

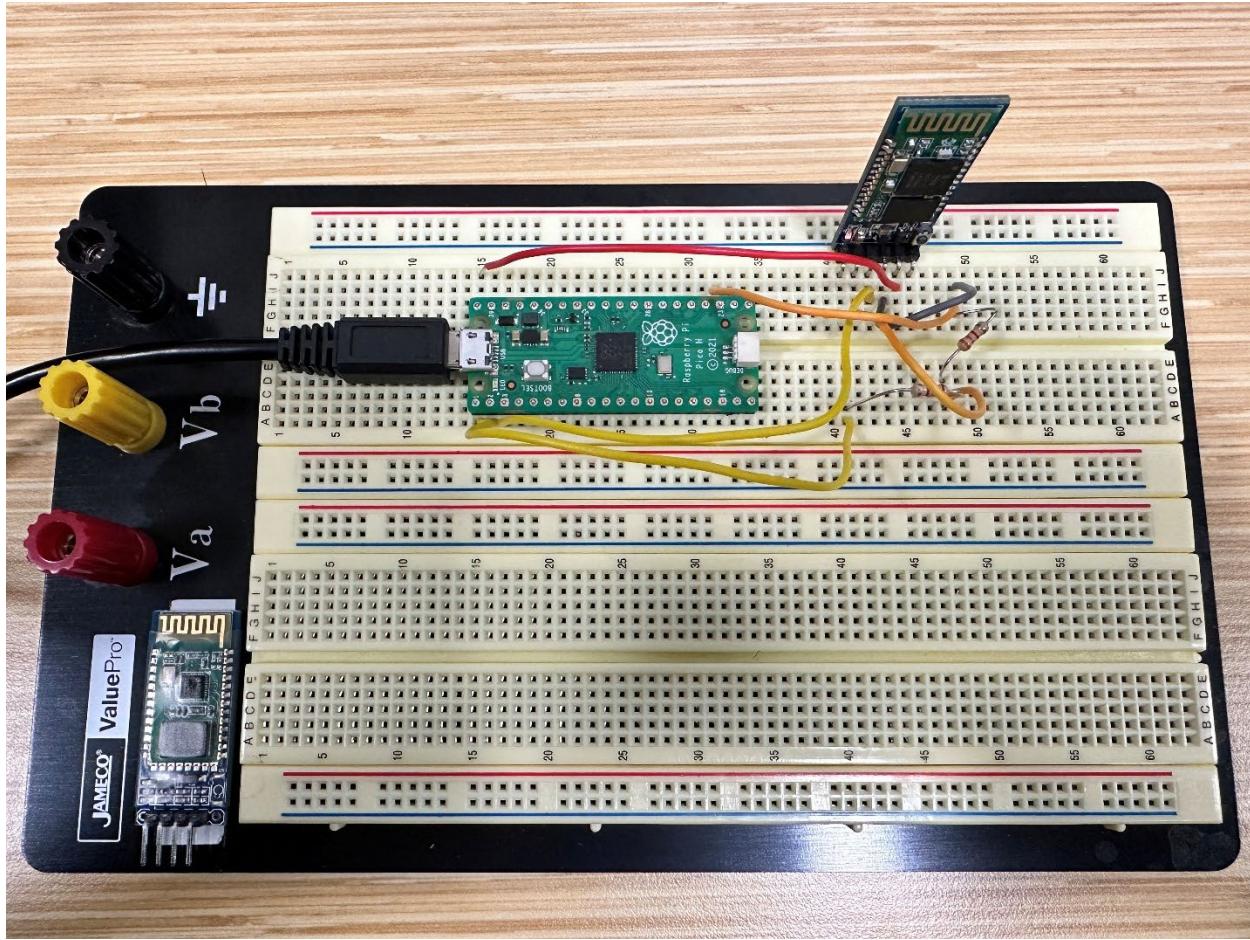
Tutorial: Getting the HC-05 Bluetooth Working with Your Raspberry Pi Pico

Why HC-05 Will NOT Show on iPhone:

- **Classic Bluetooth vs BLE:** HC-05 uses Bluetooth Classic (SPP) while iOS prefers BLE (Bluetooth Low Energy).
- **MFi Program:** Apple's Made for iPhone program restricts certain Bluetooth profiles.
- **App Requirement:** You need a dedicated app that supports SPP/Classic Bluetooth.

Introduction

Some of you mentioned having trouble with the HC-05 and the Pico. I tried it out, and it is actually very straightforward with the correct wiring and code! I have created this custom guide to walk you through it step-by-step, using both the Arduino IDE. I have tested this myself and it works perfectly.



This is what your final setup should look like. Notice the junction on the breadboard: A-J #49.

★ HC-05 + Raspberry Pi Pico Tutorial (Using Arduino IDE)

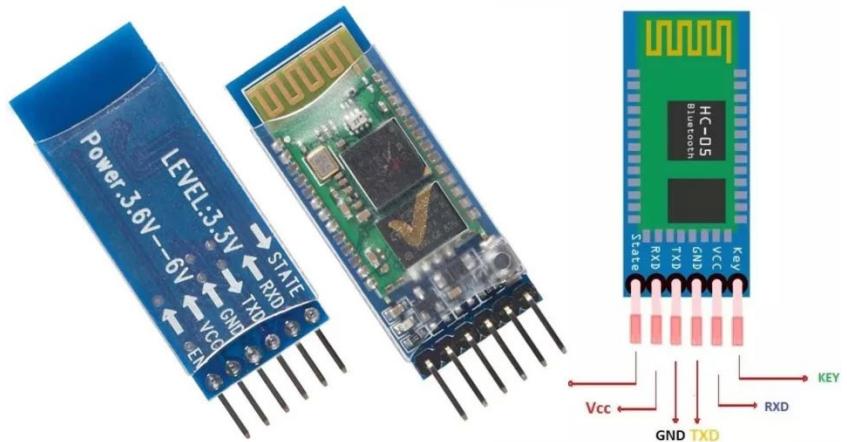
Full Steps: Wiring → AT Mode → Reset → Windows Pairing → Serial Communication

