



## PwnPad Assembly & Firmware Upload Guide

Welcome to the PwnPad Hardware CTF Building Instructions. This guide will walk you through the **hardware assembly, soldering process, and firmware uploading** necessary to get your PwnPad operational.

---

### 🔧 Tools & Materials Required

1. Arduino UNO with DIP-package ATmega328P
2. Soldering iron
3. Solder wire
4. Flux
5. Arduino IDE (download from <https://arduino.cc>)
6. Patience and attention to detail

---

### 🛒 Ordering Your PwnPad PCBs

1. Visit **JLCPCB** at <https://jlpcb.com/>

The screenshot shows the JLCPCB website homepage. A prominent banner at the top left offers a \$30 discount on premium 6-layer PCBs. Below the banner, there's a form to upload a Gerber file, with options to choose the number of layers (set to 2), dimensions (100x100 mm), and quantity (5). An 'Instant Quote' button is visible. The main content area features images of various PCBs and flexible printed circuit boards (FPCs). A news banner at the bottom left mentions a launch of transparent FPCs. A welcome message from a chatbot is visible on the right.

2. Click on “Add Gerber File” and upload `Gerber_PwnPad_v0.10.zip`

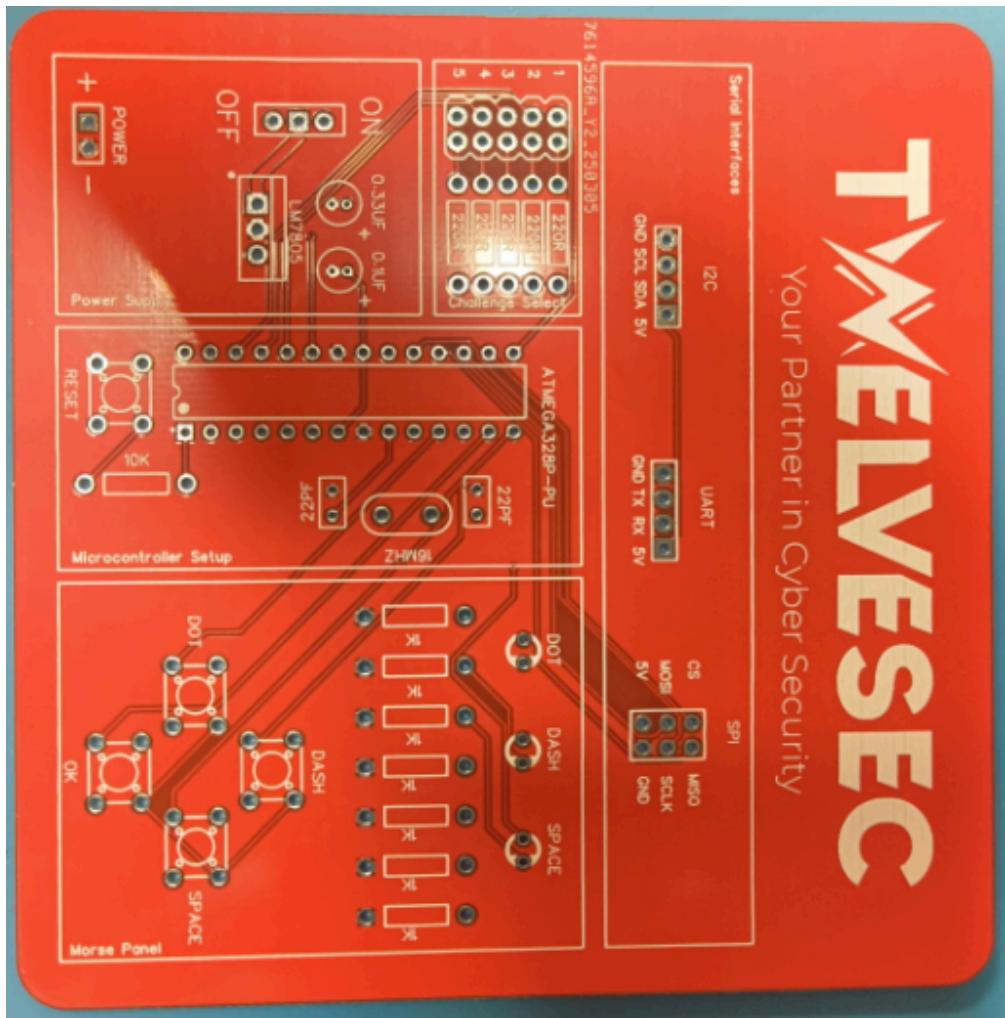
This screenshot shows the same JLCPCB homepage as above, but with a red box highlighting the 'Add gerber file' button in the upload form. The rest of the interface, including the promotional banner, PCB images, and news banner, remains the same.

