

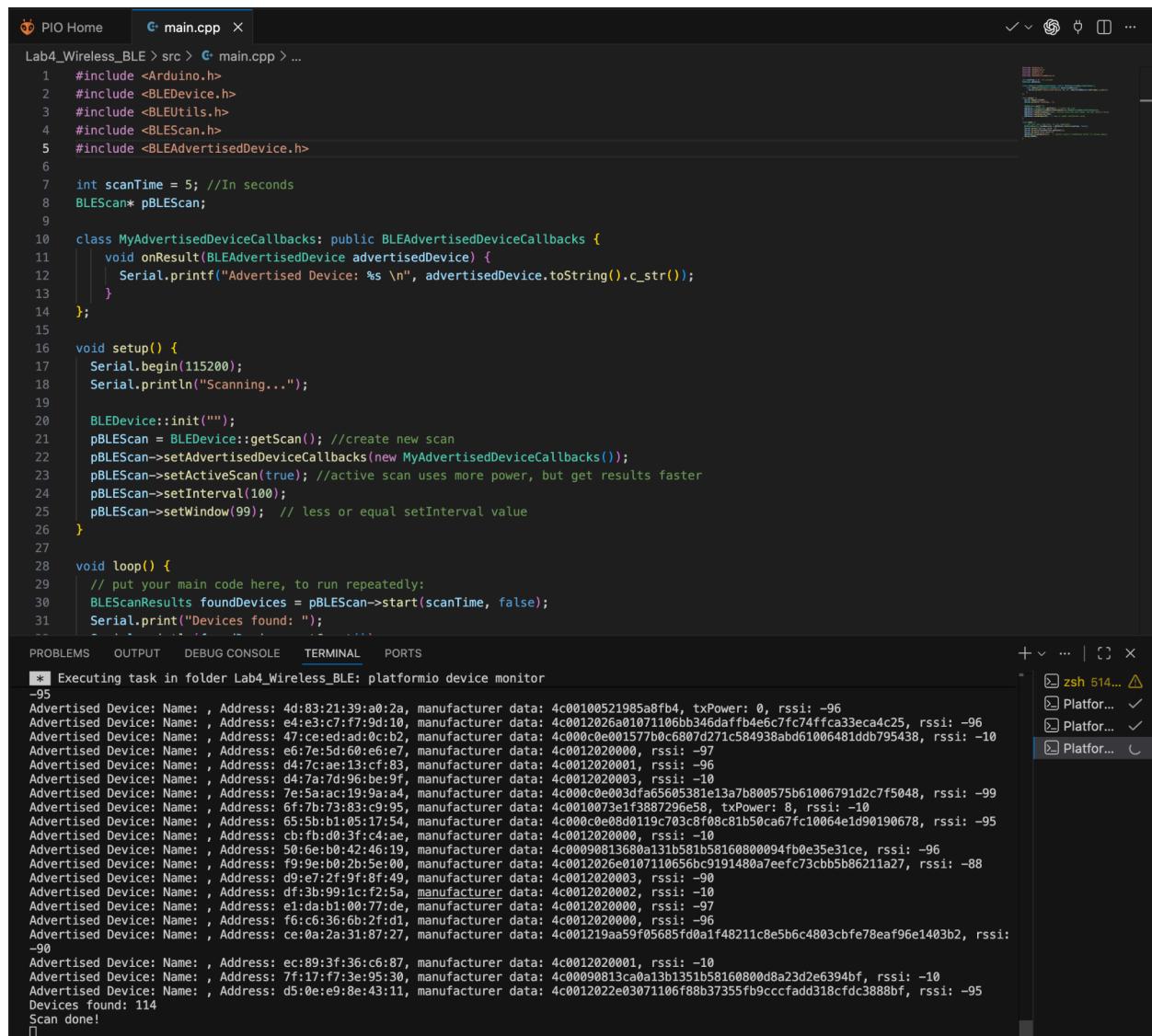
# Lab 4: Wireless

Lewis Liu, Iris Yang

Github Link: <https://github.com/IrisYzh/TECHIN-514-Hardware-Software-Lab>

## Section 1: MCU as BLEScanner

Screenshot of your serial monitor displaying the number of Bluetooth devices detected using your MCU as BLEScanner



```
* Executing task in folder Lab4_Wireless_BLE: platformio device monitor
-95
Advertised Device: Name: , Address: 4d:83:21:39:a0:2a, manufacturer data: 4c00100521985a8fb4, txPower: 0, rssi: -96
Advertised Device: Name: , Address: e4:e3:c7:f9:d1:00, manufacturer data: 4c0012026a01071196bb346daaffb4e6c7fc74ffca33eca4c25, rssi: -96
Advertised Device: Name: , Address: 47:ce:ed:ad:0c:b2, manufacturer data: 4c000c0e001577b0c6807d271c584938abd61006481ddb795438, rssi: -10
Advertised Device: Name: , Address: e6:7e:5d:60:e6:e7, manufacturer data: 4c0012020000, rssi: -97
Advertised Device: Name: , Address: d4:7cae:13:cf:83, manufacturer data: 4c0012020001, rssi: -96
Advertised Device: Name: , Address: d4:7a:7d:96:be:9f, manufacturer data: 4c0012020003, rssi: -10
Advertised Device: Name: , Address: 7e:5a:ac:19:9a:a4, manufacturer data: 4c000c0e003dfa5605381e13a7b800575b61006791d2c7f5048, rssi: -99
Advertised Device: Name: , Address: 6f:7b:73:83:c9:95, manufacturer data: 4c0010073e1f3887296e58, txPower: 8, rssi: -10
Advertised Device: Name: , Address: 65:5b:b1:05:17:54, manufacturer data: 4c000c0e008d019c703c8f08c81b50c467fc10064e1d90190678, rssi: -95