1. Hardware You Need

- Raspberry Pi Pico
 - HC-05 Bluetooth module (DSD Tech version is OK)
 - Jumper wires
 - Breadboard
 - **1.8 kΩ resistor + 3.3 kΩ resistor** (for voltage divider) on Row A-J #49 in the photo above.
-

2. HC-05 Pinout

HC-05 Pin	Use
VCC	5V input
GND	Ground
TXD	Bluetooth → Pico RX
RXD	Pico TX → Bluetooth
EN / KEY	Not needed
STATE	Not needed



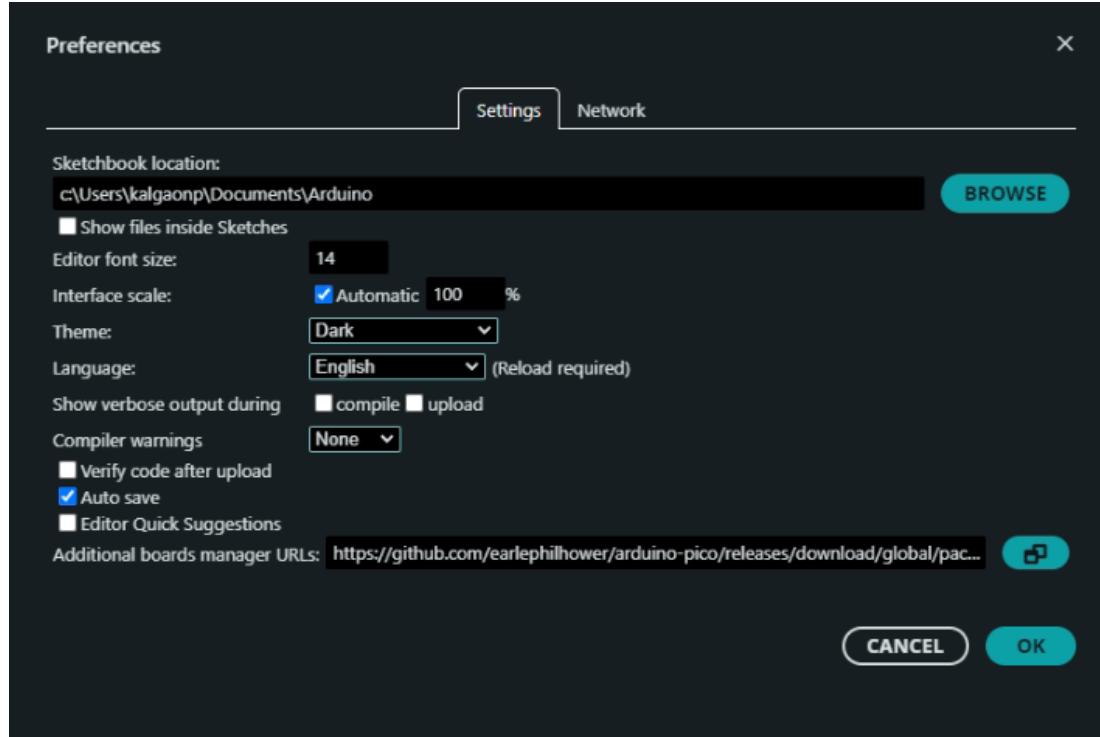
Part 1: Setup with Arduino IDE (The Quickest Way)

This is the fastest way to get started and test the demo code first. Then port it to VS Code.

1. Installing Pico Support in Arduino IDE

If you haven't already, you need to add support for the Raspberry Pi Pico to your Arduino IDE.

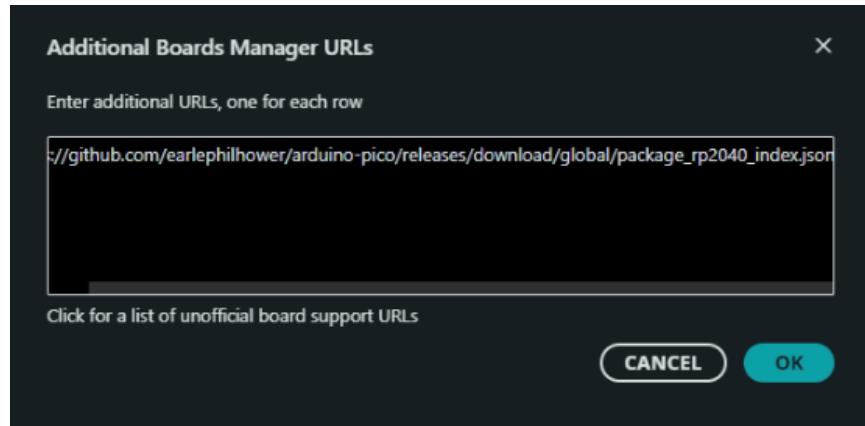
1. Open the Arduino IDE.
2. Go to **File > Preferences**.