3. Click “SAVE TO CART” and proceed with the order process

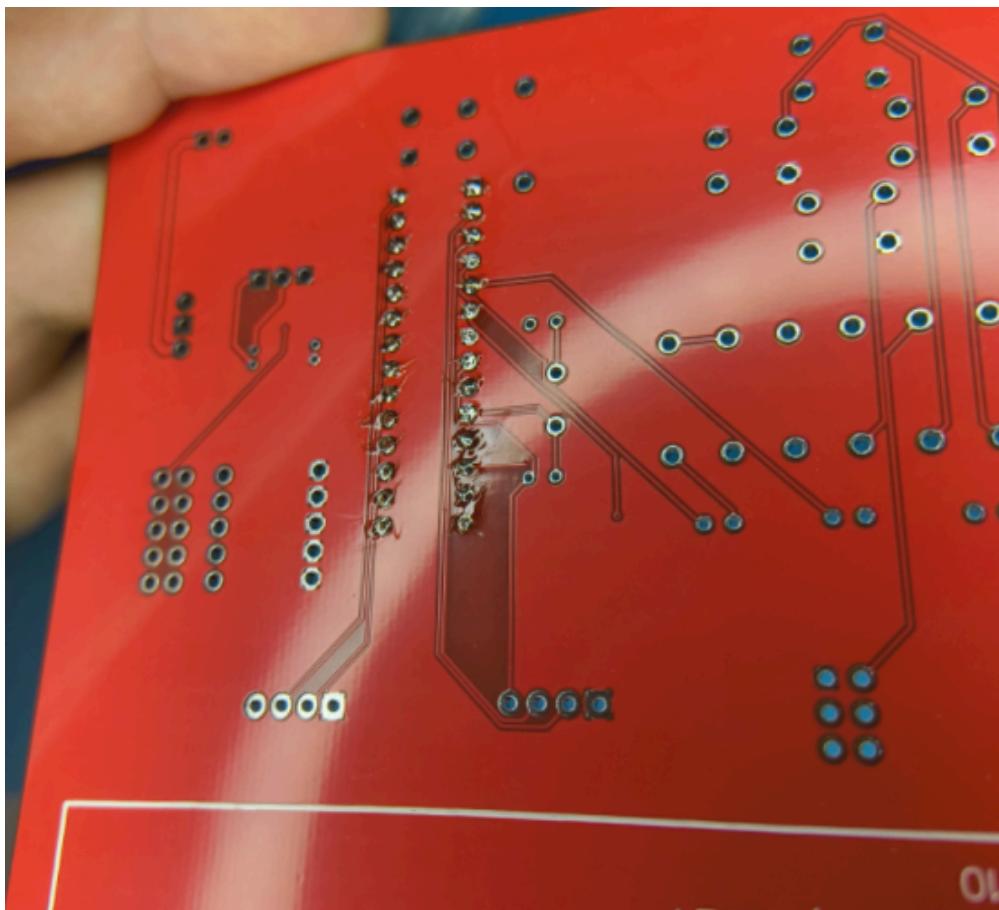
## 🔩 Building Your PwnPad (Soldering Instructions)

1. If you're new to soldering, please watch this introductory video:  
📺 [How to Solder - Beginner Guide](#)
2. Most components used on the PwnPad are **non-polarized**, meaning orientation doesn't matter. However, **some components require correct polarity**:
  - LEDs
  - Power supply components
  - Microcontroller
  - Electrolytic capacitors (⚠ safety critical)

