

## ESP32 Bluetooth RFCOMM, controlled from Android Phone with RoboRemo

Step 1. Upload code to your ESP32 board.

```
#include <BluetoothSerial.h>

// Your ESP32 will act as RFCOMM Server device
// In RoboRemo tap Menu -> CONNECT -> Bluetooth (RFCOMM) -> ESP32-RFCOMM
// (Device must be paired first from Android Bluetooth Settings)
BluetoothSerial SerialBT;

void setup() {
  Serial.begin(115200);
  delay(1000);

  Serial.println("Starting Bluetooth RFCOMM server...");

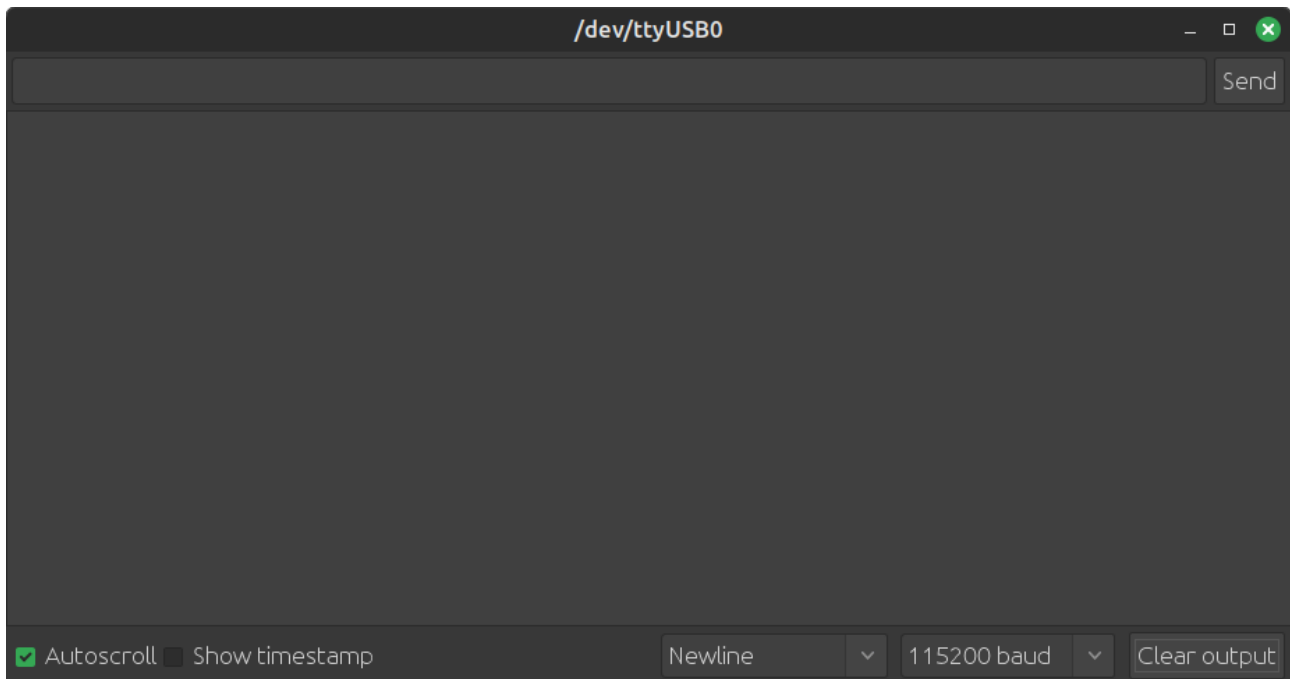
  if (!SerialBT.begin("ESP32-RFCOMM")) {
    Serial.println("An error occurred initializing Bluetooth");
    return;
  }
  Serial.println("Bluetooth device is ready to pair");
}

void loop() {

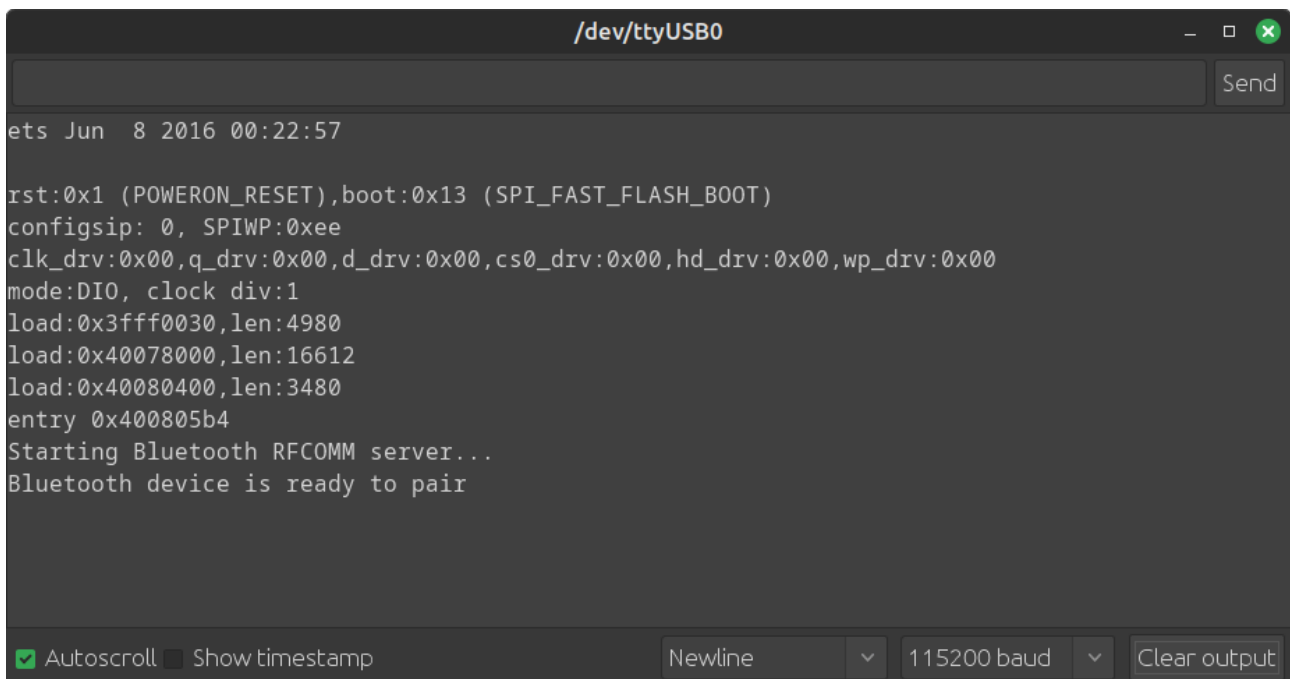
  if (SerialBT.hasClient()) {
    Serial.println("Client connected");
    String command = "";
    while (SerialBT.connected()) {
      if (SerialBT.available()) {
        char c = SerialBT.read();
        if (c == '\n') {
          // End of command
          Serial.println("Received: " + command);
          // Example: toggle an onboard LED or parse commands
          // For now, just echo back
          SerialBT.println("Received: " + command);

          // Clear command buffer
          command = "";
        } else {
          command += c;
        }
      }
    }
    Serial.println("Client disconnected");
  }
}
```