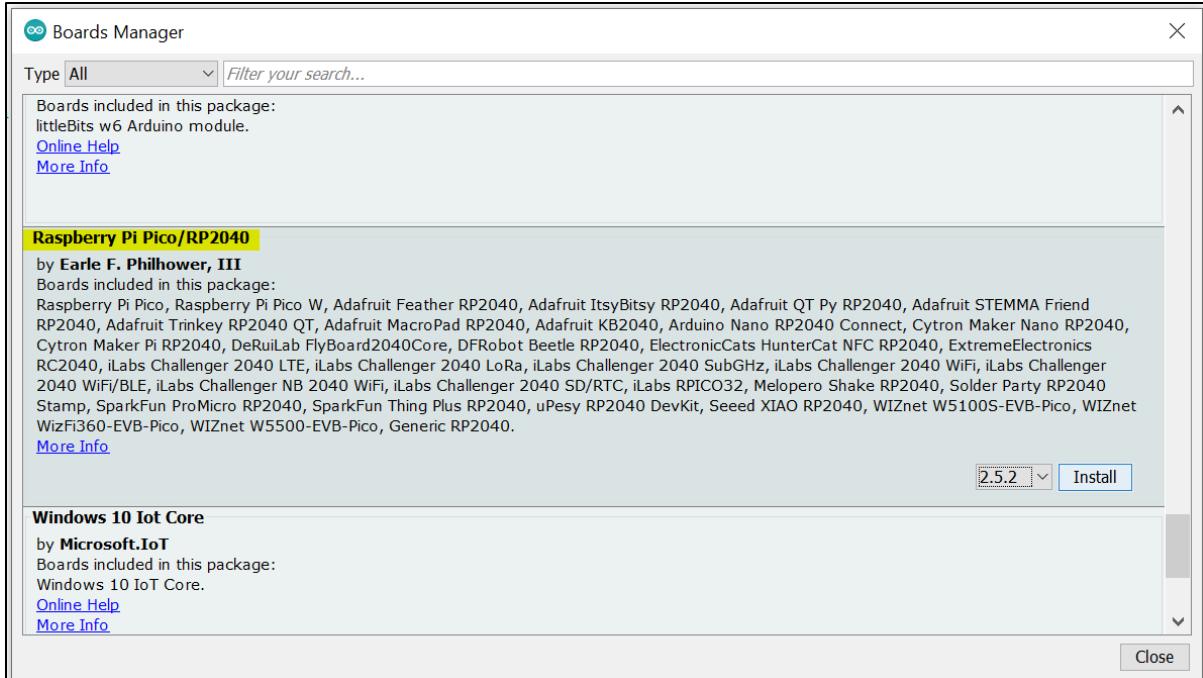
3. In the "Additional Boards Manager URLs" field, paste the following URL:

`https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json`

(If you already have other URLs in there, you can add this one on a new line.)



4. Click OK.
5. Now, go to Tools > Board > Boards Manager....
6. Search for "Raspberry Pi Pico/RP2040" and install the package by Earle F. Philhower.



3. Wiring HC-05 to Pico

⚠ HC-05 requires 5V on VCC.

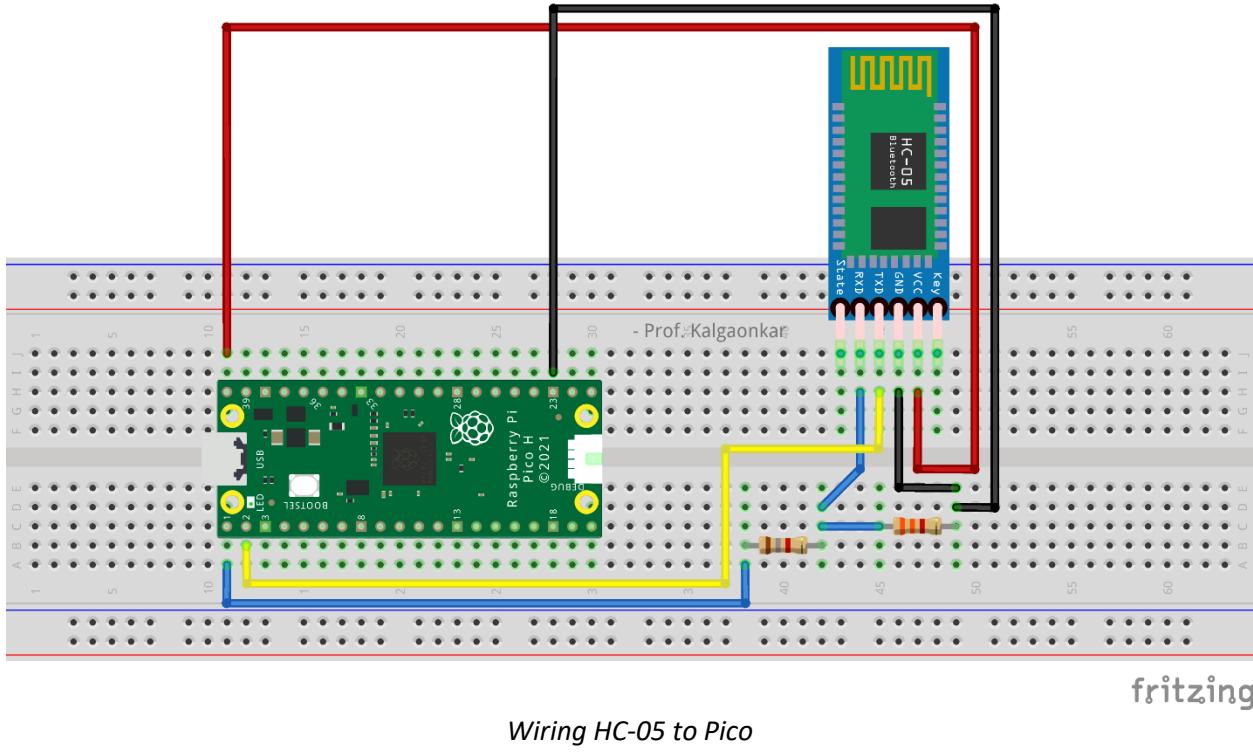
⚠ RX pin must NOT receive full 3.3V — use voltage divider.