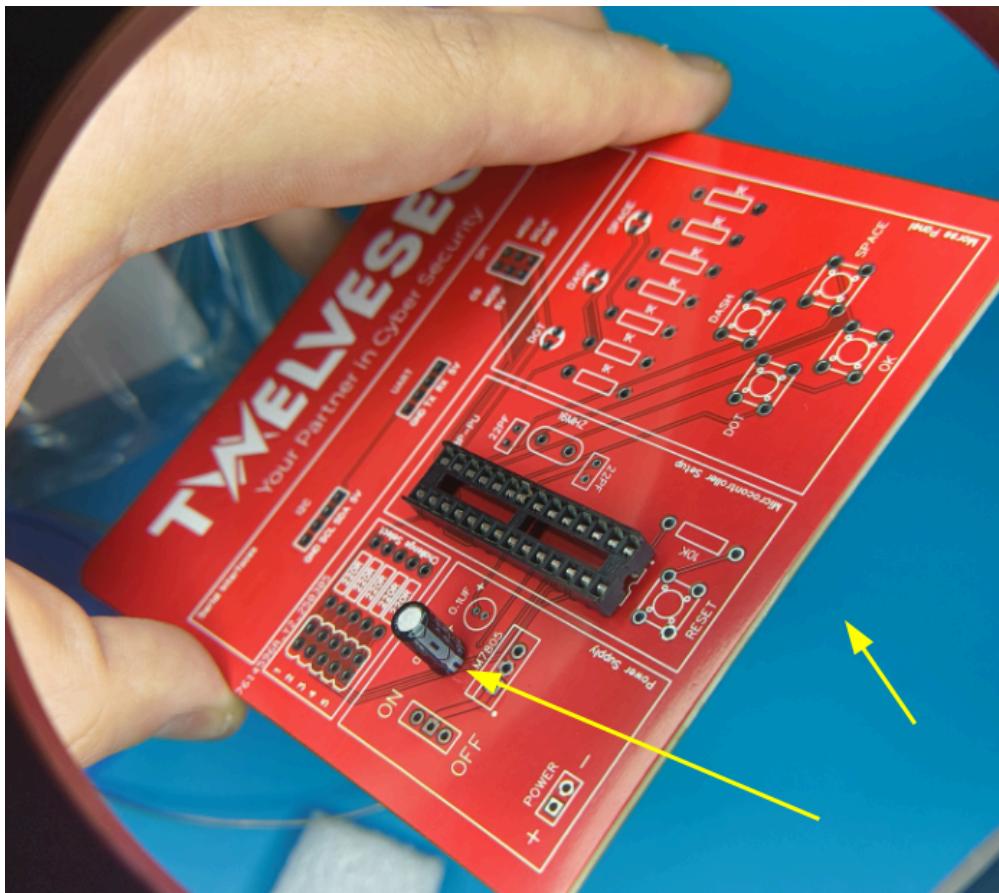
⚠ **Double-check polarity-sensitive components before applying power.** Refer to the silkscreen and diagrams to verify correct orientation.