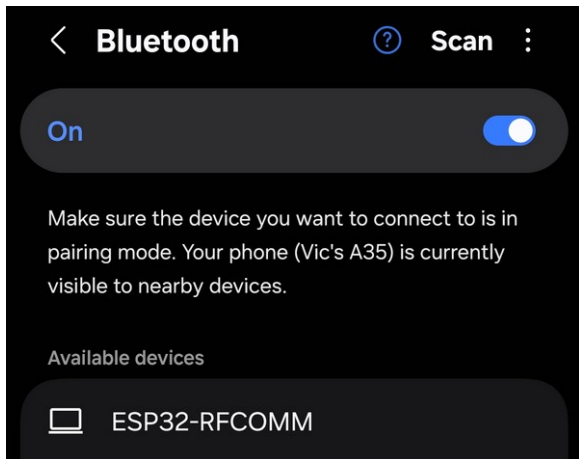
## Step 2. Open Serial Monitor



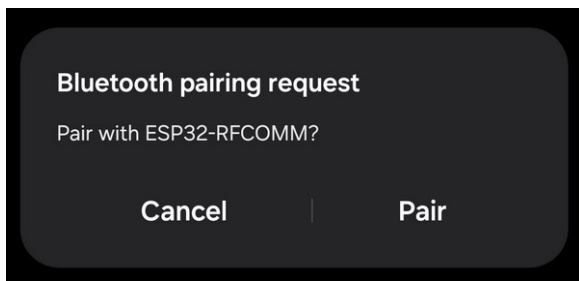
## Step 3. Reset the ESP32 by pressing the reset button on the board