Voltage Divider (for Pico TX → HC-05 RX)

- **R1 = 1.8kΩ** (from GPO → junction)
- **R2 = 3.3kΩ** (junction → GND)
- Junction goes → HC-05 RXD

Final Connections

HC-05	Pico	Notes
VCC	VBUS (Pin 40)	5V from USB
GND	GND	MUST share common ground
TXD	GP1 (UART0 RX)	Direct connection
RXD	GPO (UART0 TX)	Through voltage divider
EN (KEY)	3.3V ONLY when entering AT mode	OPTIONAL



- ✓ This protects HC-05 RX.
- ✓ HC-05 TX → Pico RX is already safe.

4. Factory Reset Using Arduino IDE

1. Copy the following code in to your Arduino IDE.
2. You need to put the Pico into **Bootloader Mode**: hold down the **BOOTSEL** button on the Pico, plug it into your computer, and then release the button.
3. Now, compile and flash your Pico through the Arduino IDE now.

```
void setup() {
    Serial.begin(115200);
    Serial1.begin(38400); // AT-mode baud
    delay(2000);

    Serial.println("Factory Resetting HC-05...");

    sendAT("AT");
    sendAT("AT+ORGL"); // Factory reset
    sendAT("AT+UART=9600,0,0"); // Set normal operation baud
    sendAT("AT+NAME=ECE414HC05"); // Device name
    sendAT("AT+PSWD=1234"); // Bluetooth PIN
    sendAT("AT+ROLE=0"); // Slave mode (default)
    sendAT("AT+CMODE=1"); // Allow any device to connect
    sendAT("AT+RESET"); // Reboot

    Serial.println("Factory reset complete!");
    Serial.println("Power cycle HC-05 WITHOUT pressing button.");
}
```

```

}

void sendAT(String cmd) {
    Serial1.println(cmd);
    delay(500);
    Serial.print(cmd);
    Serial.print(" : ");
    while (Serial1.available()) {
        Serial.write(Serial1.read());
    }
    Serial.println();
}

void loop() {}

```

The output in your terminal (TeraTerm or Arduino Serial Monitor) will be:

```

17:44:44.803 -> AT+ROLE=0 :
17:44:45.307 -> AT+CMODE=1 :
17:44:45.808 -> AT+RESET :
17:44:45.808 -> Factory reset complete!
17:44:45.808 -> Power cycle HC-05 WITHOUT pressing button.

```

This is because the code is ready to factory reset your HC-05 bluetooth module but your HC-05 is not in the *AT Mode* yet. Hence the status is blank for each AT mode command in the terminal. Follow the next steps to enter the AT mode in order to factory reset your Bluetooth module.

5. Entering AT Mode

Steps:

1. Remove the Pico-USB cable from your PC.
2. **Hold the small button** on the HC-05 module.
3. **Power the Pico** (plug in back the USB cable). Keep holding down the HC-05 button, do not let go! Timing is critical!
4. If successful, HC-05 Red LED should blink **slow** → AT mode active.
5. Release button and watch serial terminal (TeraTerm/Arduino monitor).
6. Be sure to set **Baud Rate to 38400** in your terminal!

You should now see this:

```

17:52:01.079 -> Factory Resetting HC-05...
17:52:01.585 -> AT : OK
17:52:01.585 ->
17:52:02.086 -> AT+ORGL : OK
17:52:02.086 ->
17:52:02.587 -> AT+UART=9600,0,0 : OK

```

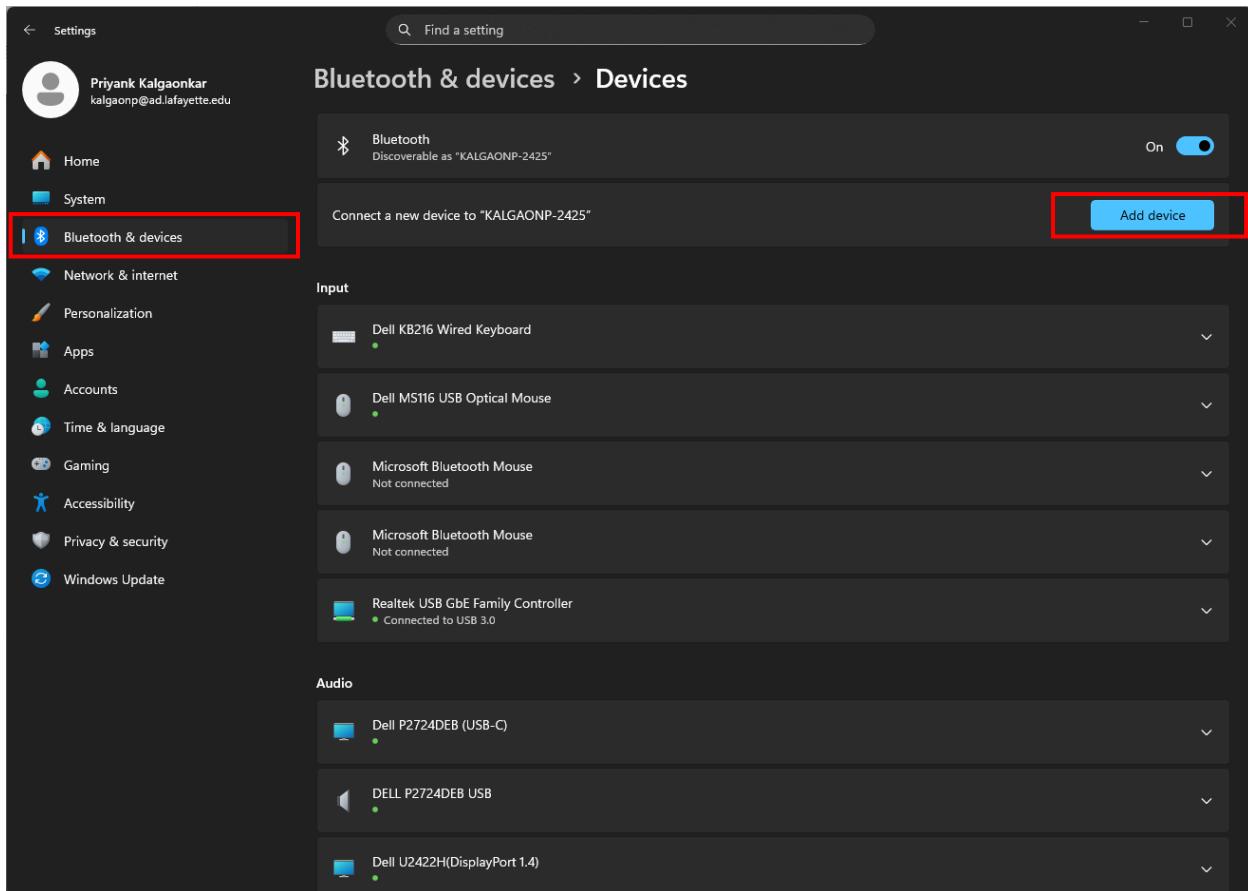
```
17:52:02.587 ->
17:52:03.075 -> AT+NAME=ECE414HC05 : OK
17:52:03.075 ->
17:52:03.577 -> AT+PSWD=1234 : OK
17:52:03.577 ->
17:52:04.081 -> AT+ROLE=0 : OK
17:52:04.081 ->
17:52:04.582 -> AT+CMODE=1 : OK
17:52:04.582 ->
17:52:05.086 -> AT+RESET : OK
```

