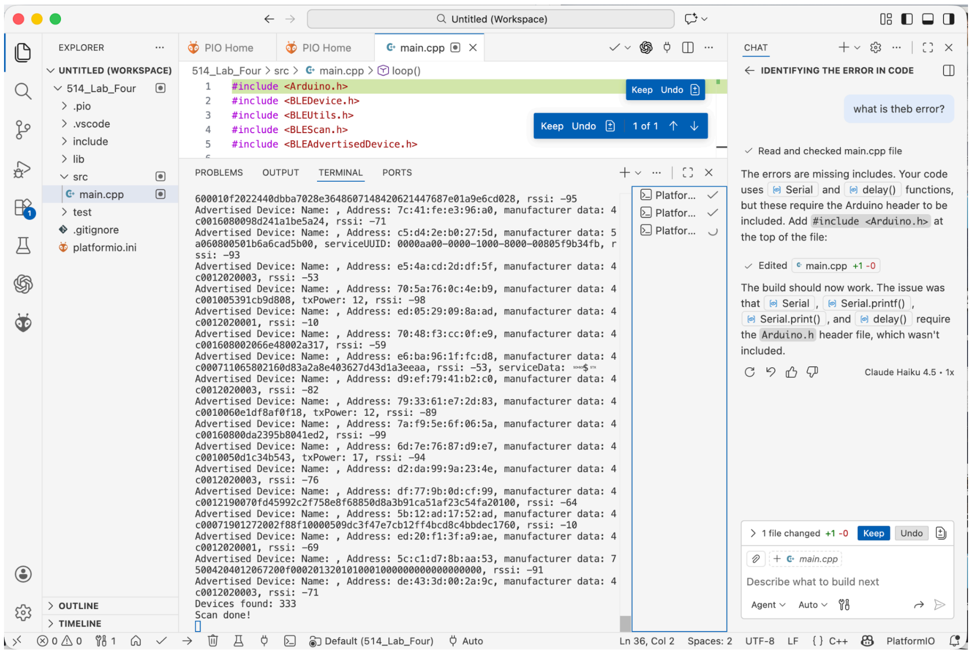
Team Member: Kitty Li (server), Peizhong Gao (client)

1. **The link to your edited code is on GitHub. You can either create a new repo or push the updates to your existing repo for all labs (recommended)**

   Kitty Li:
   https://github.com/xinyili7/514_Lab_Four_Bluetooth_Service.git

   Peizhong Gao:
   https://github.com/GPZ12138/TECHIN514_labs/tree/main/lab4/xiao-bluetooth-client

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



```
#include <Arduino.h>
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEScan.h>
#include <BLEAdvertisedDevice.h>
```

```
600010f2022440dbba7028e36486071484206214476 87e01a9e6cd028, rssi: -95
Advertised Device: Name: , Address: 7c:41:fe:e3:96:a0, manufacturer data: 4
c0016080098d241a1be5a24, rssi: -71
Advertised Device: Name: , Address: c5:d4:2e:b0:27:5d, manufacturer data: 5
a060800501b6a6cad5b00, serviceUUID: 0000aa00-0000-1000-8000-00805f9b34fb, r
ssi: -93
Advertised Device: Name: , Address: e5:4a:cd:2d:df:5f, manufacturer data: 4
c0012020003, rssi: -53
Advertised Device: Name: , Address: 70:5a:76:0c:4e:b9, manufacturer data: 4
c001005391cb9d808, txPower: 12, rssi: -98
Advertised Device: Name: , Address: ed:05:29:09:8a:ad, manufacturer data: 4
c0012020001, rssi: -10
Advertised Device: Name: , Address: 70:48:f3:cc:0f:e9, manufacturer data: 4
c001608002066e48002a317, rssi: -59
Advertised Device: Name: , Address: e6:ba:96:1f:fc:d8, manufacturer data: 4
c000711065802160d83a2a8e403627d43d1a3eeaa, rssi: -53, serviceData: $
Advertised Device: Name: , Address: d9:ef:79:41:b2:c0, manufacturer data: 4
c0012020003, rssi: -82
Advertised Device: Name: , Address: 79:33:61:e7:2d:83, manufacturer data: 4
c0010060e1df8af0f18, txPower: 12, rssi: -89
Advertised Device: Name: , Address: 7a:f9:5e:6f:06:5a, manufacturer data: 4
c00160800da2395b8041ed2, rssi: -99
Advertised Device: Name: , Address: 6d:7e:76:87:d9:e7, manufacturer data: 4
c0010050d1c34b543, txPower: 17, rssi: -94
Advertised Device: Name: , Address: d2:da:99:9a:23:4e, manufacturer data: 4
c0012020003, rssi: -76
Advertised Device: Name: , Address: df:77:9b:0d:cf:99, manufacturer data: 4
c0012190070fd45992c2f758e8f68850d8a3b91ca51af23c54fa20100, rssi: -64
Advertised Device: Name: , Address: 5b:12:ad:17:52:ad, manufacturer data: 4
c00071901272002f88f10000509dc3f47e7cb12ff4bcd8c4bbdec1760, rssi: -10
Advertised Device: Name: , Address: ed:20:f1:3f:a9:ae, manufacturer data: 4
c0012020001, rssi: -69
Advertised Device: Name: , Address: 5c:c1:d7:8b:aa:53, manufacturer data: 7
5004204012067200f000201320101000100000000000000000, rssi: -91
Advertised Device: Name: , Address: de:43:3d:00:2a:9c, manufacturer data: 4
c0012020003, rssi: -71
Devices found: 333
Scan done!
```