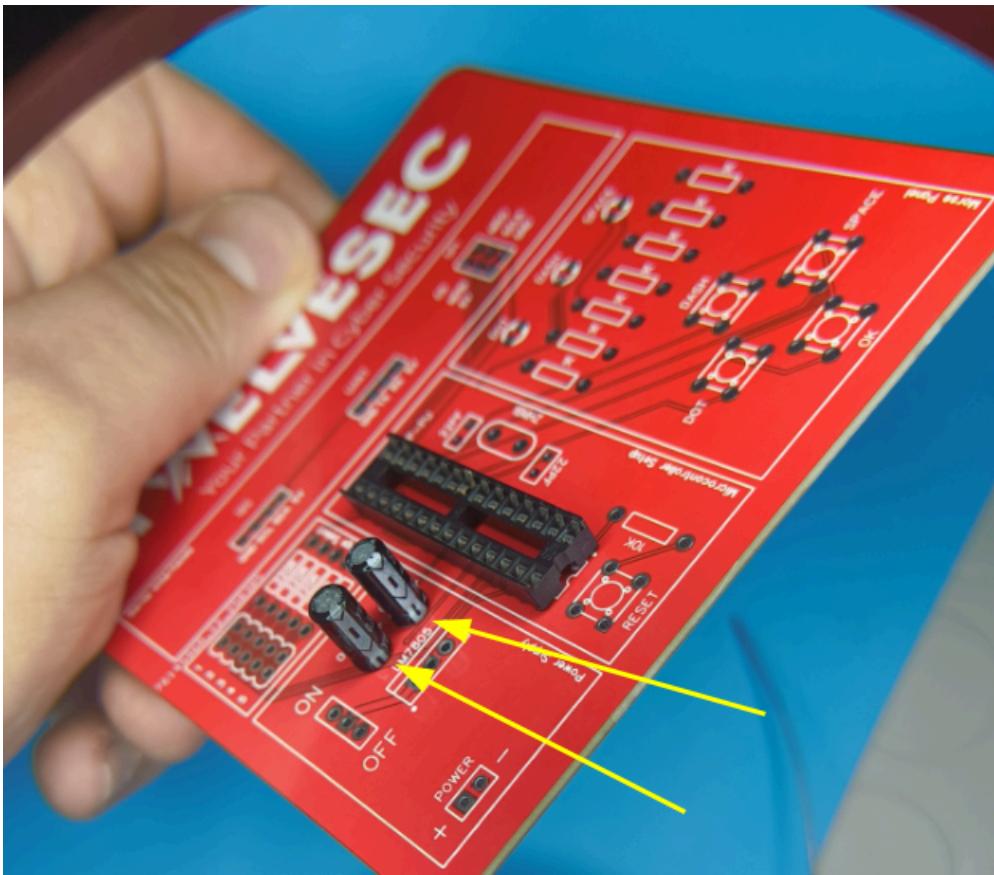
Bare PwnPad PCB



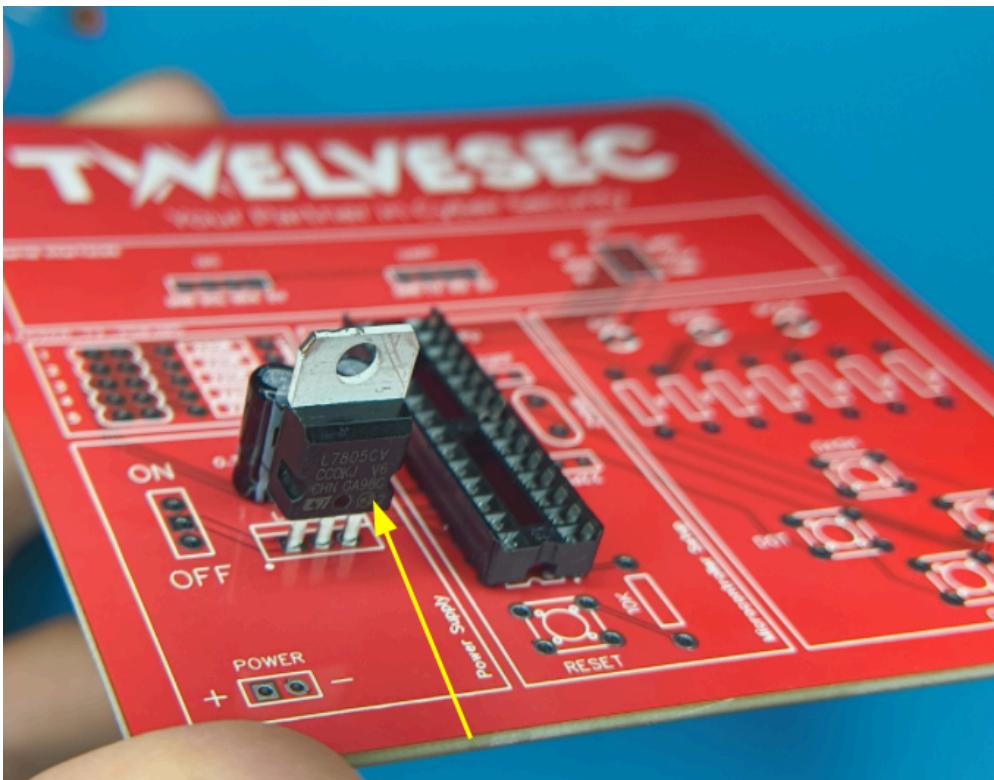
| How soldered joins "should" look like