After this step, Red LED on HC-05 bluetooth module should be rapidly blinking to indicate ready to pair.

6. Pairing HC-05 to Windows 10/11

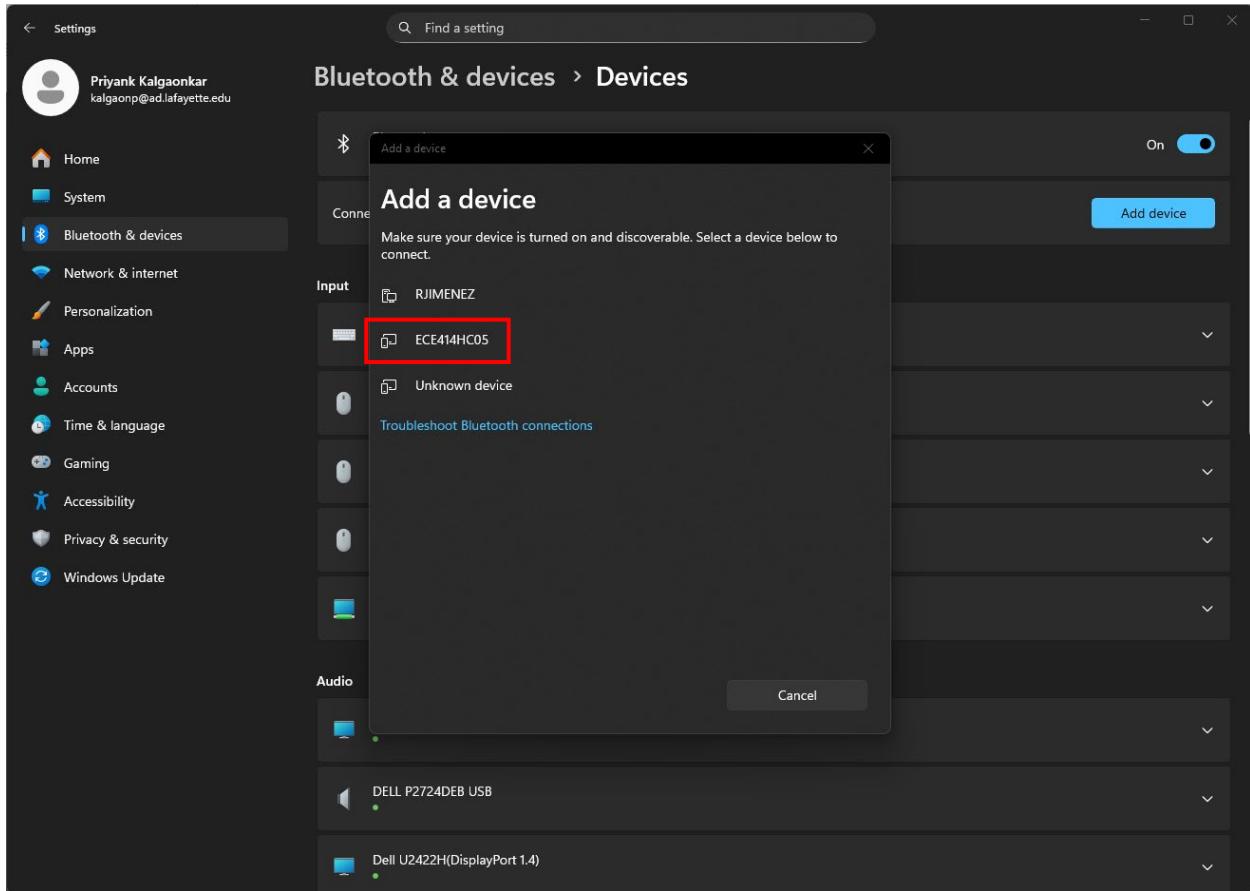
A. Open Windows Bluetooth Settings:

