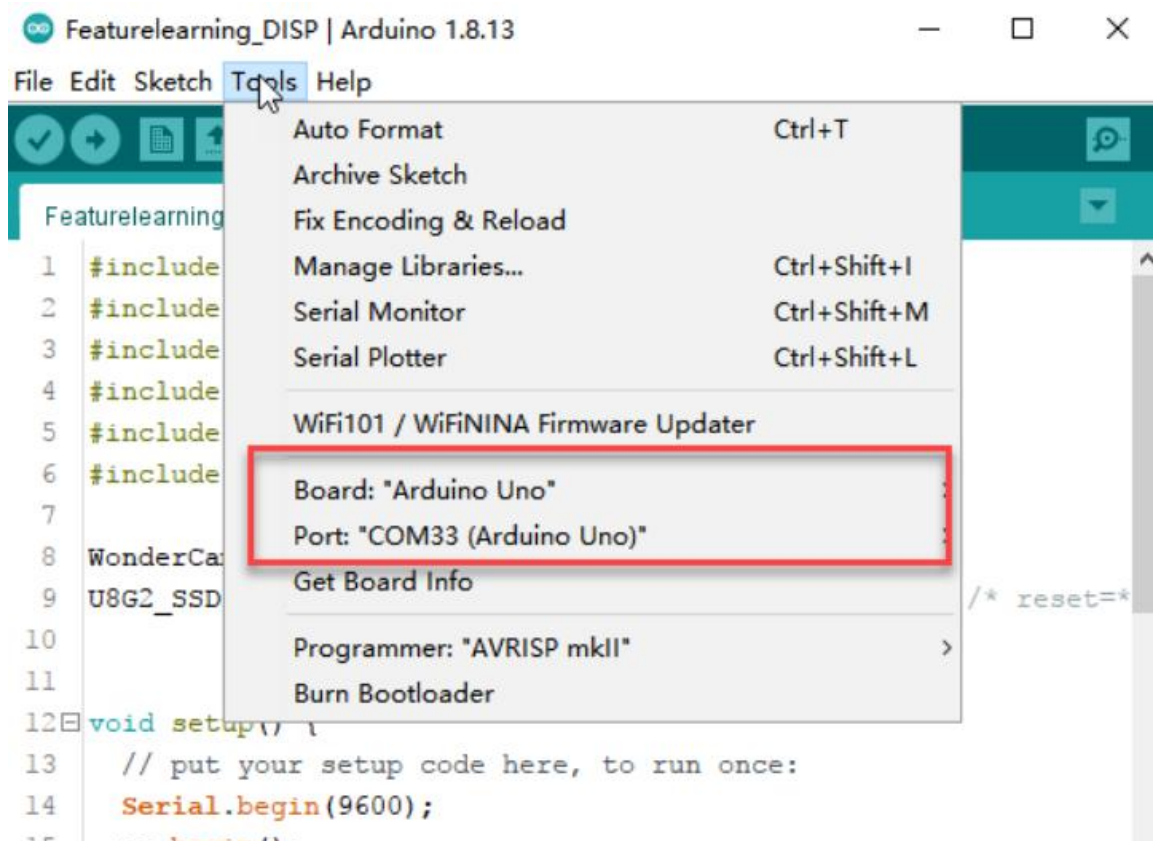
To initialize WonderCam when power-up. When a feature image is presented to WonderCam, WonderCam will make comparison and match with the images it had learned before and present the feature ID on OLED Display Module if a match is made.


