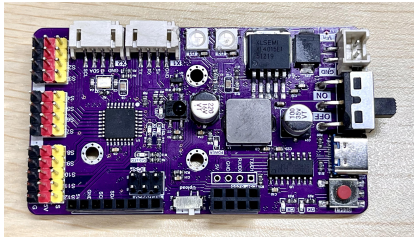


Lesson 7 Battery Level Detection

7.1 Overview

This course focuses on battery level detection, mainly introducing how to use Adeept Pixie Board and Arduino development environment to monitor battery level. Through learning, readers will master the principle of battery voltage acquisition, hardware connection methods, and code writing methods.

7.2 Required Components

Components	Quantity	Picture
Adeept Pixie Board	1	
Type-C USB Cable	1	
OLED Screen	1	

7.3 Principle Introduction

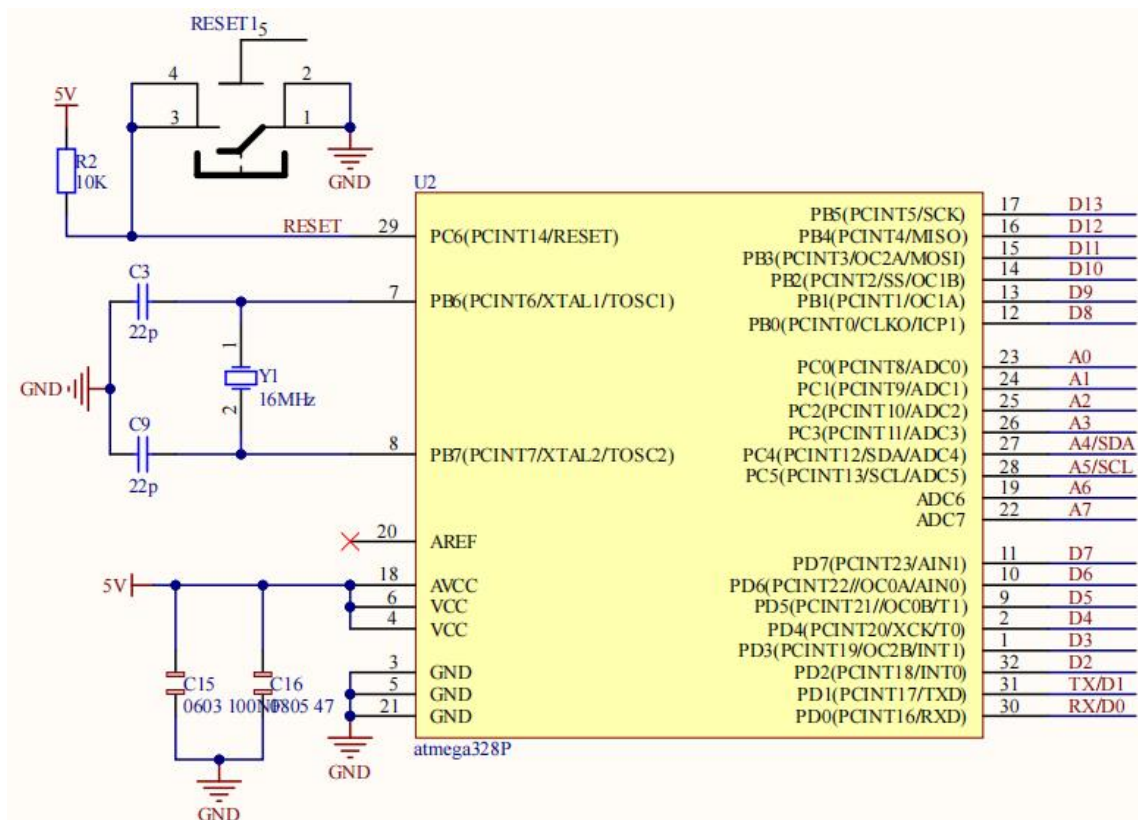
The voltage collection range of pin A7 on the Arduino Pixie board is 0-5V. However, the robot in this experiment is powered by two 18650 lithium batteries, which can reach a voltage of 8.4V