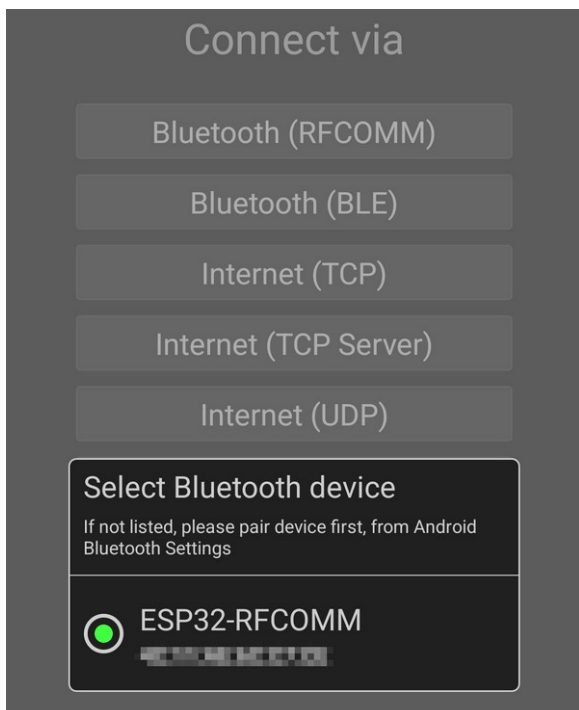
Step 4. ON your Android phone, activate Bluetooth and scan for devices.



Step 5. Pair the “ESP32-RFCOMM” device

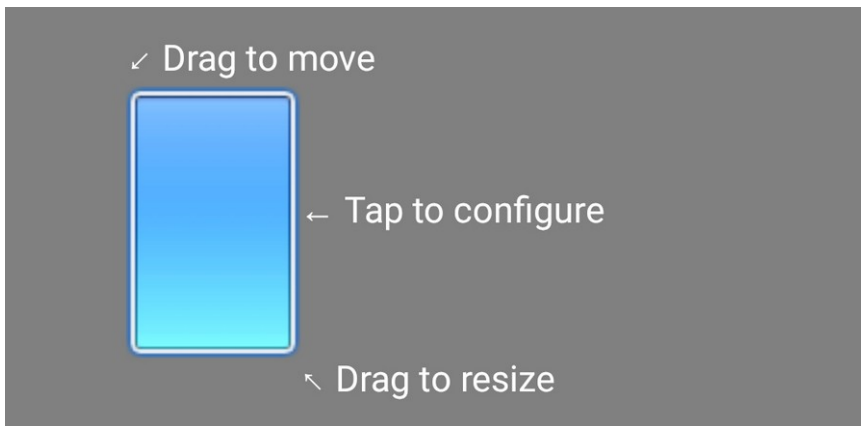


Step 6. Open RoboRemo and tap Menu → CONNECT → Bluetooth (RFCOMM) → “ESP32-RFCOMM”.

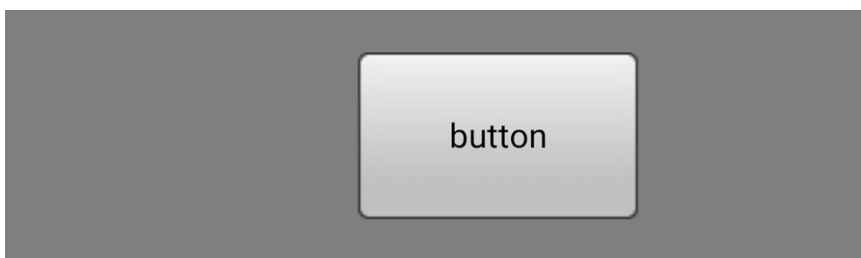


Step 7. Tap Menu → EDIT UI

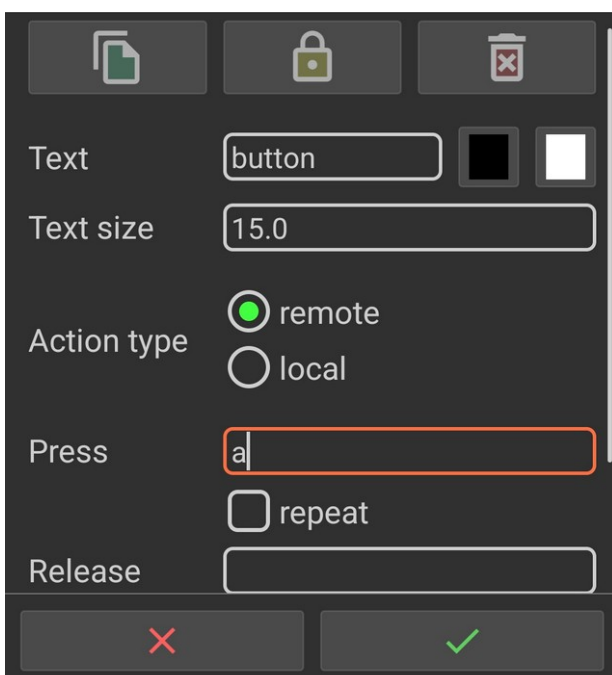
Step 8. Tap on blank space → add Text log. The Text log will appear on screen and you can drag and resize it.