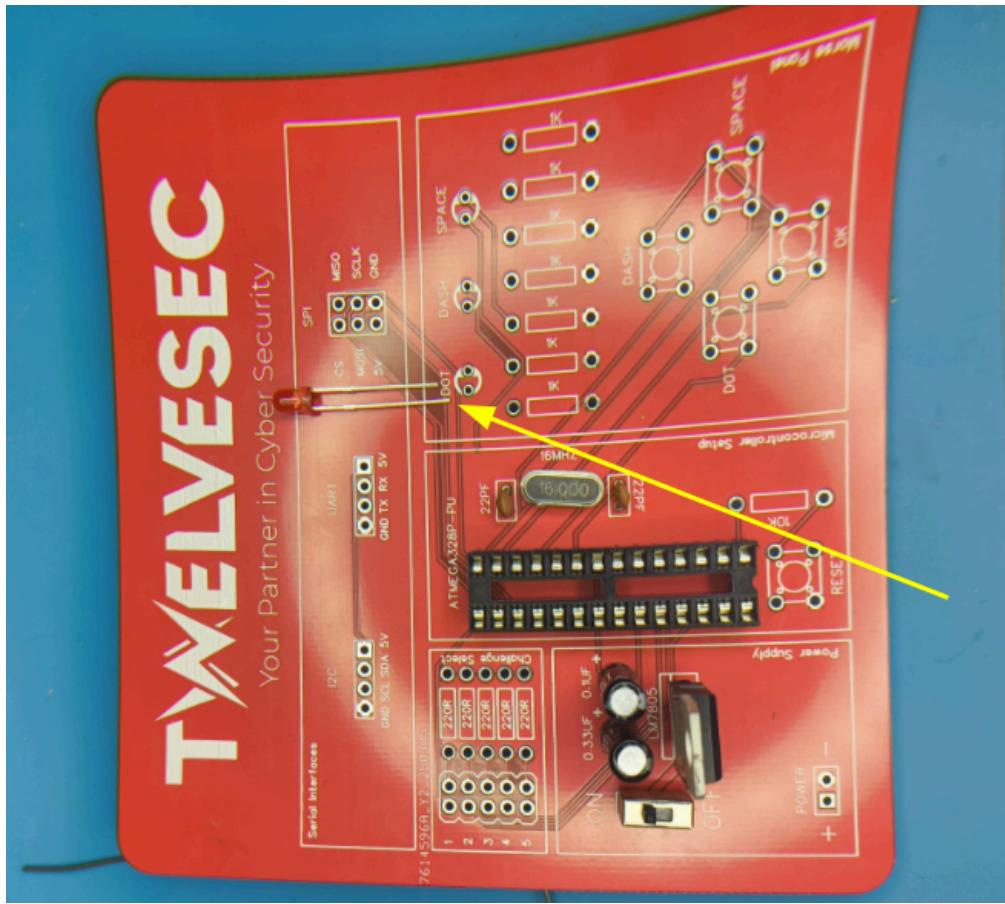
### Capacitor 0.33uF Polarity



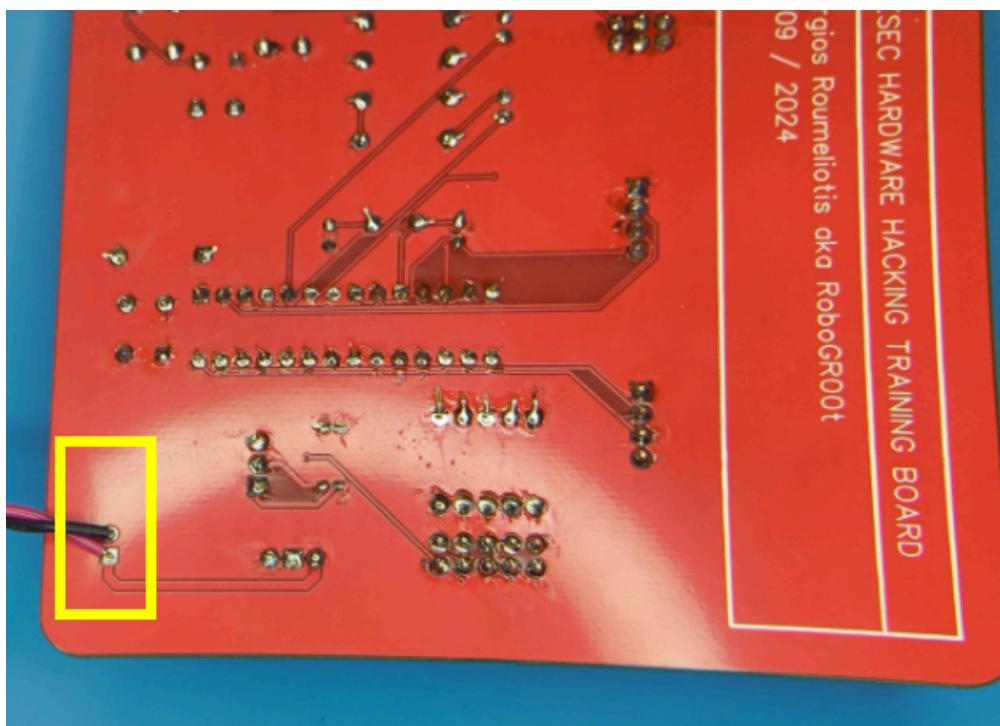