Advertised Device: Name: , Address: cb:fb:d0:3f:c4:ae, manufacturer data: 4c0012020000, rssi: -10
Advertised Device: Name: , Address: 50:6e:b0:42:46:19, manufacturer data: 4c00090813680a131b581b58160000947fb0e35e31ce, rssi: -96
Advertised Device: Name: , Address: f9:9e:b0:2b:5e:00, manufacturer data: 4c0012026e0107110656bc9191480a7eefc73ccb5b86211a27, rssi: -88
Advertised Device: Name: , Address: d9:e7:2f:9f:8f:49, manufacturer data: 4c0012020003, rssi: -90
Advertised Device: Name: , Address: df:3b:99:1c:f2:5a, manufacturer data: 4c0012020002, rssi: -10
Advertised Device: Name: , Address: el:da:b1:00:77:de, manufacturer data: 4c0012020000, rssi: -97
Advertised Device: Name: , Address: f6:c6:36:6b:2f:df, manufacturer data: 4c0012020000, rssi: -96
Advertised Device: Name: , Address: ce:0a:2a:31:87:27, manufacturer data: 4c001219aa5f05685fd0a1f48211c8e5b6c4803cbfe78eaf96e1403b2, rssi: -90
Advertised Device: Name: , Address: ec:89:3f:36:c6:87, manufacturer data: 4c0012020001, rssi: -10
Advertised Device: Name: , Address: 7f:17:f7:3e:95:30, manufacturer data: 4c00090813ca0a13b1351b581600008d8a23d2e6394bf, rssi: -10
Advertised Device: Name: , Address: d5:0e:e9:8e:43:11, manufacturer data: 4c0012022e03071106f88b37355fb9ccfad318cfcd3888bf, rssi: -95
Devices found: 114
Scan done!
```

## Section 2: Establishing Bluetooth Connection

Screenshot of the serial monitor of your client device to show successfull connection with your server device (make sure the server device's name is included).