what is theb error?

✓ Read and checked main.cpp file

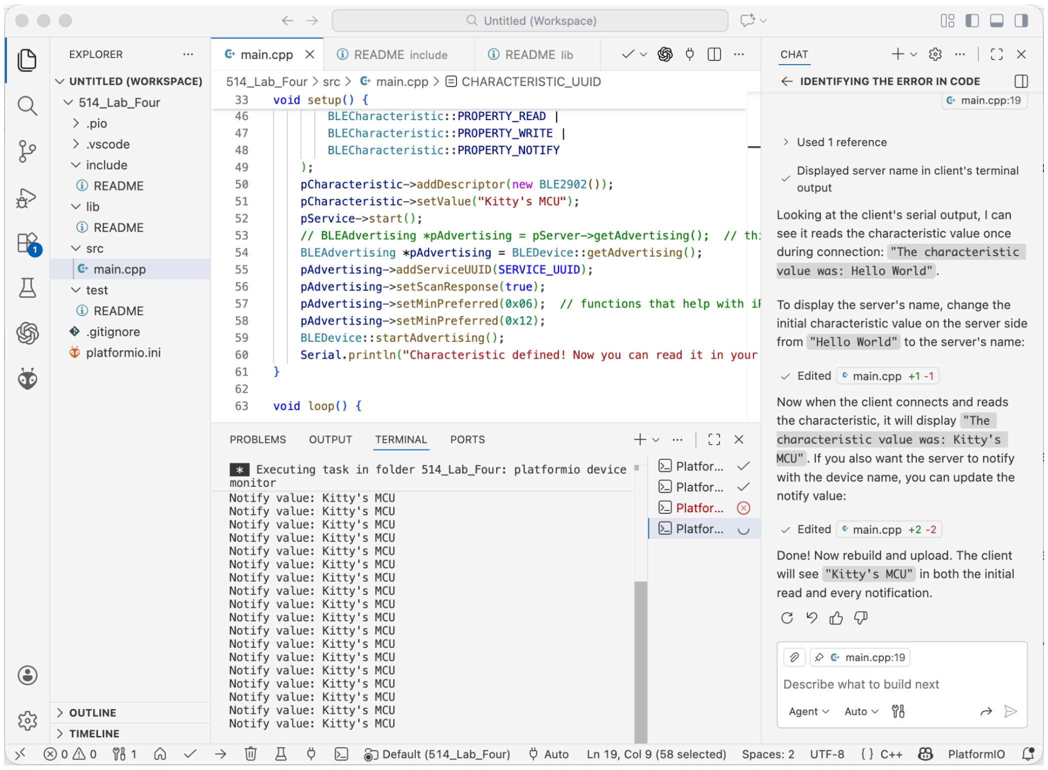
The errors are missing includes. Your code uses Serial and delay() functions, but these require the Arduino header to be included. Add `#include <Arduino.h>` at the top of the file:

✓ Edited main.cpp +1 -0

The build should now work. The issue was that Serial, Serial.printf(), Serial.print(), and delay() require the `Arduino.h` header file, which wasn't included.