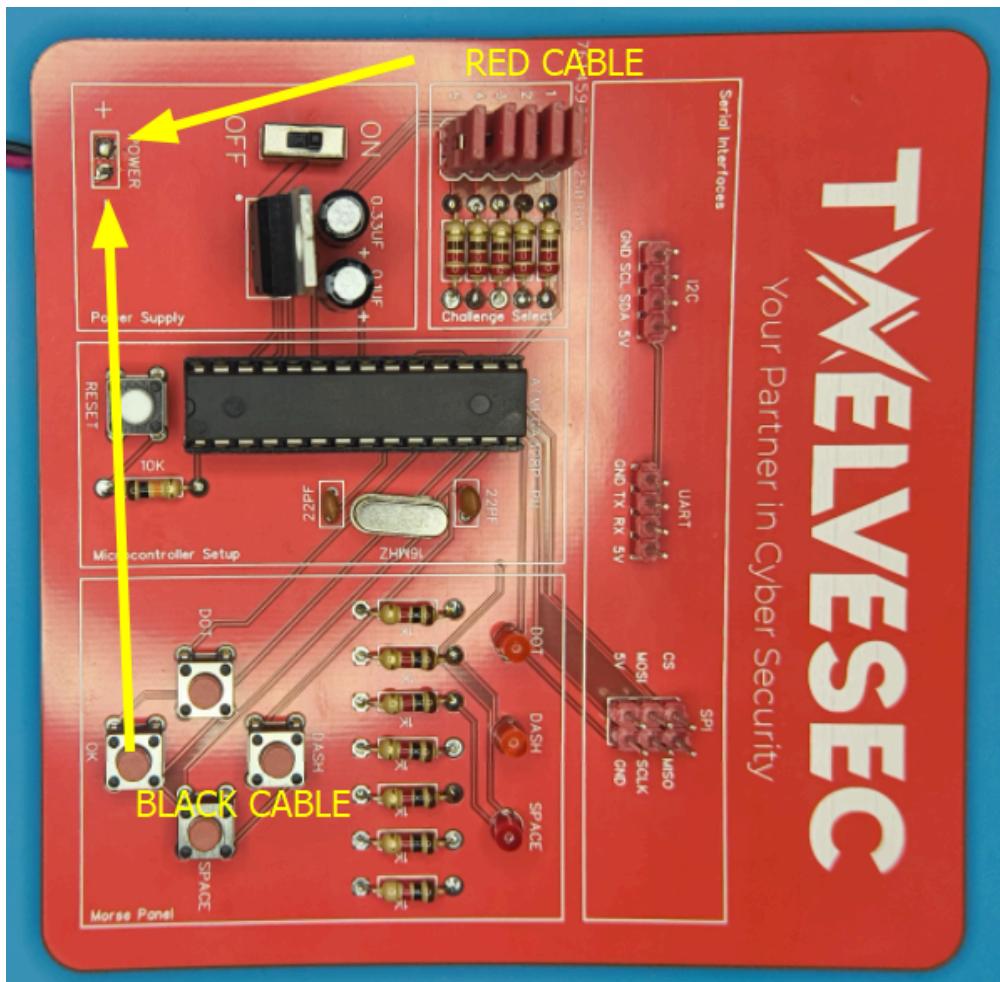
### Capacitor 0.1uF Polarity



### Power Supply Polarity



LEDs Polarity (Long leg is the anode)