Server

The screenshot shows the PlatformIO IDE interface. The left sidebar displays project files under 'XIAO-BLUETOOTH-SERVER'. The main editor window shows the 'main.cpp' file with code for a BLE server. The code includes defines for service and characteristic UUIDs, and a callback function for handling connections. The serial monitor at the bottom shows a continuous stream of log messages from the device, all of which read 'Hello World', indicating successful communication.

```
#include <Arduino.h>
#include <BLEDevice.h>
#include <BLEServer.h>
#include <BLEUtils.h>
#include <BLE2902.h>
#include <stdlib.h>

BLEServer* pServer = NULL;
BLECharacteristic* pCharacteristic = NULL;

bool deviceConnected = false;
bool oldDeviceConnected = false;

unsigned long previousMillis = 0;
const long interval = 1000;

// Print device name periodically so it is guaranteed to appear in your screenshot
unsigned long namePrintillis = 0;
const long namePrintinterval = 5000;

// Server device name (will show in Serial Monitor)
static const char* SERVER_NAME = "BLE_SERVER";

ImplementWithCode
// TODO: Change the UUID to your own if needed
#define SERVICE_UUID      "724fcfe5-485e-467c-a7b9-ef2796515386"
#define CHARACTERISTIC_UUID "976e3398-600d-4d49-ac5d-95383f1c14da"

class MyServerCallbacks : public BLEServerCallbacks {
public:
    void onConnect(BLEServer* pServer) override {
        deviceConnected = true;
    }
};

// Select successful connection or choose capital letters
BLEServerCallbacks callbacks;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Executing task: platformio device monitor

16:07:30.352 > Notify value (BLE\_SERVER): Hello World  
16:07:30.352 > Notify value (BLE\_SERVER): Hello World  
16:07:32.359 > Notify value (BLE\_SERVER): Hello World  
16:07:32.739 > Server Device Name: BLE\_SERVER  
16:07:34.363 > Notify value (BLE\_SERVER): Hello World  
16:07:34.363 > Notify value (BLE\_SERVER): Hello World  
16:07:35.372 > Notify value (BLE\_SERVER): Hello World  
16:07:36.373 > Notify value (BLE\_SERVER): Hello World  
16:07:36.373 > Notify value (BLE\_SERVER): Hello World  
16:07:37.739 > Server Device Name: BLE SERVER  
16:07:38.388 > Notify value (BLE\_SERVER): Hello World  
16:07:38.388 > Notify value (BLE\_SERVER): Hello World  
16:07:39.405 > Notify value (BLE\_SERVER): Hello World  
16:07:41.393 > Notify value (BLE\_SERVER): Hello World  
16:07:42.394 > Notify value (BLE\_SERVER): Hello World  
16:07:43.400 > Notify value (BLE\_SERVER): Hello World  