Settings → Bluetooth & Devices

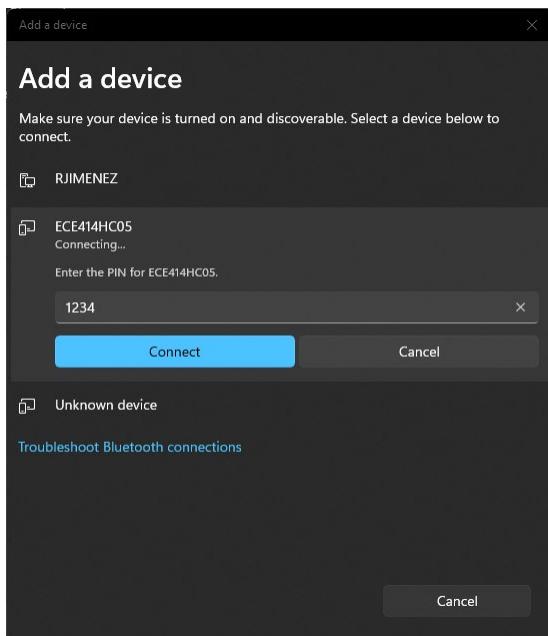


B. Click Add Device → Bluetooth → Show All Devices

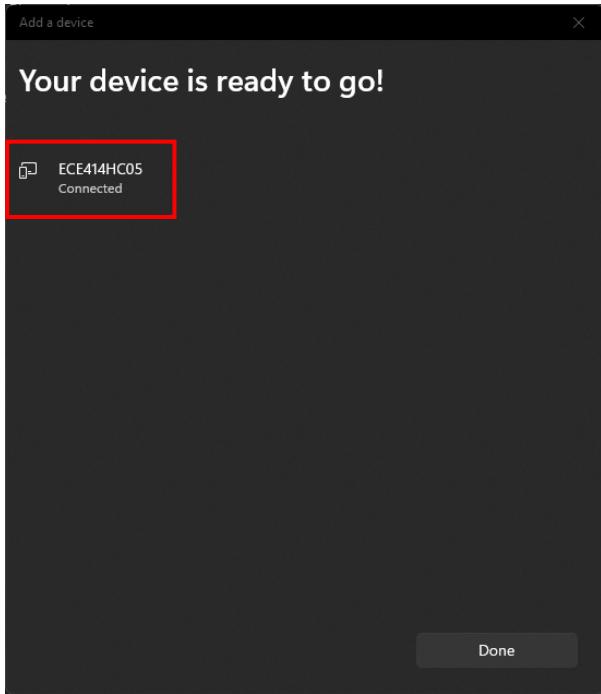
You should see:



Select it → enter PIN **1234**.

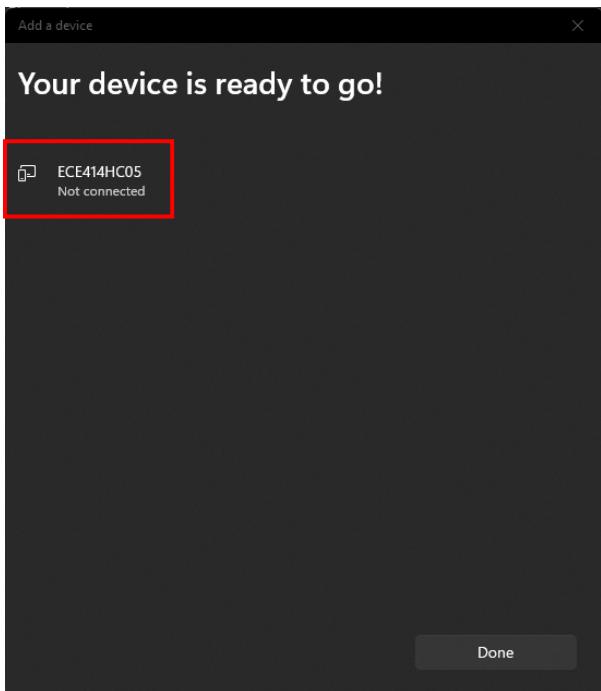


Hit ‘Connect’. Windows now says: **Connected**



→ Shortly after, it will then disconnect (This is normal! Do not panic! 😊)

This is expected.