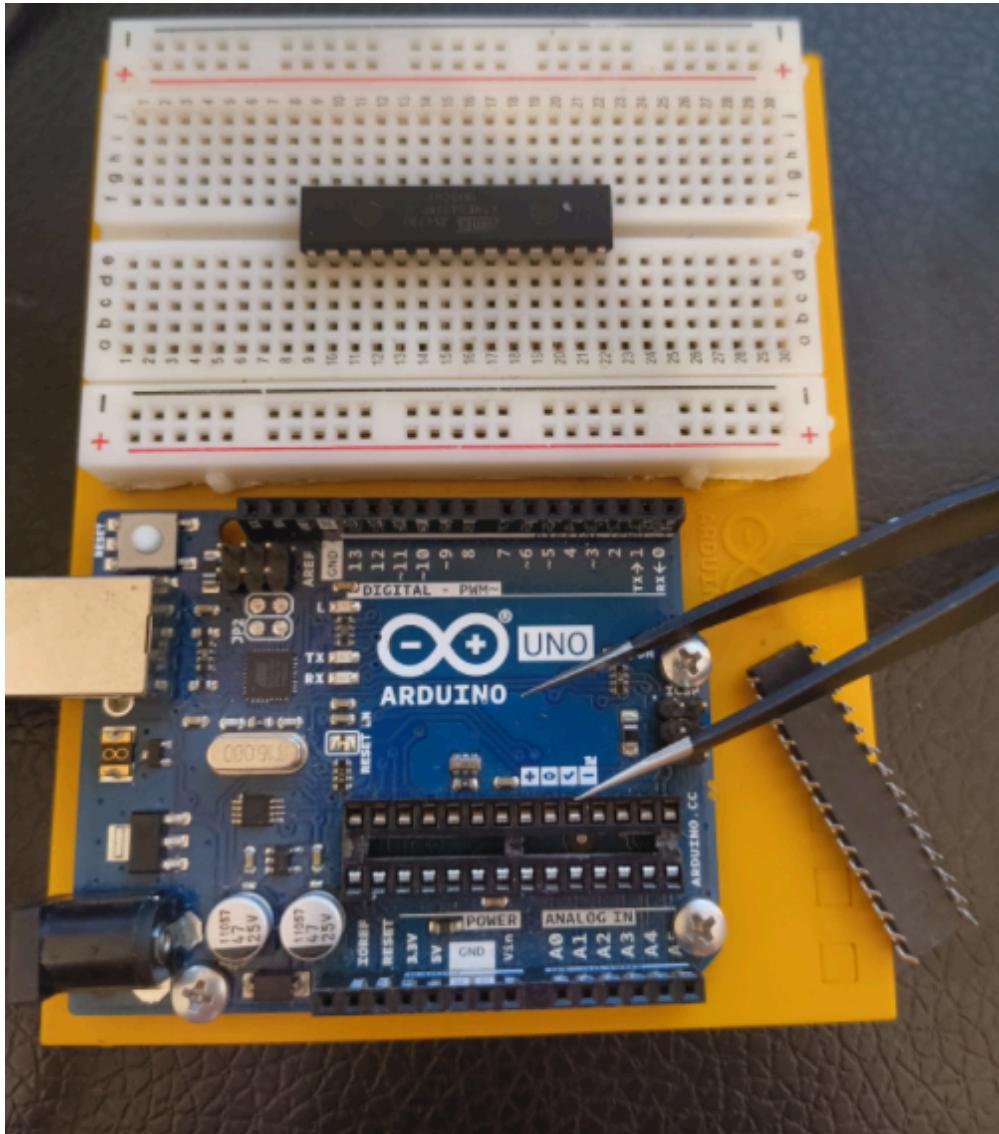


Battery Clip Polarity

# Uploading Firmware to the Microcontroller

## Step 0: Swap the Microcontroller

- Carefully remove the **ATmega328P microcontroller** from your Arduino UNO.
  - Insert the PwnPad's microcontroller into the Arduino socket.
- ⚠ Mind the polarity:** Match the notch or dot with the socket orientation.



