troniction

BUILD IT. UNDERSTAND IT.

Arduino Bluetooth Make It Connect

Every way the link fails, why, and the fix.
HC-05, HC-06 and HM-10 BLE.

HC-05 & HC-06 | HM-10 BLE | CLONES TOO | 2026

/// START HERE

What is it doing?

Find the symptom. Go to the page. This book is written to be opened in the middle, at the moment something stopped working.

/// BEFORE YOU REPLACE ANYTHING

Almost none of these are a broken module. Power, baud rate, line endings and which pins you used account for most of them — and every one of those is free to fix.

Page numbers throughout refer to the full 62-page book. This sample is its first 11 pages.

/// FRONT MATTER

How to use this book

This is a book to be opened in the middle.

If something is broken right now, use the symptom index on the page before this, or Chapter 13's matrix at the end. Both are written in the words you would use, not the words an engineer would use.

If you are starting out, read Chapters 1 to 4 in order and stop there for the night: four boxes, one purchase decision, four wires, and a proof that the radio works.

If you are choosing what to buy, Chapters 2 and 10 are the whole decision — and Chapter 10 is the one software cannot undo.

What you need on the desk

- [ ] An **Arduino Uno**, Nano or clone
- [ ] A **Bluetooth module** — which one is Chapter 2
- [ ] **Two resistors**, 1 kΩ and 2 kΩ — Chapter 3 explains why
- [ ] **Jumper wires** and a breadboard
- [ ] A **phone** — Android for the simplest path

No soldering. No oscilloscope. Nothing on that list is exotic.

> /// CONVENTIONS
>
> `Like this` is something you type. A coral bar means a box can cost you money; teal means it saves you an evening.

/// PART ONE

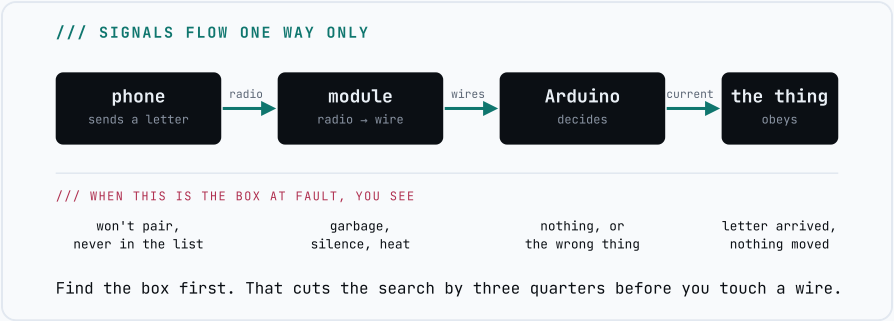
Know what you’re holding

Four things in a line, and signals only ever flow one way through them. Before any of it can work, you need to know which module is in your hand — because it is quite possibly not the one on the listing.

/// CHAPTER 1

The four-box model

Every project in this book is the same four things, and signals only ever flow one way through them.



The **module** is a radio glued to a wire. It receives over the air and hands what it got to the Arduino as though it had been typed on a keyboard — and it understands none of it. The **Arduino** is the only part that decides anything.

That is the whole system. Every chapter after this one fills in exactly one of those boxes, and we never take on two at once.

/// CHAPTER 1

Each box fails in its own way

This is why the model earns its place: **the boxes have different symptoms**. Once you can recognise which one is complaining, you have cut the search by three quarters before touching anything.

BOX	WHEN IT IS THE PROBLEM, YOU SEE
Phone	Won't pair. Never appears in the list. Pairs and drops
Module	Garbage characters. Total silence. The module gets warm
Arduino	The module is clearly fine, but the sketch does nothing, or the wrong thing
The thing	All of the above works, the letter arrives — and the motor sits there

Notice that last row. **A dead motor is not a Bluetooth problem**, and a surprising number of people spend a weekend on the radio because a battery was flat. The model is as useful for ruling boxes *out* as for finding the culprit.

> /// ONE WAY ONLY
>
> The phone cannot talk to the Arduino. It talks to the module, which talks to the Arduino. When a tutorial says "send a command to your Arduino", what physically happens is two separate handoffs — and either can be the broken one.

/// CHAPTER 1

Prove them in order

The habit is worth more than any single fix in this book: **prove the boxes left to right, and stop at the first one that fails.**

01 **Is the module powered and advertising?** A blinking LED. No phone involved yet. → Chapter 3

02 **Can a phone see it and connect?** A terminal app, no code written yet. → Chapter 4

03 **Does a typed letter reach the Arduino?** → Chapters 4 and 5

04 **Does the sketch do the right thing with it?** → Chapter 8

05 **Does the thing move?** → Chapter 9

Most people start at step 4 or 5 — writing the whole project, uploading it, and finding nothing happens. At that point every box is unproven at once and there is nothing to do but guess. Starting at step 1 costs ten minutes and turns a mystery into a location.

/// THE TWO RULES THAT GO WITH IT

Change one thing, then re-test — two changes at once and the result tells you nothing. And **don't debug the code until the link is proved**. Chapter 4 exists entirely for that.

/// CHAPTER 2

Which module did you actually buy?