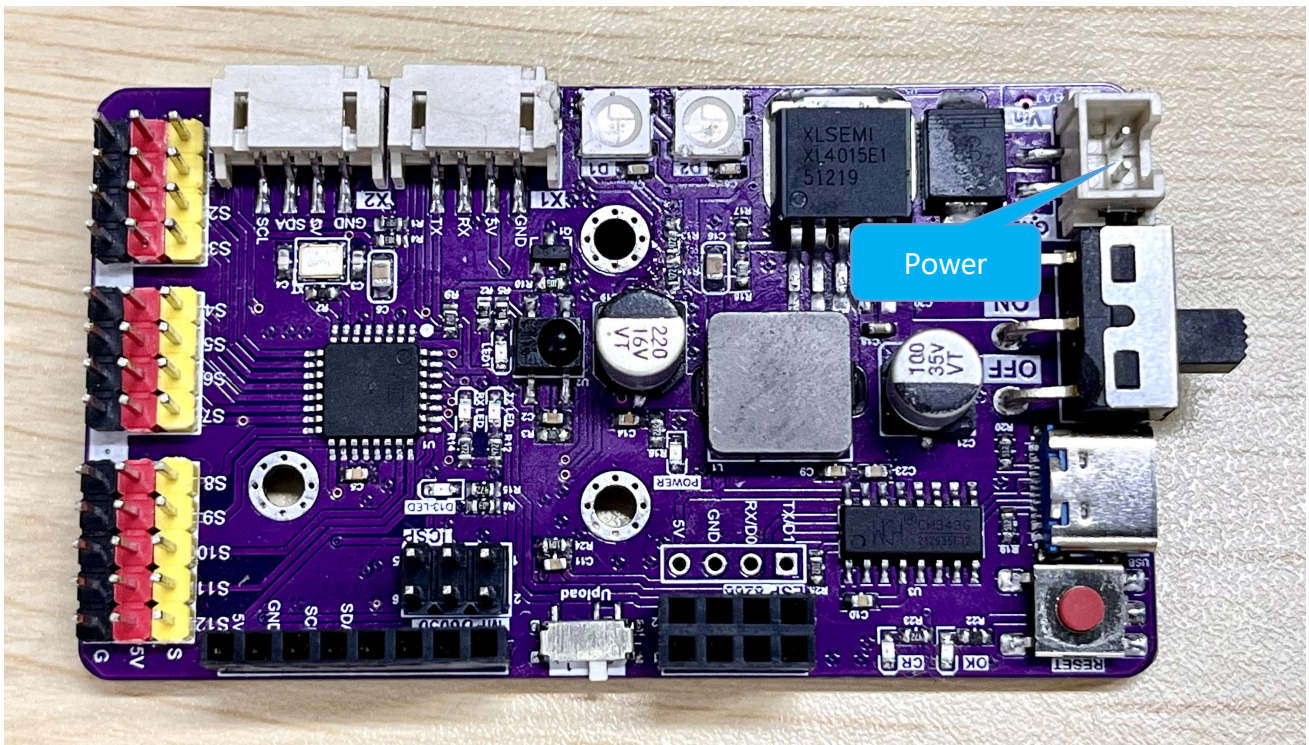
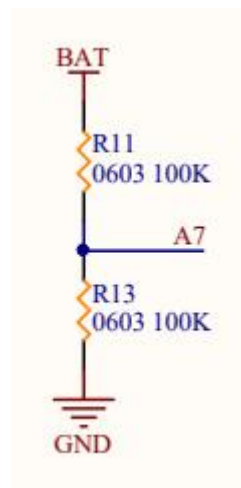
when fully charged—exceeding this collection range. To address this issue, we have introduced two voltage divider resistors with the same resistance value, reducing the maximum voltage measured by pin A7 to 4.2V. Subsequently, the digital value is obtained via the Arduino's ADC (Analog-to-Digital Conversion) function, and the actual battery voltage can be calculated by combining this ADC value with the parameters of the voltage divider resistors for our observation and reference.

The Arduino ADC has a default precision of 10 bits, with a digital value range of 0-1023, corresponding to a reference voltage range of 0-5V. The voltage calculation formula is as follows:

$$\text{Actual Voltage (Vin)} = (\text{ADC Reading} \div 1023) \times \text{Reference Voltage (Vref)} \times \text{Voltage Division Ratio } ((R1+R2)/R2)$$