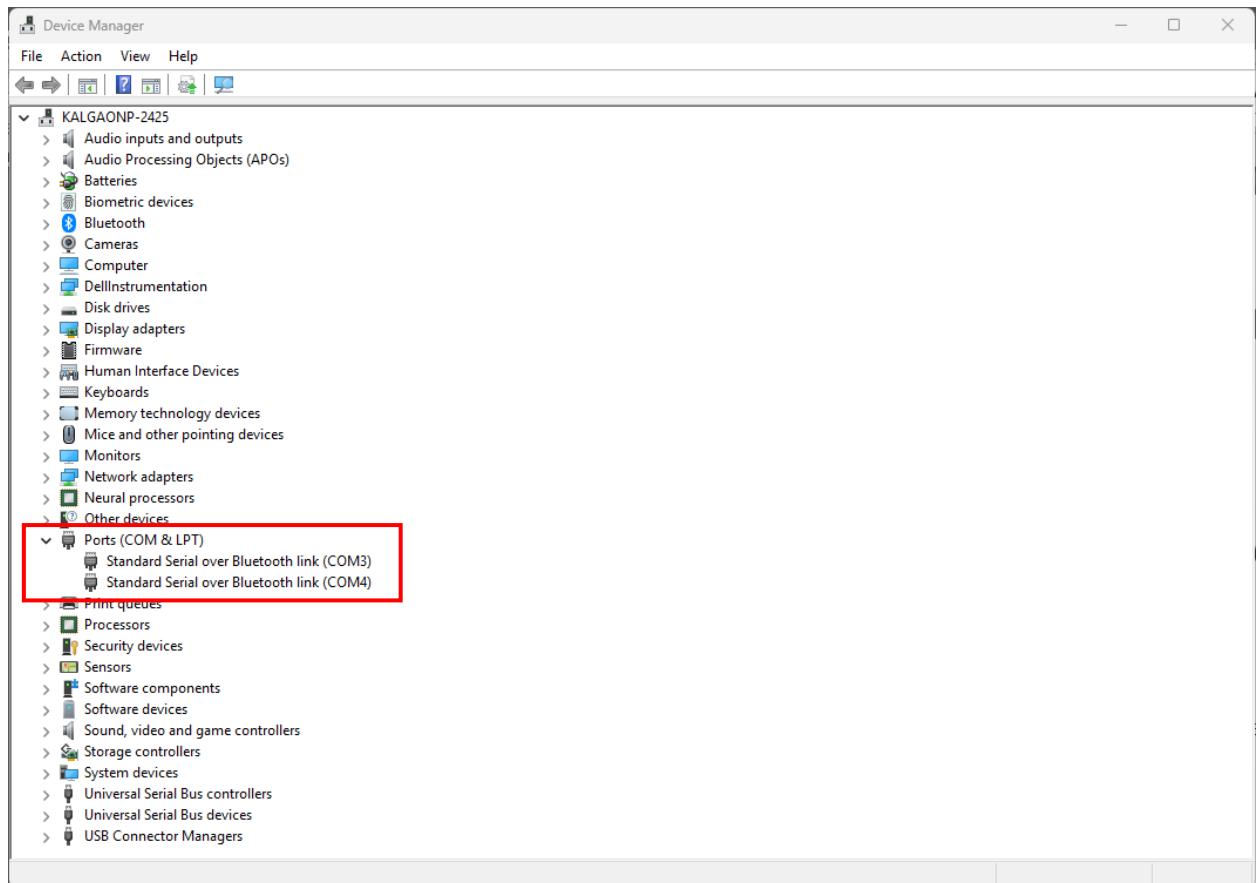
7. Find the Bluetooth COM Port (Windows 11)

Use Device Manager (most reliable)

1. Press:

Win + X → Device Manager

2. Expand: Ports (COM & LPT)



Either of these is the actual Bluetooth SPP port.

Pssst: In my case, it is COM3 which I found out after trying out both COM3 and COM4 ports in the later stages of this tutorial.

8. Pico Code for Sending/Receiving Data

Replace and copy-paste this echo/pass-through code in your Arduino IDE in its entirety, and flash your Pico:

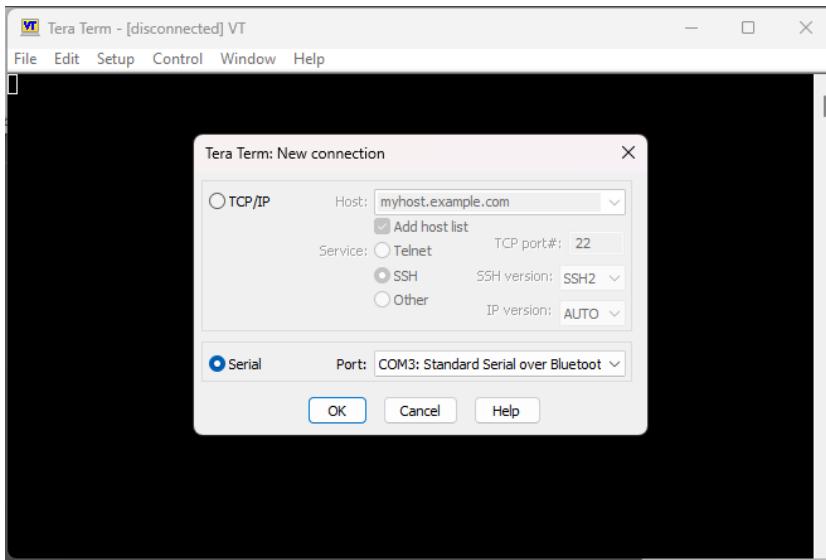
```
void setup() {
    Serial.begin(115200);      // USB to PC
    Serial1.begin(9600);       // HC-05 UART
}

void loop() {
    if (Serial1.available()) {
        Serial.write(Serial1.read());
    }
    if (Serial.available()) {
        Serial1.write(Serial.read());
    }
}
```

9. Open the Bluetooth Serial Port in TeraTerm

Steps:

1. Open a new **TeraTerm** terminal



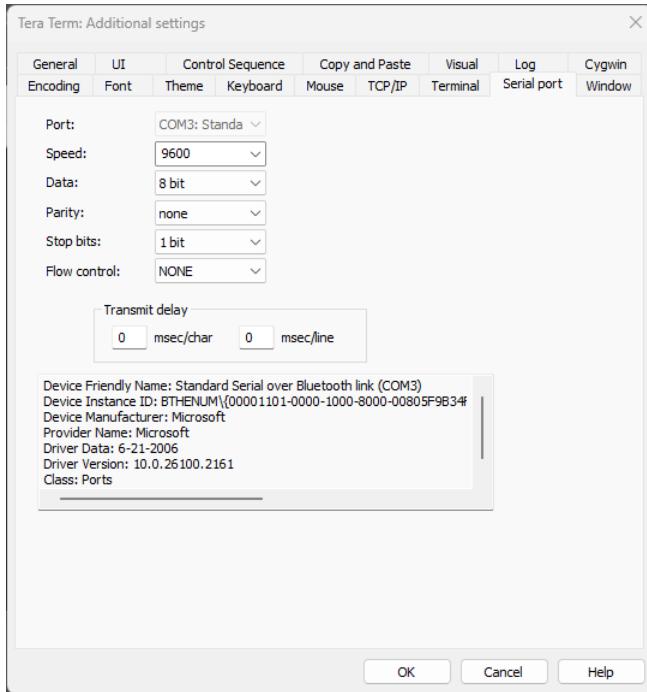
Select: Serial and COM port. (Note: You will have to experiment and see which COM port actually works for Bluetooth communication).