16:07:43.400 > Notify value (BLE\_SERVER): Hello World  
16:07:44.401 > Notify value (BLE\_SERVER): Hello World  
16:07:45.407 > Notify value (BLE\_SERVER): Hello World  
16:07:46.414 > Notify value (BLE\_SERVER): Hello World  
16:07:47.754 > Server Device Name: BLE\_SERVER  
16:07:48.419 > Notify value (BLE\_SERVER): Hello World  
16:07:49.421 > Notify value (BLE\_SERVER): Hello World

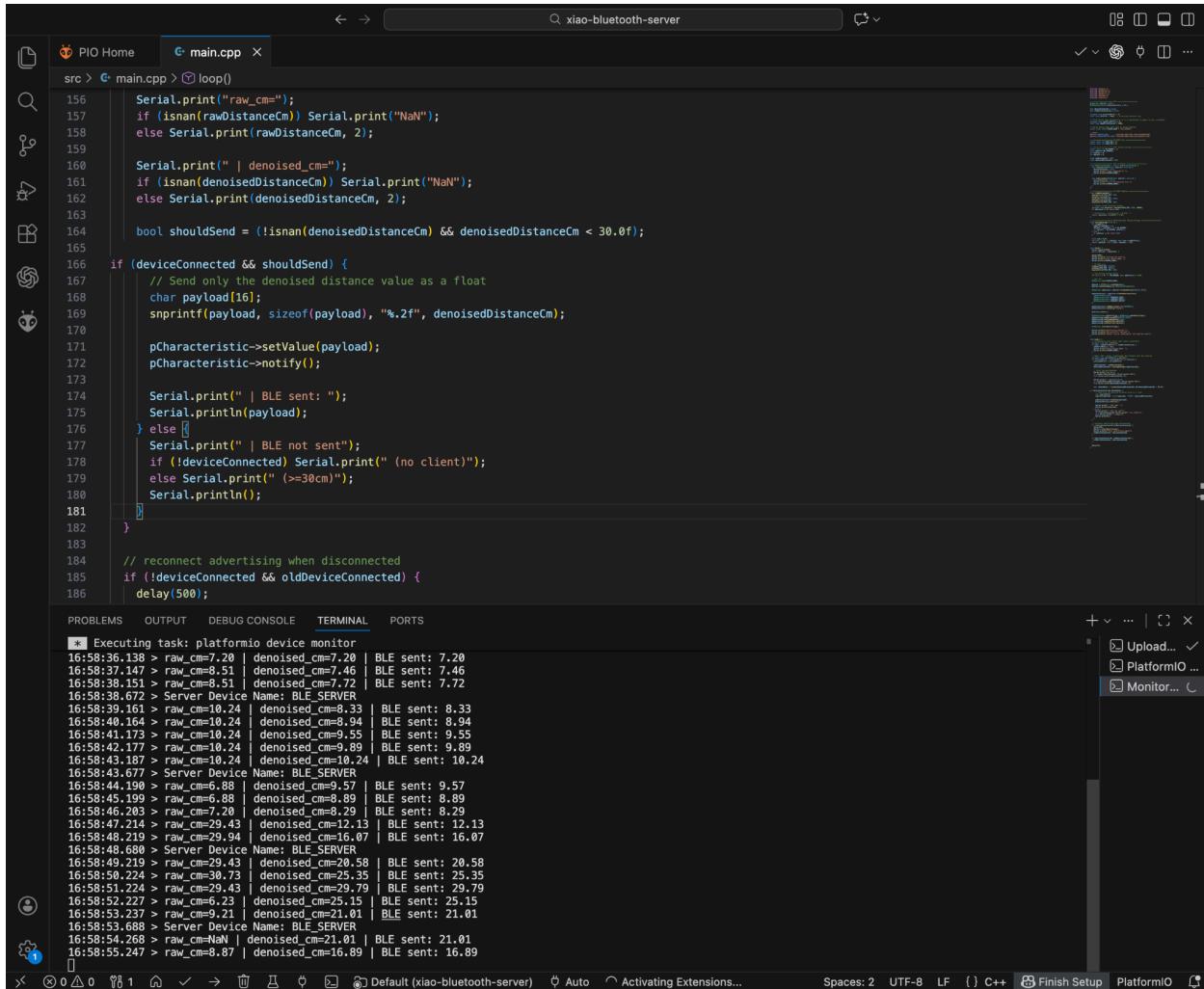
## Client

The screenshot shows the PlatformIO IDE interface with the following details:

- Project Explorer:** Shows the project structure for "XIAO-BLUETOOTH-CLIENT" with files like .pio, .vscode, include, lib, src/main.cpp, .gitignore, and platformio.ini.
- Code Editor:** The main.cpp file is open, showing C++ code for a BLE client. Key parts include:
  - #include <Arduino.h>
  - #include <BLEDevice.h>
  - #include <BLEUtils.h>
  - #include <BLEScan.h>
  - #include <BLEAdvertisedDevice.h>
  - static BLEUUID serviceUUID("724fc8e5-485e-467c-a7b9-ef2796515386");
  - static BLEUUID charUUID("976e3398-600d-4d49-ac5d-95383f1c14da");
  - static boolean doConnect = false;
  - static boolean connected = false;
  - static boolean doScan = false;
  - static BLERemoteCharacteristic\* pRemoteCharacteristic = nullptr;
  - static BLEAdvertisedDevice\* myDevice = nullptr;
  - // NEW: store server name for screenshot
  - static String serverName = "Unknown";
  - static void notifyCallback(BLERemoteCharacteristic\* pBLERemoteCharacteristic, uint8\_t\* pData, size\_t length, bool isNotify) {
    - Serial.print("Notify from ");
    - Serial.print(serverName);
- Terminal:** Displays the output of the "platformio.exe device monitor" command, showing the device seed and communication logs between the client and a BLE server.
- Build Area:** A sidebar on the right titled "Build with Agent" includes options like "Upload...", "Build Task", "Upload ...", "Build (se...)", and "Monitor...".
- Status Bar:** Shows the current file (main.cpp), line (Ln 18, Col 37), and encoding (UTF-8).

## 4.3 Sensor Device Signal Processing and Data Transmission

Screenshot of the serial monitor of your server device to show the raw and denoised sensor data.



The screenshot shows the PlatformIO IDE interface with the serial monitor open. The code in main.cpp is displayed in the editor, showing the logic for printing raw and denoised distance values to the serial port. The serial monitor window shows the output of the code, displaying raw\_cm and denoised\_cm values along with BLE sent values over time.