Claude Haiku 4.5 • 1x

3. Screenshot of the serial monitor of your client device to show successful connection with your server device



```
void setup() {
        BLECharacteristic::PROPERTY_READ |
        BLECharacteristic::PROPERTY_WRITE |
        BLECharacteristic::PROPERTY_NOTIFY
    );
    pCharacteristic->addDescriptor(new BLE2902());
    pCharacteristic->setValue("Kitty's MCU");
    pService->start();
    // BLEAdvertising *pAdvertising = pServer->getAdvertising();  // th
    BLEAdvertising *pAdvertising = BLEDevice::getAdvertising();
    pAdvertising->addServiceUUID(SERVICE_UUID);
    pAdvertising->setScanResponse(true);
    pAdvertising->setMinPreferred(0x06);  // functions that help with i
    pAdvertising->setMinPreferred(0x12);
    BLEDevice::startAdvertising();
    Serial.println("Characteristic defined! Now you can read it in your
}

void loop() {
```

```
 *  Executing task in folder 514_Lab_Four: platformio device
monitor
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
Notify value: Kitty's MCU
```

CHAT — IDENTIFYING THE ERROR IN CODE

Used 1 reference

Displayed server name in client's terminal output

Looking at the client's serial output, I can see it reads the characteristic value once during connection: `"The characteristic value was: Hello World"`.

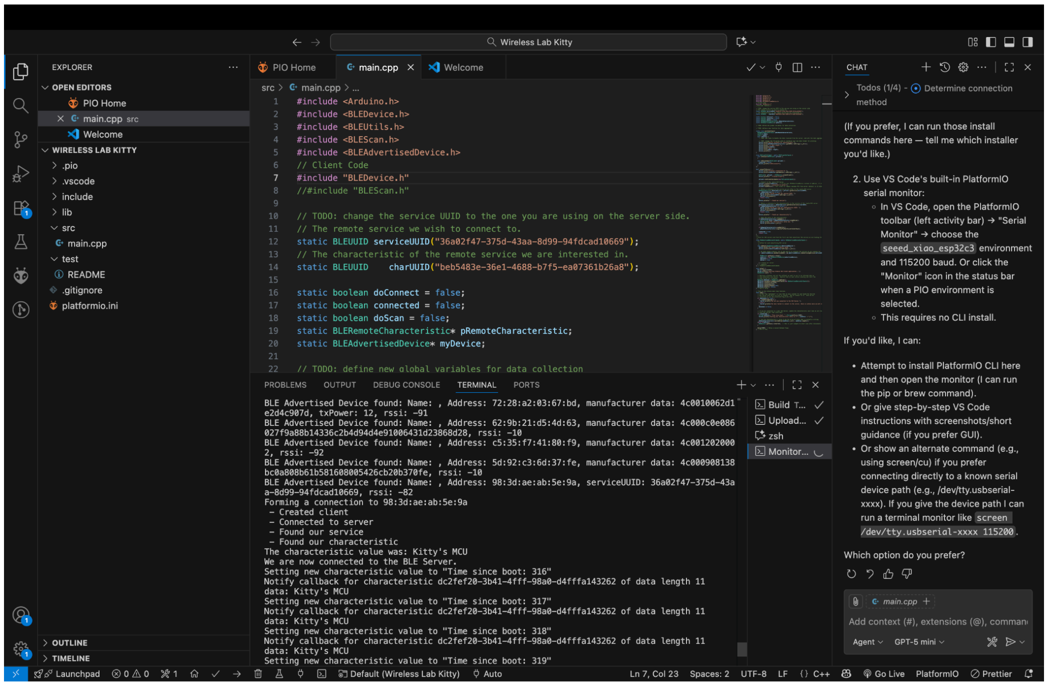
To display the server's name, change the initial characteristic value on the server side from `"Hello World"` to the server's name:

Edited main.cpp +1 -1

Now when the client connects and reads the characteristic, it will display `"The characteristic value was: Kitty's MCU"`. If you also want the server to notify with the device name, you can update the notify value:

Edited main.cpp +2 -2

Done! Now rebuild and upload. The client will see `"Kitty's MCU"` in both the initial read and every notification.



```
#include <Arduino.h>
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEScan.h>
#include <BLEAdvertisedDevice.h>
// Client Code
#include "BLEDevice.h"
//#include "BLEScan.h"

// TODO: change the service UUID to the one you are using on the server side.
// The remote service we wish to connect to.
static BLEUUID serviceUUID("36a02f47-375d-43aa-8d99-94fdcad10669");
// The characteristic of the remote service we are interested in.
static BLEUUID    charUUID("beb5483e-36e1-4688-b7f5-ea07361b26a8");

static boolean doConnect = false;
static boolean connected = false;
static boolean doScan = false;
static BLERemoteCharacteristic* pRemoteCharacteristic;
static BLEAdvertisedDevice* myDevice;

// TODO: define new global variables for data collection
```

```
BLE Advertised Device found: Name: , Address: 72:28:a2:03:67:bd, manufacturer data: 4c0010062d1
e2d4c907d, txPower: 12, rssi: -91
BLE Advertised Device found: Name: , Address: 62:9b:21:d5:4d:63, manufacturer data: 4c000c0e086
027f9a88b14336c2b4d94d4e91006431d23868d28, rssi: -10
BLE Advertised Device found: Name: , Address: c5:35:f7:41:80:f9, manufacturer data: 4c001202000
2, rssi: -92
BLE Advertised Device found: Name: , Address: 5d:92:c3:6d:37:fe, manufacturer data: 4c000908138
bc0a808b61b581608005426cb20b370fe, rssi: -10
BLE Advertised Device found: Name: , Address: 98:3d:ae:ab:5e:9a, serviceUUID: 36a02f47-375d-43a
a-8d99-94fdcad10669, rssi: -82
Forming a connection to 98:3d:ae:ab:5e:9a
 - Created client
 - Connected to server
 - Found our service
 - Found our characteristic
The characteristic value was: Kitty's MCU
We are now connected to the BLE Server.
Setting new characteristic value to "Time since boot: 316"
Notify callback for characteristic dc2fef20-3b41-4fff-98a0-d4fffa143262 of data length 11
data: Kitty's MCU
Setting new characteristic value to "Time since boot: 317"
Notify callback for characteristic dc2fef20-3b41-4fff-98a0-d4fffa143262 of data length 11
data: Kitty's MCU
Setting new characteristic value to "Time since boot: 318"
Notify callback for characteristic dc2fef20-3b41-4fff-98a0-d4fffa143262 of data length 11
data: Kitty's MCU
Setting new characteristic value to "Time since boot: 319"
```

CHAT

Todos (1/4) - Determine connection method

(If you prefer, I can run those install commands here — tell me which installer you'd like.)

2. Use VS Code's built-in PlatformIO serial monitor:
   - In VS Code, open the PlatformIO toolbar (left activity bar) → "Serial Monitor" → choose the `seeed_xiao_esp32c3` environment and 115200 baud. Or click the "Monitor" icon in the status bar when a PIO environment is selected.
   - This requires no CLI install.

