Absolutely. This is a fantastic project that combines biosensing with IoT hardware. Here is a complete guide to get your AD8232 ECG sensor and OLED display working with your ESP32.

We'll start by showing the raw analog data on the screen.

### Hardware & Software Needed

* **Hardware:**
  1. ESP32 Development Board
  2. AD8232 ECG Sensor Module
  3. ECG Electrodes (3-lead)
  4. SSD1306 0.96" I2C OLED Display
  5. Breadboard and Jumper Wires
* **Software:**
  1. Arduino IDE
  2. **Required Libraries (Install via Library Manager):**
     + Adafruit\_GFX
     + Adafruit\_SSD1306

### Step 1: Hardware Connections (Wiring)

This is the most critical part. Follow these connections carefully.

#### 1. AD8232 Sensor to ESP32

The AD8232 is an analog sensor, so its output needs to go to an Analog-to-Digital Converter (ADC) pin on the ESP32. We'll use GPIO 34 for this, as it's a reliable ADC input pin.

| **AD8232 Pin** | **ESP32 Pin** | **Purpose** |
| --- | --- | --- |
| GND | GND | Common Ground |
| 3.3V | 3.3V | Power for Sensor |
| OUTPUT | GPIO 34 | **Analog ECG Signal** |
| LO+ | *(Leave Unconnected)* | Leads-Off Detection (Positive) |
| LO- | *(Leave Unconnected)* | Leads-Off Detection (Negative) |
| SDN | *(Leave Unconnected)* | Shutdown Pin (Module pulls it high) |

#### 2. OLED Display to ESP32

This display uses the I2C communication protocol. The ESP32 has default pins for this.

| **OLED (SSD1306) Pin** | **ESP32 Pin** | **Purpose** |
| --- | --- | --- |
| GND | GND | Common Ground |
| VCC | 3.3V | Power for Display |
| SDA | GPIO 21 | I2C Data Line |
| SCL | GPIO 22 | I2C Clock Line |

### Step 2: The Arduino Code

This code will initialize the OLED display, read the analog value from the AD8232 sensor, and print that raw value to both the Serial Monitor (for debugging) and the OLED screen.

Copy and paste this entire code into your Arduino IDE.

C++

#**include** <Wire.h> // For I2C communication  
#**include** <Adafruit\_GFX.h> // Core graphics library  
#**include** <Adafruit\_SSD1306.h> // Hardware-specific library for SSD1306  
  
// --- Pin Definitions ---  
#**define** ECG\_PIN 34 // The ESP32 pin connected to the AD8232 OUTPUT  
  
// --- OLED Display Setup ---  
#**define** SCREEN\_WIDTH 128 // OLED display width, in pixels  
#**define** SCREEN\_HEIGHT 64 // OLED display height, in pixels  
#**define** OLED\_RESET -1 // Reset pin # (or -1 if sharing ESP32 reset pin)  
  
// Initialize the OLED display (I2C)  
// Default I2C pins for ESP32 are SDA: 21, SCL: 22  
Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);  
  
void setup() {  
 // Start serial for debugging (optional, but recommended)  
 Serial.begin(115200);  
  
 // Initialize OLED display  
 if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x64  
 Serial.println(F("SSD1306 allocation failed"));  
 for(;;); // Don't proceed, loop forever if display fails  
 }  
  
 // Clear the display buffer  
 display.clearDisplay();  
   
 // Display a startup message  
 display.setTextSize(1);  
 display.setTextColor(SSD1306\_WHITE);  
 display.setCursor(0, 0);  
 display.println("AD8232 ECG Monitor");  
 display.println("Initializing...");  
 display.display(); // Send buffer to the screen  
 delay(2000);  
}  
  
void loop() {  
 // Read the analog value from the AD8232  
 // The ESP32 ADC is 12-bit, so the value will be between 0 and 4095  
 int ecgValue = analogRead(ECG\_PIN);  
  
 // --- 1. Print to Serial Monitor ---  
 // You can use the Arduino IDE's Serial Plotter (Tools > Serial Plotter)  
 // to see the waveform! This is great for debugging.  
 Serial.println(ecgValue);  
  
 // --- 2. Display on OLED ---  
 display.clearDisplay(); // Clear the display buffer  
   
 // Set text size and color  
 display.setTextSize(1);  
 display.setTextColor(SSD1306\_WHITE);  
   
 // Display a static label  
 display.setCursor(0, 0);  
 display.print("ECG Signal Value:");  
   
 // Display the live ECG value in a larger font  
 display.setTextSize(2);  
 display.setCursor(0, 20);  
 display.print(ecgValue);  
   
 // Send the completed buffer to the screen  
 display.display();   
  
 // Small delay to make the screen readable.  
 // For a faster waveform, you would lower this.  
 delay(50);   
}

### Step 3: How It Works (Code Explanation)

1. **Includes & Definitions:** We include the Wire library for I2C and the two Adafruit libraries for the display. We #define our pins to make the code easy to read.
2. **Adafruit\_SSD1306 display(...):** This line creates an "object" that represents our physical screen.
3. **setup():**
   * Serial.begin(115200): Starts the serial monitor for debugging.
   * display.begin(...): Initializes the OLED screen. The if statement checks to make sure it was found.
   * The display.print() and display.display() commands show a one-time startup message.
4. **loop():**
   * **int ecgValue = analogRead(ECG\_PIN);**: This is the core command. It reads the voltage from GPIO 34 and converts it to a number between 0 (for 0V) and 4095 (for 3.3V).
   * **Serial.println(ecgValue);**: This prints the value to your computer. **Pro-Tip:** Open **Tools > Serial Plotter** in the Arduino IDE. You will see a live graph of your ECG signal!
   * **display.clearDisplay();**: This is essential. It wipes the screen blank before drawing the new frame.
   * **display.setCursor(x, y);**: Sets where the "pen" will start writing.
   * **display.print(ecgValue);**: Writes the number we just read.
   * **display.display();**: This command "publishes" everything you've drawn in the buffer to the actual screen.

### ⚠️ Important Medical Disclaimer

**This project is for educational and prototyping purposes ONLY.** The AD8232 sensor and this setup are **NOT** medical-grade devices. Do not use this for any medical diagnosis, treatment, or life-support applications.

### Next Steps

Right now, this code just shows a number. Your next step will be to plot the actual **waveform** on the OLED. This is more complex, as it involves storing past readings in an array and drawing lines (display.drawLine()) between them to create a scrolling graph.

Would you like me to help you modify this code to plot a basic, real-time waveform on the OLED display?

This video provides a great overview of using the AD8232 sensor, which can be very helpful for seeing the expected signal.

ECG Monitoring with AD8232 Sensor