



**CSCE 5612 Embedded Hardware/Software Design**

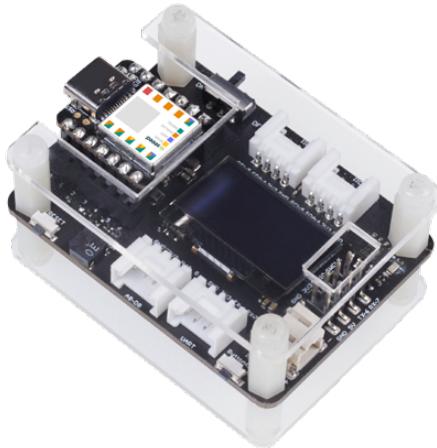
**Spring 2026**

**Lab #1-2: Setting Up the Hardware Kit**

**Due: 1/14/2026 @ 11:59pm**

## 0. Assembling the Components

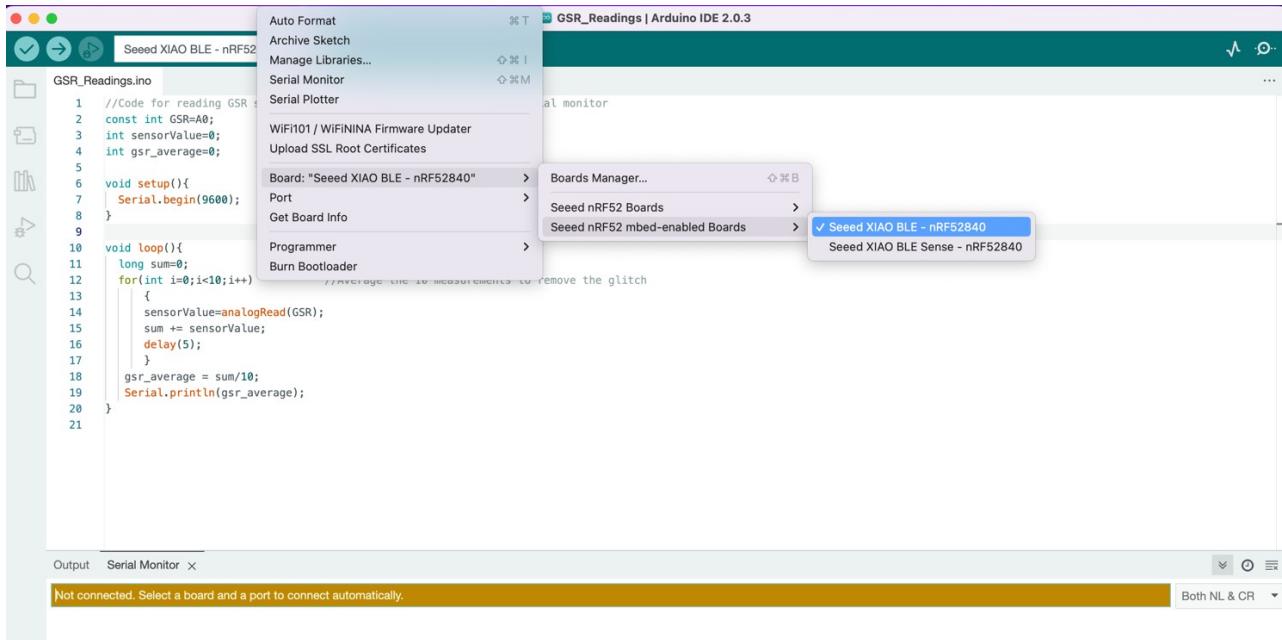
Please watch the [recorded video on YouTube](#) to assemble the components. The below image shows the assembled components.



## 1. IDE and Xiao nRF52840

Please follow the official [Seeed Studio instruction](#) to install Arduino IDE and the required library for Seeed Xiao nRF52840 Sense board. The following six steps will help you to finish the setup:

1. Download and Install the [latest version of Arduino IDE](#) according to your operating system
2. Add Seeed Studio XIAO nRF52840 (Sense) board package to your Arduino IDE
3. Select your board and port according to the tutorial
4. Navigate to File > Examples > 01.Basics > Blink to open Blink example
5. Select the board (Tools > Boards > Seeed nRF52 mbed-enabled Boards > Seeed Xiao BLE – nRF52840 or the other option, which is BLE Sense) according to the following image:



## 6. Click the Upload button to upload the Blink example code to the board

The microcontroller board has a programmable onboard three-in-one LED. You can change the color of the LED to Red, Green, and Blue. The blink example uses the red LED, which can be accessed by calling “LED\_BUILTIN” or “LEDR”. You may replace it with “LED” for green LED and “LEDB” for blue LED, respectively.

## 2. Built-in IMU and Microphone

After successfully installing the IDE and required library for the board, we now move on to sample data from the built-in microphone and IMU.