4. Compiling Program and Upload

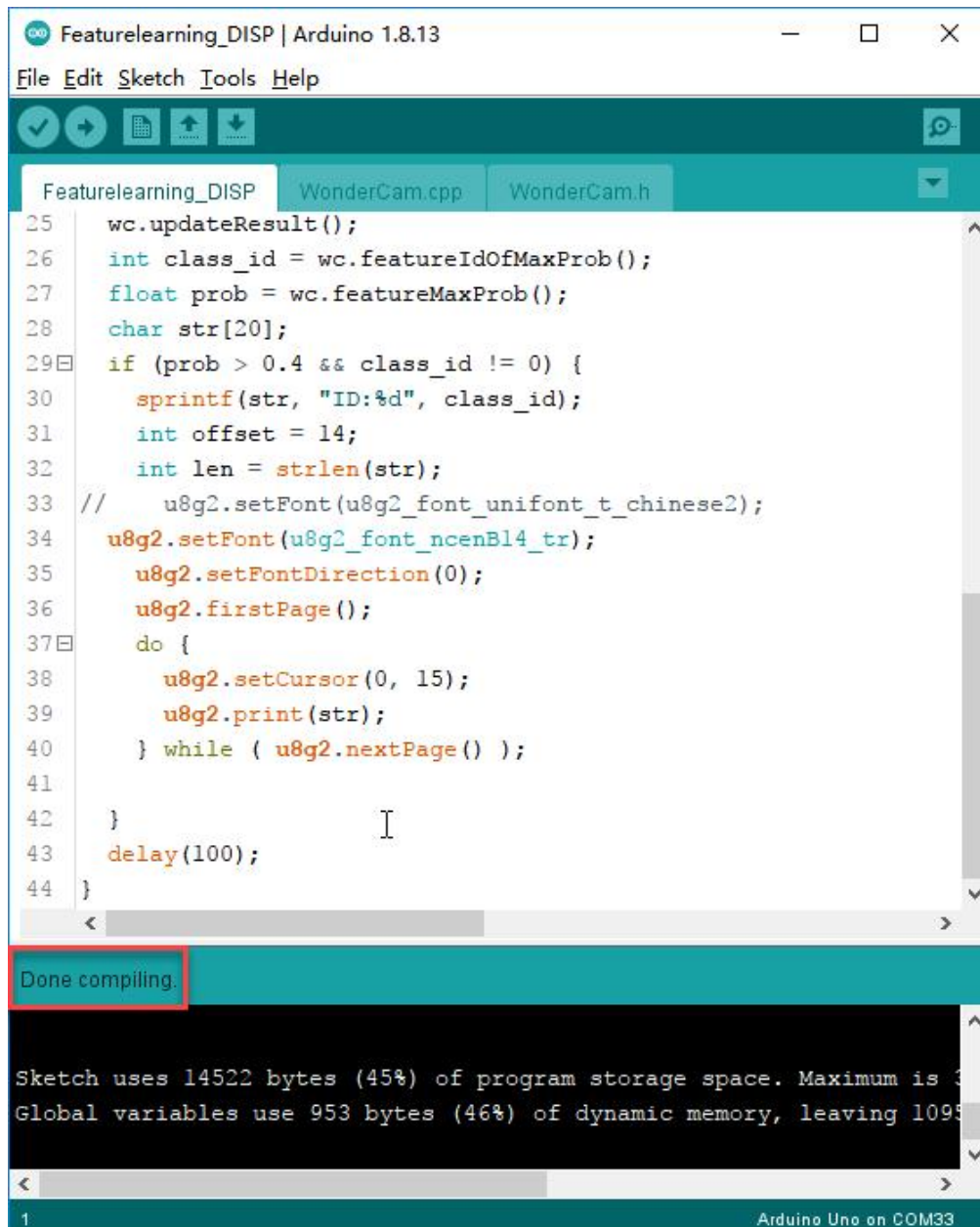


This full program code can be found in folder “Feature Learning Program” in “04_Featurelearning_DISP”.

- 1) Connect Arduino UNO board to computer.
- 2) In “Feature Learning Program” in “004_Featurelearning_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 differs in individual computer environment).