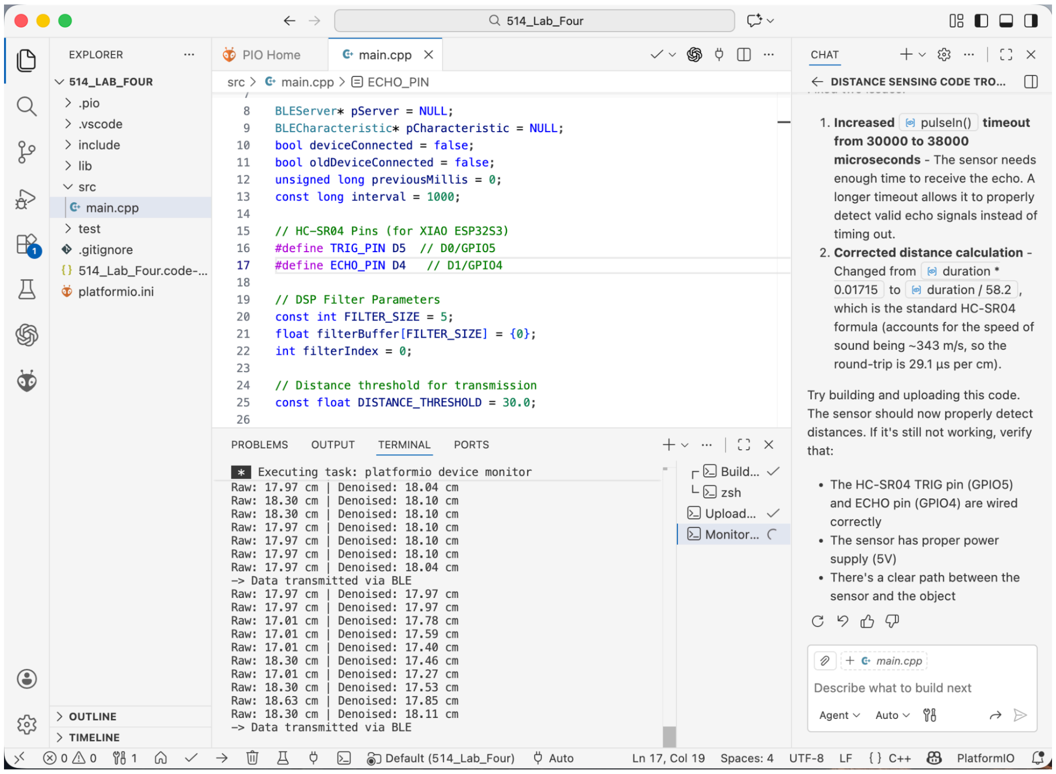
If you'd like, I can:

- Attempt to install PlatformIO CLI here and then open the monitor (I can run the pip or brew command).
- Or give step-by-step VS Code instructions with screenshots/short guidance (if you prefer GUI).
- Or show an alternate command (e.g., using screen/cu) if you prefer connecting directly to a known serial device path (e.g., /dev/tty.usbserial-xxxx). If you give the device path I can run a terminal monitor like `screen /dev/tty.usbserial-xxxx 115200`.

Which option do you prefer?

Server: Kitty's MCU

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



```
BLEServer* pServer = NULL;
BLECharacteristic* pCharacteristic = NULL;
bool deviceConnected = false;
bool oldDeviceConnected = false;
unsigned long previousMillis = 0;
const long interval = 1000;

// HC-SR04 Pins (for XIAO ESP32S3)
#define TRIG_PIN D5  // D0/GPIO5
#define ECHO_PIN D4   // D1/GPIO4

// DSP Filter Parameters
const int FILTER_SIZE = 5;
float filterBuffer[FILTER_SIZE] = {0};
int filterIndex = 0;

// Distance threshold for transmission
const float DISTANCE_THRESHOLD = 30.0;
```

```
 * Executing task: platformio device monitor
Raw: 17.97 cm | Denoised: 18.04 cm
Raw: 18.30 cm | Denoised: 18.10 cm
Raw: 18.30 cm | Denoised: 18.10 cm
Raw: 17.97 cm | Denoised: 18.10 cm
Raw: 17.97 cm | Denoised: 18.10 cm
Raw: 17.97 cm | Denoised: 18.10 cm
Raw: 17.97 cm | Denoised: 18.04 cm
-> Data transmitted via BLE
Raw: 17.97 cm | Denoised: 17.97 cm
Raw: 17.97 cm | Denoised: 17.97 cm
Raw: 17.01 cm | Denoised: 17.78 cm
Raw: 17.01 cm | Denoised: 17.59 cm
Raw: 17.01 cm | Denoised: 17.40 cm
Raw: 18.30 cm | Denoised: 17.46 cm
Raw: 17.01 cm | Denoised: 17.27 cm
Raw: 18.30 cm | Denoised: 17.53 cm
Raw: 18.63 cm | Denoised: 17.85 cm
Raw: 18.30 cm | Denoised: 18.11 cm
-> Data transmitted via BLE
```