```
src > main.cpp > loop()
156     Serial.print("raw_cm=");
157     if (isnan(rawDistanceCm)) Serial.print("NaN");
158     else Serial.print(rawDistanceCm, 2);
159
160     Serial.print(" | denoised_cm=");
161     if (isnan(denoisedDistanceCm)) Serial.print("NaN");
162     else Serial.print(denoisedDistanceCm, 2);
163
164     bool shouldSend = (!isnan(denoisedDistanceCm) && denoisedDistanceCm < 30.0f);
165
166     if (deviceConnected && shouldSend) {
167         // Send only the denoised distance value as a float
168         char payload[16];
169         sprintf(payload, sizeof(payload), "%2f", denoisedDistanceCm);
170
171         pCharacteristic->setValue(payload);
172         pCharacteristic->notify();
173
174         Serial.print(" | BLE sent: ");
175         Serial.println(payload);
176     } else {
177         Serial.print(" | BLE not sent");
178         if (!deviceConnected) Serial.print(" (no client)");
179         else Serial.print(" (>=30cm)");
180     }
181 }
182
183
184 // reconnect advertising when disconnected
185 if (!deviceConnected && oldDeviceConnected) {
186     delay(500);
187 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

\* Executing task: platformio device monitor

```
16:58:36.138 > raw_cm=7.20 | denoised_cm=7.20 | BLE sent: 7.20
16:58:37.147 > raw_cm=8.51 | denoised_cm=7.46 | BLE sent: 7.46
16:58:38.151 > raw_cm=8.51 | denoised_cm=7.72 | BLE sent: 7.72
16:58:38.668 > Server Device Name: BLE SERVER
16:58:39.161 > raw_cm=10.24 | denoised_cm=9.33 | BLE sent: 8.33
16:58:40.164 > raw_cm=10.24 | denoised_cm=8.94 | BLE sent: 8.94
16:58:41.173 > raw_cm=10.24 | denoised_cm=9.55 | BLE sent: 9.55
16:58:42.177 > raw_cm=10.24 | denoised_cm=9.89 | BLE sent: 9.89
16:58:43.187 > raw_cm=10.24 | denoised_cm=10.24 | BLE sent: 10.24
16:58:43.677 > Server Device Name: BLE SERVER
16:58:44.198 > raw_cm=6.88 | denoised_cm=9.57 | BLE sent: 9.57
16:58:45.199 > raw_cm=6.88 | denoised_cm=8.89 | BLE sent: 8.89
16:58:46.203 > raw_cm=7.20 | denoised_cm=8.29 | BLE sent: 8.29
16:58:47.214 > raw_cm=29.43 | denoised_cm=12.13 | BLE sent: 12.13
16:58:48.219 > raw_cm=29.94 | denoised_cm=16.07 | BLE sent: 16.07
16:58:48.686 > Server Device Name: BLE SERVER
16:58:49.219 > raw_cm=29.43 | denoised_cm=20.58 | BLE sent: 20.58
16:58:50.224 > raw_cm=30.73 | denoised_cm=25.35 | BLE sent: 25.35
16:58:51.224 > raw_cm=29.43 | denoised_cm=29.79 | BLE sent: 29.79
16:58:52.227 > raw_cm=6.23 | denoised_cm=25.15 | BLE sent: 25.15
16:58:53.237 > raw_cm=9.21 | denoised_cm=21.01 | BLE sent: 21.01
16:58:53.686 > Server Device Name: BLE SERVER
16:58:54.268 > raw_cm=NaN | denoised_cm=21.01 | BLE sent: 21.01
16:58:55.247 > raw_cm=8.87 | denoised_cm=16.89 | BLE sent: 16.89
```

## 4.4 Display Device Signal Processing and Data Receiving

Screenshot of the serial monitor of your client device to show the currently-reading, maximum, and minimum data transmitted from your server device.