- IMU: Please follow the provided instructions on [this page](#) to download the required library for the built-in IMU:
  - 1- Download the [Seeed\\_Arduino\\_LSMDS3](#) library to your local machine.
  - 2- Open Arduino IDE, navigate to Sketch > Include Library > Add .ZIP Library... and open the downloaded zip file.
  - 3- Navigate to File > Examples > Seeed Arduino LSM6DS3 > HighLevelExample to open the **HighLevelExample**.
  - 4- Upload the codes and open the **Serial Monitor** (**Tools > Serial Monitor**). The provided example will sample the accelerometer, gyroscope, and temperature sensor from the IMU and print the data into the serial terminal.

- Microphone: Arduino IDE brought a few libraries when we installed the Seeed Xiao Sense board, which can be found in **File > Examples > PDM > PDMSerialPlotter** to open the **PDMSerialPlotter**. Open the Serial Plotter and from "Tools" and you can observe the live microphone stream.

### 3. OLED Display

Please follow through with the [detailed tutorial](#) provided by Seeed Studio to be able to work with the display on the Xiao Shield. These are the required steps:

- 1- Install the [u8g2 library](#), which is required for the display. [This tutorial](#) shows how you can download and install an Arduino library.
- 2- Create a new project, copy the following code, and upload it to your microcontroller.

```
#include <Arduino.h>
#include <U8x8lib.h> //Display library
#include <Wire.h> //I2C protocol library (the display uses I2C to interact with MCU)

U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* clock= */ PIN_WIRE_SCL, /* data= */ PIN_WIRE_SDA, /* reset= */ U8X8_PIN_NONE); //OLEDs without Reset of the Display
void setup(void) {
    u8x8.begin();
    u8x8.setFlipMode(1); // set number from 1 to 3, the screen word will rotary 180
}

void loop(void) {
    u8x8.setFont(u8x8_font_chroma48medium8_r); //try u8x8_font_px437wyse700a_2x2_r
    u8x8.setCursor(0, 0); // It will start printing from (0,0) location
    u8x8.print("Welcome to ");

    u8x8.setCursor(0, 1); // (columns, row)
    u8x8.print("CSCE 5612");

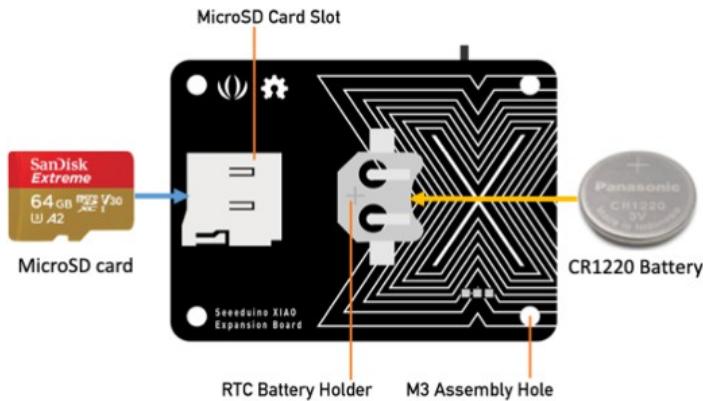
    u8x8.setCursor(0, 2);
    u8x8.print("Class");
}
```

### 4. MicroSD Card

Xiao Shield is equipped with a microSD card slot that can be used for data storage. In this section, we will learn about writing data into external memory (SD card). Arduino IDE already comes with the SD library, the same library we need to work with SD cards. Navigate to **File > Examples > SD >**

Datalogger and open the example. This example continuously stores the data from analog pins 0, 1, and 2 into an SD card. There are a few changes that we need to make in order for the code to work with our setup:

- 1- We use the SPI protocol to interact (read/write) with SD cards. SPI hardware setup includes a chip select line. The SD card chip select is connected to Pin 2 of the microcontroller. Therefore, we need to set the chip select accordingly (**const int chipSelect = 2;**)
- 2- Please place the MicroSD card into the expansion board, similar to the image below
- 3- You can modify the for loop that samples and stores data for three sensors into a single loop that samples and stores data for a single sensor.