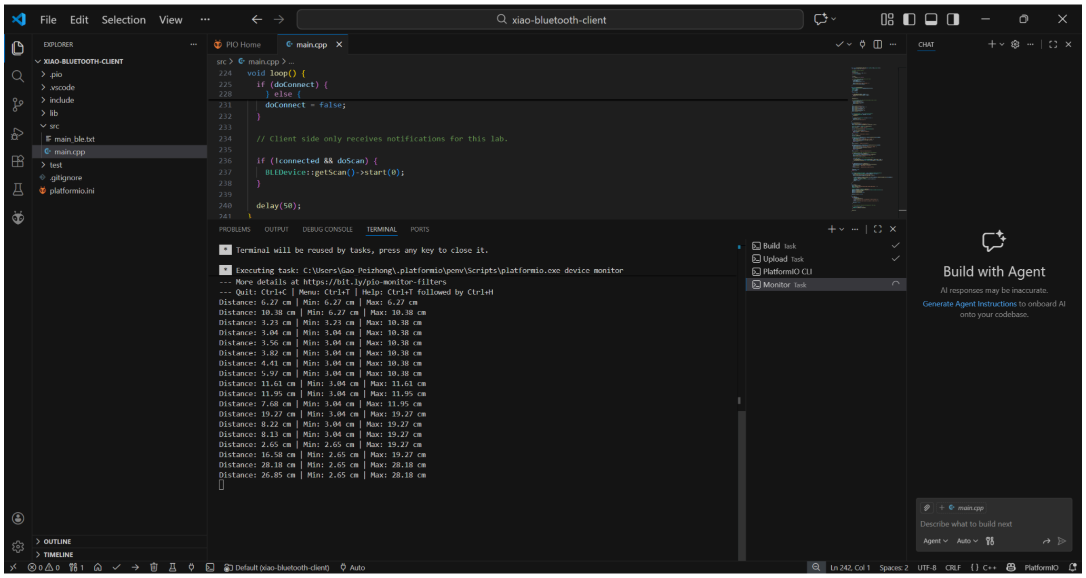
DISTANCE SENSING CODE TRO...

1. **Increased** pulseIn() **timeout from 30000 to 38000 microseconds** - The sensor needs enough time to receive the echo. A longer timeout allows it to properly detect valid echo signals instead of timing out.
2. **Corrected distance calculation** - Changed from duration * 0.01715 to duration / 58.2, which is the standard HC-SR04 formula (accounts for the speed of sound being ~343 m/s, so the round-trip is 29.1 µs per cm).

Try building and uploading this code. The sensor should now properly detect distances. If it's still not working, verify that:

- The HC-SR04 TRIG pin (GPIO5) and ECHO pin (GPIO4) are wired correctly
- The sensor has proper power supply (5V)
- There's a clear path between the sensor and the object

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



```
void loop() {
  if (doConnect) {
    } else {
    doConnect = false;
  }

  // Client side only receives notifications for this lab.

  if (!connected && doScan) {
    BLEDevice::getScan()->start(0);
  }

  delay(50);
```

```
 * Terminal will be reused by tasks, press any key to close it.

 * Executing task: C:\Users\Gao Peizhong\.platformio\penv\Scripts\platformio.exe device monitor
--- More details at https://bit.ly/pio-monitor-filters
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H
Distance: 6.27 cm | Min: 6.27 cm | Max: 6.27 cm
Distance: 10.38 cm | Min: 6.27 cm | Max: 10.38 cm
Distance: 3.23 cm | Min: 3.23 cm | Max: 10.38 cm
Distance: 3.04 cm | Min: 3.04 cm | Max: 10.38 cm
Distance: 3.56 cm | Min: 3.04 cm | Max: 10.38 cm
Distance: 3.82 cm | Min: 3.04 cm | Max: 10.38 cm
Distance: 4.41 cm | Min: 3.04 cm | Max: 10.38 cm
Distance: 5.97 cm | Min: 3.04 cm | Max: 10.38 cm
Distance: 11.61 cm | Min: 3.04 cm | Max: 11.61 cm
Distance: 11.95 cm | Min: 3.04 cm | Max: 11.95 cm
Distance: 7.68 cm | Min: 3.04 cm | Max: 11.95 cm
Distance: 19.27 cm | Min: 3.04 cm | Max: 19.27 cm
Distance: 8.22 cm | Min: 3.04 cm | Max: 19.27 cm
Distance: 8.13 cm | Min: 3.04 cm | Max: 19.27 cm
Distance: 2.65 cm | Min: 2.65 cm | Max: 19.27 cm
Distance: 16.58 cm | Min: 2.65 cm | Max: 19.27 cm
Distance: 28.18 cm | Min: 2.65 cm | Max: 28.18 cm
Distance: 26.85 cm | Min: 2.65 cm | Max: 28.18 cm
```