Click **OK**.

If it is the correct COM Port, your HC-05 Bluetooth module's Red LED will flash twice every second.

This means that the Pico-HC-05 is now paired and connected to your Windows PC.

2. Configure your Serial Port in TeraTerm:



Set **baud rate** to **9600**.

What is happening now:

- ✓ HC-05 LED changes to **double-blink every second**
 - ✓ Windows shows “**Connected**”
 - ✓ You now have a working Bluetooth serial link
-

10. Test 2-Way Communication

```

sketch_dec2a | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Raspberry Pi Pico
sketch_dec2a.ino
1 void setup() {
2   Serial.begin(115200); // USB to PC
3   Serial1.begin(9600); // HC-05 UART
4 }
5
6 void loop() {
7   if (Serial1.available()) {
8     Serial.write(Serial1.read());
9   }
10  if (Serial.available()) {
11    Serial1.write(Serial.read());
12  }
13 }
14

```

Output Serial Monitor ×

Message (Enter to send message to 'Raspberry Pi Pico' on 'COM7')

No Line Ending 38400 baud

Building sketch

Ln 14, Col 1 Raspberry Pi Pico on COM7 2

A. Send from message over Bluetooth

In Arduino Serial Monitor (HC-05 COM), type in:

Hello Pico!

```

sketch_dec2a | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Raspberry Pi Pico
sketch_dec2a.ino
1 void setup() {
2   Serial.begin(115200); // USB to PC
3   Serial1.begin(9600); // HC-05 UART
4 }
5
6 void loop() {
7   if (Serial1.available()) {
8     Serial.write(Serial1.read());
9   }
10  if (Serial.available()) {
11    Serial1.write(Serial.read());
12  }
13 }
14

```

Output Serial Monitor ×

Hello Pico!

No Line Ending 38400 baud

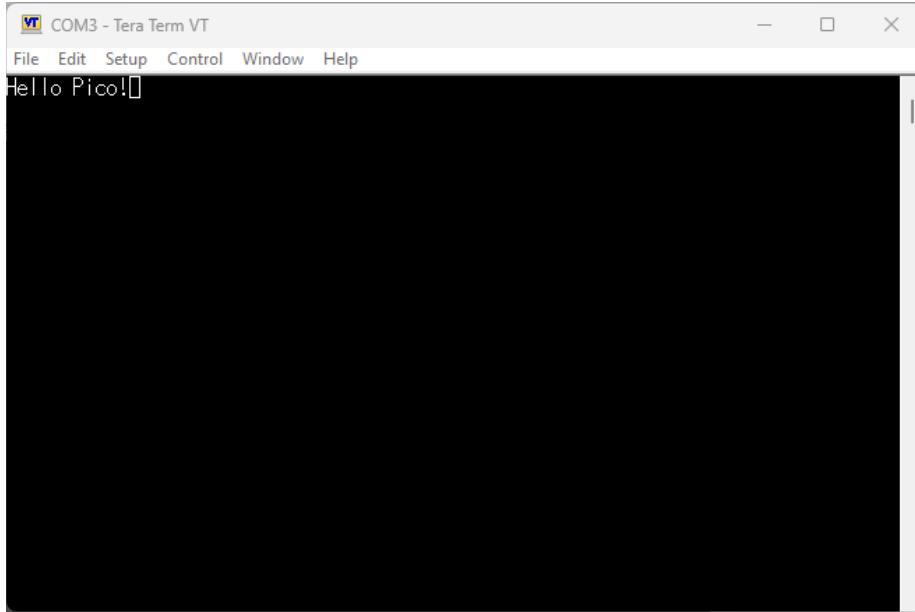
Building sketch

Ln 14, Col 1 Raspberry Pi Pico on COM7 2

Hit the **Enter** key on your keyboard.

This sends the ‘Hello Pico!’ string from your Arduino IDE to the Windows PC COM3 port via Pico + HC-05 over Bluetooth.

Now, in the TeraTerm terminal, you should see the message sent wirelessly over bluetooth:



→ Appears on the **Bluetooth Serial Monitor**

- ✓ 2-way Bluetooth communication confirmed
- ✓ HC-05 fully working
- ✓ Wiring & baud rate correct

Note that I am using two different windows PCs to create this tutorial, one is used to send the message the other is to receive it. The receiving windows PC has an inbuilt Bluetooth functionality.

Tutorial Complete

You now successfully:

- ✓ Wired HC-05 safely
- ✓ Entered AT Mode
- ✓ Factory reset & configured the module
- ✓ Paired it to Windows
- ✓ Found outgoing COM port
- ✓ Opened a Bluetooth serial session
- ✓ Sent & received data between Pico and Windows PC

Good luck with your projects! 😊

Links for reference:

Physical setup photos:

- <https://drive.google.com/file/d/1acZ8--c8RCdmcMjjqUjLoCQeHgeaTxGS/view?usp=sharing>

Video:

- <https://drive.google.com/file/d/1GIWd8nUkWbApizmgz8QzmVkbEznXsuho/view?usp=sharing>

GitHub Link:

- <https://github.com/priyankkalgaonkar/Pico-HC05-Bluetooth-Tutorial.git>