## 5. RTC (Real-time clock)

Our expansion board has a built-in RTC chip (PCF8563) with a separate power source (coin cell battery), which can keep time when the device is off or the main battery is completely discharged. RTC is a vital function in embedded solutions, as time-of-event information is essential. We will learn how to set and use the RTC function of our expansion board. Please follow the below steps:

- 1- Download the PCF8563 Arduino library from [here](#).
- 2- Follow [this tutorial](#) (similar to the previous modules) to install the downloaded library to your Arduino IDE.
- 3- We will use the CR1220 battery for RTC, which would serve as a separate power source. Please place the battery into the battery holder, similar to the image above.
- 4- Navigate to **File > Examples > PCF8563 > Simple** and open the example.
- 5- Update the values in the setup function to reflect the current time, then program the microcontroller. For example, if we want to set the time to 01/13/2026, 17:00:00, then we need to change the parameters:

```

pcf.setYear(26);      //set year
pcf.setMonth(1);      //set month
pcf.setDay(13);       //set day
pcf.setHour(17);      //set hour
pcf.setMinut(0);      //set minut
pcf.setSecond(0);     //set second

```

- 6- The above code will set the current time. Now, we need to reprogram the board by removing the lines above from the setup function, since we do not want the microcontroller to set those values every time.

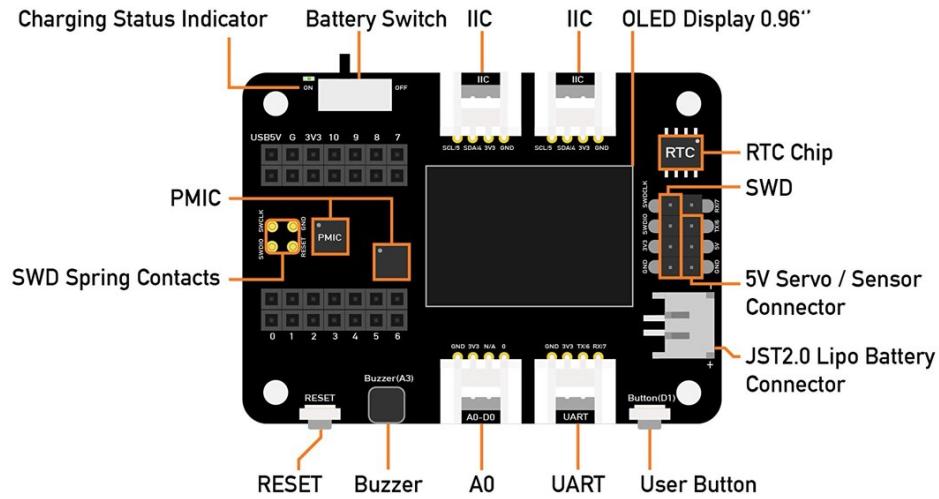
```

#include <PCF8563.h>
PCF8563 pcf;
void setup() {
    Serial.begin(9600);
    pcf.init();           //initialize the clock
    pcf.stopClock();     //stop the clock
    pcf.startClock();    //start the clock
}
void loop() {
    Time nowTime = pcf.getTime(); //get current time
    //print the current time through the serial monitor
    Serial.print(nowTime.day);
    Serial.print("/");
    Serial.print(nowTime.month);
    Serial.print("/");
    Serial.print(nowTime.year);
    Serial.print(" ");
    Serial.print(nowTime.hour);
    Serial.print(":");
    Serial.print(nowTime.minute);
    Serial.print(":");
    Serial.println(nowTime.second);
    delay(1000);
}

```

## 6. Buzzer

This example uses the Buzzer on the Seeed Studio Expansion Base for XIAO to play Happy Birthday!



```
int speakerPin = A3;
int length = 28; // the number of notes
char notes[] = "GGAGcB GGAGdc GGxecBA yyecdc";
int beats[] = { 2, 2, 8, 8, 8, 16, 1, 2, 2, 8, 8, 8, 16, 1, 2, 2, 2, 8, 8, 8, 8, 16, 1,
2, 2, 8, 8, 8, 16 };
int tempo = 150;
void playTone(int tone, int duration) {
    for (long i = 0; i < duration * 1000L; i += tone * 2) {
        digitalWrite(speakerPin, HIGH);
        delayMicroseconds(tone);
        digitalWrite(speakerPin, LOW);
        delayMicroseconds(tone);
    }
}

void playNote(char note, int duration) {
    char names[] = {'C', 'D', 'E', 'F', 'G', 'A', 'B',
                    'c', 'd', 'e', 'f', 'g', 'a', 'b',
                    'x', 'y'
    };
    int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014,
                    956, 834, 765, 593, 468, 346, 224,
                    655 , 715
    };
    int SPEE = 5;

    // play the tone corresponding to the note name
```

```

for (int i = 0; i < 16; i++) {
    if (names[i] == note) {
        int newduration = duration / SPEE;
        playTone(tones[i], newduration);
    }
}
}

void setup() {
    pinMode(speakerPin, OUTPUT);
}

void loop() {
    for (int i = 0; i < length; i++) {
        if (notes[i] == ' ') {
            delay(beats[i] * tempo); // rest
        } else {
            playNote(notes[i], beats[i] * tempo);
        }
        // pause between notes
        delay(tempo);
    }
}

```

## Deliverables:

- Submit the screenshots and photos of each step to your website!