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




```

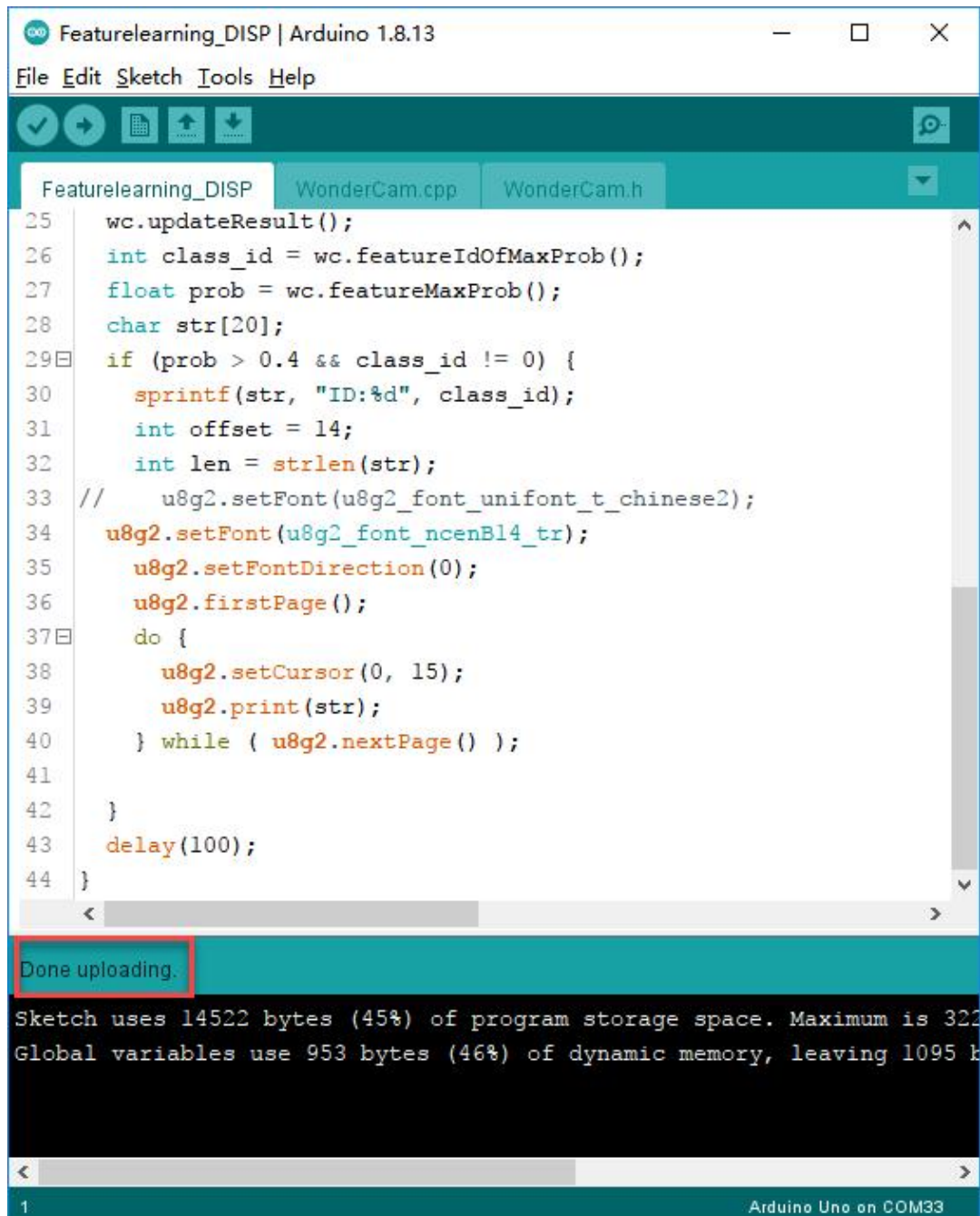
25  wc.updateResult();
26  int class_id = wc.featureIdOfMaxProb();
27  float prob = wc.featureMaxProb();
28  char str[20];
29  if (prob > 0.4 && class_id != 0) {
30      sprintf(str, "ID:%d", class_id);
31      int offset = 14;
32      int len = strlen(str);
33      // u8g2.setFont(u8g2_font_unifont_t_chinese2);
34      u8g2.setFont(u8g2_font_ncenB14_tr);
35      u8g2.setFontDirection(0);
36      u8g2.firstPage();
37      do {
38          u8g2.setCursor(0, 15);
39          u8g2.print(str);
40      } while ( u8g2.nextPage() );
41
42  }
43  delay(100);
44  }
    
```

Done compiling.

Sketch uses 14522 bytes (45%) of program storage space. Maximum is 32768 bytes. Global variables use 953 bytes (46%) of dynamic memory, leaving 1093 bytes free.

1 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.



```
Featurelearning_DISP | Arduino 1.8.13
File Edit Sketch Tools Help

Featurelearning_DISP WonderCam.cpp WonderCam.h
25 wc.updateResult();
26 int class_id = wc.featureIdOfMaxProb();
27 float prob = wc.featureMaxProb();
28 char str[20];
29 if (prob > 0.4 && class_id != 0) {
30     sprintf(str, "ID:%d", class_id);
31     int offset = 14;
32     int len = strlen(str);
33     // u8g2.setFont(u8g2_font_unifont_t_chinese2);
34     u8g2.setFont(u8g2_font_ncenB14_tr);
35     u8g2.setFontDirection(0);
36     u8g2.firstPage();
37     do {
38         u8g2.setCursor(0, 15);
39         u8g2.print(str);
40     } while ( u8g2.nextPage() );
41 }
42 }
43 delay(100);
44 }

Done uploading.

Sketch uses 14522 bytes (45%) of program storage space. Maximum is 322
Global variables use 953 bytes (46%) of dynamic memory, leaving 1095 b

1 Arduino Uno on COM33
```

5. Result

* Please refer to Lesson 8 Feature Learning on how to program Visual Line Following.

Once program had been uploaded, WonderCam will automatically switch to Feature Learning interface. When a object is recognized, it will present the corresponding ID presented on OLED display module.