Three modules sit behind almost every Arduino Bluetooth project, and they are not interchangeable. Picking the wrong one produces no error — just a project that works for everybody except you.

The whole decision is one question: **what has to talk to it?**

	HC-06	HC-05	HM-10
Bluetooth	Classic 2.0	Classic 2.0	**BLE 4.0**
Android	yes	yes	yes
iPhone	**no**	**no**	**yes**
Role	slave only	slave *or master*	slave or master
Setting up	barely needed	full AT set	full AT set
Price	cheapest	a little more	the dearest

If you remember one row, remember the iPhone row. It cannot be fixed in software, and it does not look like a hardware problem. Your code is fine, your wiring is fine — the phone simply cannot see the module, and never will.

/// THE SHORT ANSWER

First project, Android phone — **HC-06**.
An iPhone in the room — **HM-10**.
Two Arduinos talking — **HC-05**.

/// CHAPTER 2

What you give up, either way

The HC-06 gets out of your way

It is a slave and only a slave: it advertises, and something else connects to it. For a phone-drives-a-thing project that is the entire requirement, and the simplicity is a real feature — there is almost nothing to configure, so almost nothing to configure wrongly. What you give up is initiating. An HC-06 can never be the one that reaches out.

The HC-05 is the same part with ambition

Same footprint, same four wires, same 9600 baud out of the box. What you gain is **master mode** — it can go and find another module, which is what you need the day one Arduino has to talk to another across a room. Plenty of people buy one and use it purely as a slave. That is a sensible thing to do.

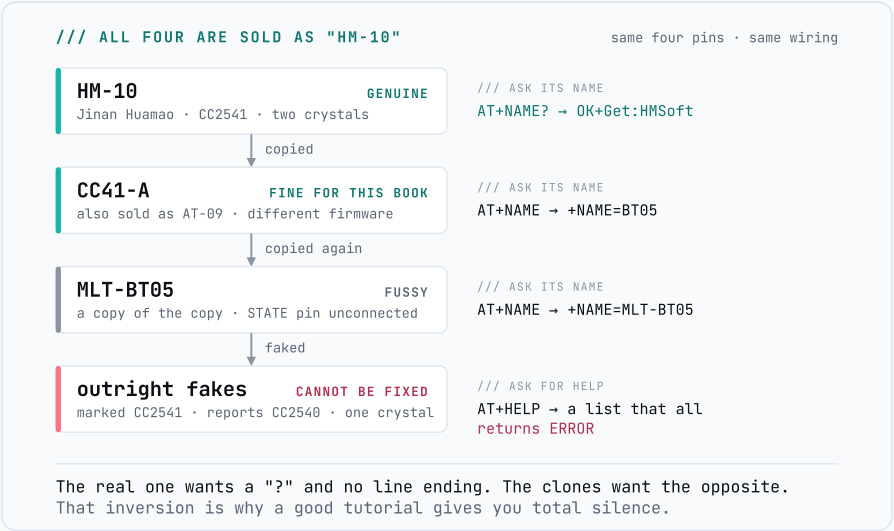
The HM-10 is a different radio, not a better one

This is the part people get wrong. The HM-10 is not "the new HC-05". It speaks **Bluetooth Low Energy** — a different protocol that happens to share a name and a logo. That buys two things nothing else here can offer: iPhones can see it, and a web page can drive it directly.

/// CHAPTER 2

You may not own the module you think you own

The HM-10 is made by Jinan Huamao. It sells well. So it was copied — and then the copy was copied, and then the copy of the copy was faked. All of them are sold as "HM-10", often in the same listing, at the same price.



They are not equally bad, and that is the useful part.

All four use the **same four pins** and the same wiring. Nothing in the next chapter changes. What changes is which AT commands exist, what they answer, and — for the fakes — whether the thing can be made to work at all.

/// CHAPTER 2

The sixty-second test

You do not need a programmer or a microscope. You need a terminal and two commands. Wire the module as in Chapter 3, open the Serial Monitor at **9600**, and stop at the first line that answers.

01 Send `AT+NAME?` — with the question mark, line ending set to **No line ending**. An answer of `OK+Get:HMSoft` means a **genuine HM-10**.

02 No answer? Send `AT+NAME` — *without* the question mark, line ending **Both NL & CR**. `+NAME=BT05` is a **CC41-A** (an AT-09 is the same thing); `+NAME=MLT-BT05` is an **MLT-BT05**.

03 Confirm with the version. `AT+VERS?` returns an `HMSoft` string on a genuine module.

04 Suspect a fake if `AT+HELP` lists a generous command set and those commands then return `ERROR`.

/// WHAT THAT ACTUALLY TESTED

The real module wants a `?` and no line ending. The clones want no `?` and a proper line ending. The syntax is **inverted** between them — which is why a tutorial written for one produces total silence on the other, and why so many people conclude their module is dead when it is merely foreign.

/// THAT WAS THE SAMPLE

Fifty-one more pages

You have read the mental model, and how to work out which module is actually in your hand — the part that explains why a good tutorial gave you a bad result. **The rest is the part that fixes it.**

/// THE FULL BOOK — $9

troniction.com/bluetooth

62 pages, yours to keep. Every diagram drawn from the real circuit; every number checked against a datasheet.