The screenshot shows two instances of the PlatformIO Serial Monitor interface. Both instances are connected to the same project, "XIAO-BLUETOOTH-CLIENT", which includes files like .pio, .vscode, include, lib, and src/main.cpp. The main.cpp file contains code for handling BLE connections and distance data. The serial monitor displays multiple messages from a BLE\_SERVER, each containing raw data and calculated current, maximum, and minimum distances in centimeters. The data is received sequentially, with message IDs ranging from 124 to 159. The interface includes a Chat window, build options (Upload, Build Task, etc.), and an AI integration feature called "Build with Agent".

```
=====
Data #124 received from BLE_SERVER
Raw data: 18.23
-----
Current Distance: 18.23 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.27 cm
=====

=====
Data #125 received from BLE_SERVER
Raw data: 18.68
-----
Current Distance: 18.68 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.27 cm
=====

=====
Data #126 received from BLE_SERVER
Raw data: 21.46
-----
Current Distance: 21.46 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.27 cm
=====

=====
Data #127 received from BLE_SERVER
Raw data: 21.85
-----
Current Distance: 21.85 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.27 cm
=====

=====
Data #128 received from BLE_SERVER
Raw data: 24.16
-----
Current Distance: 24.16 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.27 cm
=====

=====
Data #129 received from BLE_SERVER
Raw data: 24.14
-----
Current Distance: 24.14 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.27 cm
=====

=====
Data #155 received from BLE_SERVER
Raw data: 12.77
-----
Current Distance: 12.77 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.06 cm
=====

=====
Data #156 received from BLE_SERVER
Raw data: 13.52
-----
Current Distance: 13.52 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.06 cm
=====

=====
Data #157 received from BLE_SERVER
Raw data: 17.32
-----
Current Distance: 17.32 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.06 cm
=====

=====
Data #158 received from BLE_SERVER
Raw data: 20.70
-----
Current Distance: 20.70 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.06 cm
=====

=====
Data #159 received from BLE_SERVER
Raw data: 24.14
-----
Current Distance: 24.14 cm
Maximum Distance: 29.55 cm
Minimum Distance: 11.06 cm
=====
```

The screenshot shows the PlatformIO IDE interface. The left sidebar displays the project structure for 'XIAO-BLUETOOTH-CLIENT' with files like .pio, .vscode, lib, and src. The main area shows serial output from the terminal tab:

```
Data #180 received from BLE_SERVER  
Raw data: 25.75  
-----  
Current Distance: 25.75 cm  
Maximum Distance: 29.94 cm  
Minimum Distance: 11.06 cm  
-----  
  
Data #181 received from BLE_SERVER  
Raw data: 28.42  
-----  
Current Distance: 28.42 cm  
Maximum Distance: 29.94 cm  
Minimum Distance: 11.06 cm  
-----  
  
Data #182 received from BLE_SERVER  
Raw data: 28.30  
-----  
Current Distance: 28.30 cm  
Maximum Distance: 29.94 cm  
Minimum Distance: 11.06 cm  
-----  
  
Data #183 received from BLE_SERVER  
Raw data: 25.34  
-----  
Current Distance: 25.34 cm  
Maximum Distance: 29.94 cm  
Minimum Distance: 11.06 cm  
-----  
  
Data #184 received from BLE_SERVER  
Raw data: 20.71  
-----  
Current Distance: 20.71 cm  
Maximum Distance: 29.94 cm  
Minimum Distance: 11.06 cm  
-----
```

The right sidebar contains build-related options: Upload..., Build Task, Upload ..., Build (se...), and Monitor... (disabled). A 'Build with Agent' feature is shown with a message: 'AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.' Below the terminal are status indicators for the current file (main.cpp), build settings (Agent v Auto, Spaces v 2, LF, C++), and environment (envseed xiao esp32c3 (xao-bluetooth-client)).