Step 9. Tap on blank space → add Button

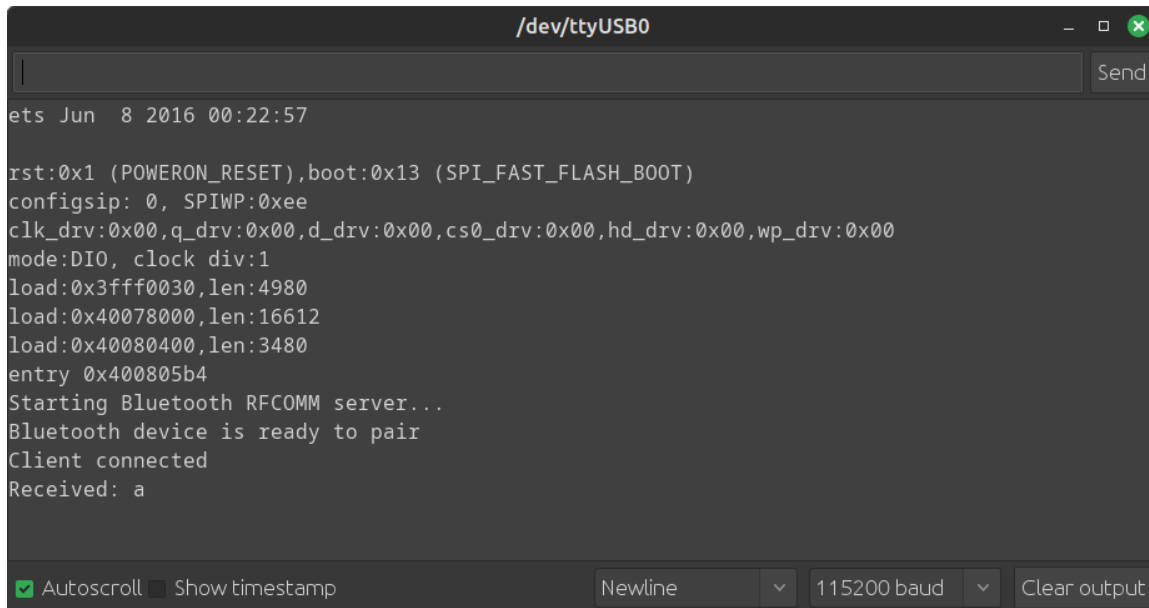


Step 10. Tap on the new button to configure → Press → type “a” → OK (the button will send “a” when pressed)



Step 11. Tap Menu → EXIT THE EDITOR

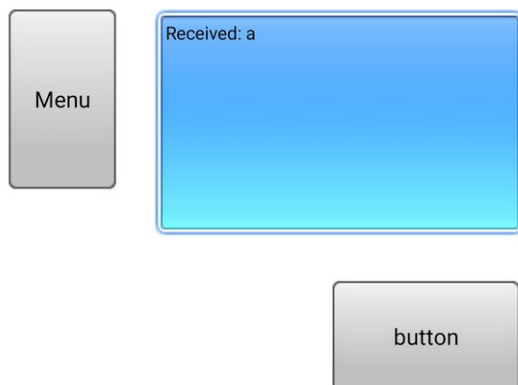
Step 12. Press the button → RoboRemo will send “a\n” to the ESP32. ESP32 will send “Received: a\n” to the Serial Port and it will appear in the Serial Monitor.



```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:4980
load:0x40078000,len:16612
load:0x40080400,len:3480
entry 0x400805b4
Starting Bluetooth RFCOMM server...
Bluetooth device is ready to pair
Client connected
Received: a
```

ESP32 will also send “Received: a\n” back via the Bluetooth RFCOMM connection and “Received: a” will appear in the Text log in RoboRemo interface.



Further development:

ESP32 code can be extended to parse commands from multiple control items in RoboRemo (buttons, sliders, joystick, accelerometer, etc.)

RoboRemo interface can be extended to include multiple indicators, each with its own ID, then ESP32 can send values prefixed by the IDs and update those indicators.

RoboRemo app – free demo version:

<https://play.google.com/store/apps/details?id=com.hardcodedjoy.roboremoofree>

RoboRemo User Manual:

[https://roboremo.app/roboremo\\_app\\_manual.pdf](https://roboremo.app/roboremo_app_manual.pdf)

This project on GitHub:

<https://github.com/hardcodedjoy/esp32-bluetooth-rfcomm-demo>