7.4 Wiring Diagram





7.5 Demonstration

1. Connect your computer and Adept Pixie Board with a USB cable.

2. Open " 07_Battery_Level_Detection" folder in "/Code" ,double-click "07_Battery_Level_Detection.ino" .



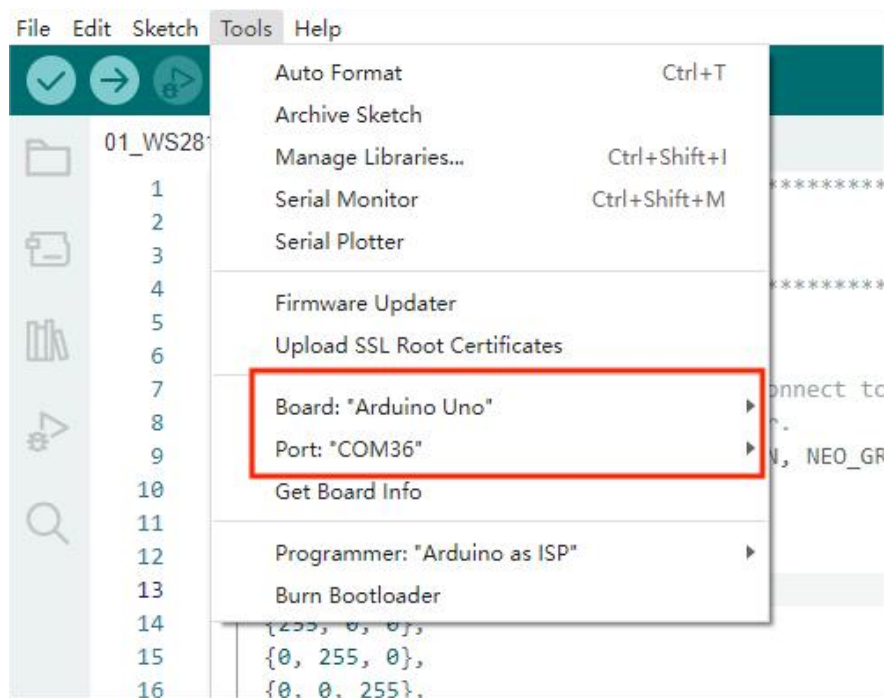
```
07_Battery_Level_Detection | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Arduino Uno
07_Battery_Level_Detection.ino Adeept_Battery_For_Arduino.cpp Adeept_Battery_For_Arduino.h
10
11 variable display
12 Type: class SSD1306AsciiWire
13 SSD1306AsciiWire display
14
15 display.begin(&Adafruit128x64, 0x3C);
16 display.setFont(Adafruit5x7);
17 display.clear();
18 }
19
20 void loop() {
21   display.set1X();
22   display.setCursor(0, 0); // Center display
23   display.print("Bat ADC : ");
24   display.print(Get_Battery_Voltage_ADC()); //Gets the battery ADC value
25   display.setCursor(0, 3); // Center display
26   display.print("Bat Voltage : ");
27   display.print(Get_Battery_Voltage()); //Get the battery voltage value
28   display.print("V");
29   display.setCursor(0, 6); // Center display
30   display.print("Bat Level : ");
31   display.print(Get_Battery_ratio());
32   display.print("%");
33   delay(1000);
34 }
```

3. Select development board and serial port.

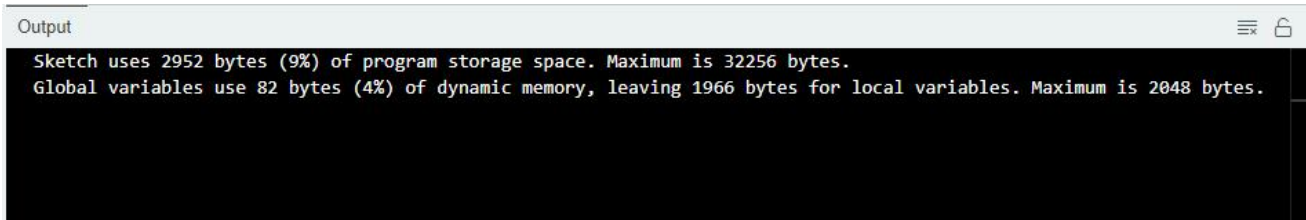
Board: **Tools---**>**Board---**>**Arduino AVR Boards---**>**Arduino Uno**

Port: **Tools --->Port---**>**COMx**

Note: The port number will be different in different computers.



4. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



5. After successfully uploading the code, run the program. At this point, the OLED will display the ADC value of the battery and the calculated battery voltage value in real-time, measured in volts.

7.6 Code

```

01  /*****
02   Description : Read battery voltage.
03   Author      : www.adeept.com
04   *****/
05  #include <Wire.h>
06  #include <SSD1306AsciiWire.h>
07  #include "Adeept_Battery_For_Arduino.h"
08
09  SSD1306AsciiWire display;
10
11  void setup() {

```

```
12 // Initialize OLED
13 Wire.begin();
14 Wire.setClock(400000L);
15 display.begin(&Adafruit128x64, 0x3C);
16 display.setFont(Adafruit5x7);
17 display.clear();
18 }
19
20 void loop() {
21   display.set1X();
22   display.setCursor(0, 0); // Center display
23   display.print("Bat ADC : ");
24   display.print(Get_Battery_Voltage_ADC()); //Gets the battery ADC value
25   display.setCursor(0, 3); // Center display
26   display.print("Bat Voltage : ");
27   display.print(Get_Battery_Voltage()); //Get the battery voltage value
28   display.print("V");
29   display.setCursor(0, 6); // Center display
30   display.print("Bat Level : ");
31   display.print(Get_Battery_ratio());
32   display.print("%");
33   delay(1000);
34 }
```

Code explanation

Initialization (setup function): Executed only once. It starts I2C communication (for connecting the OLED screen), initializes the OLED screen's model, address, and font, and finally clears the screen for display.

Loop Execution (loop function): Runs repeatedly. It displays "Battery ADC Value", "Voltage (unit: V)", and "Power Percentage (unit: %)" at different positions on the OLED screen in sequence, then delays for 1 second before refreshing.