---

## Step 1: Launch Arduino IDE

The screenshot shows the Arduino IDE interface with the title bar "sketch\_jun2a | Arduino IDE 2.3.4". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for Save, Undo, Redo, and a dropdown for Board (set to Arduino Uno). The code editor contains the following sketch:

```
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8
9 }
10
```

The status bar at the bottom right shows "Ln 1, Col 1 Arduino Uno on COM5 [not connected]".

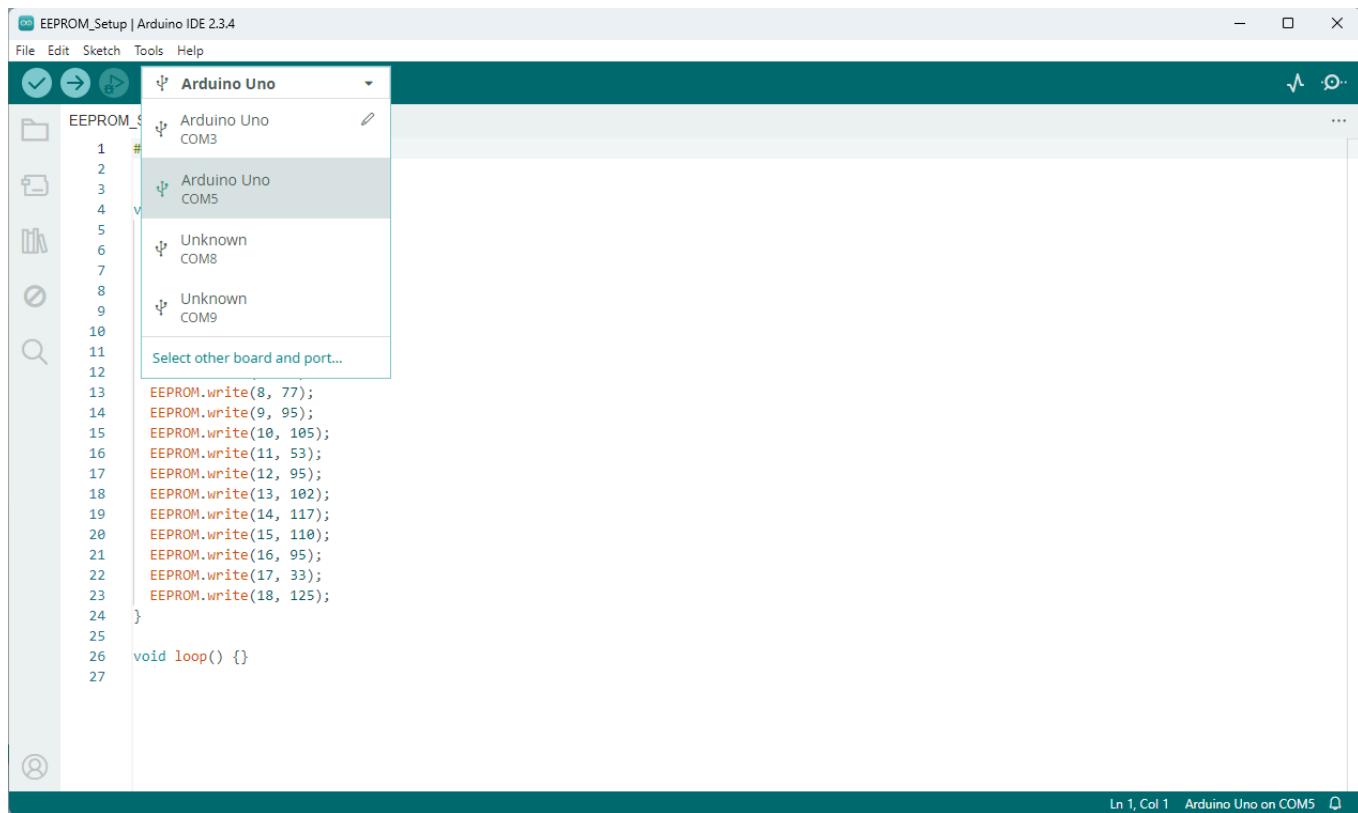
## Step 2: Open EEPROM\_Setup.ino

The screenshot shows the Arduino IDE interface with the title bar "EEPROM\_Setup | Arduino IDE 2.3.4". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for Save, Undo, Redo, and a dropdown for Board (set to Arduino Uno). The code editor contains the following sketch:

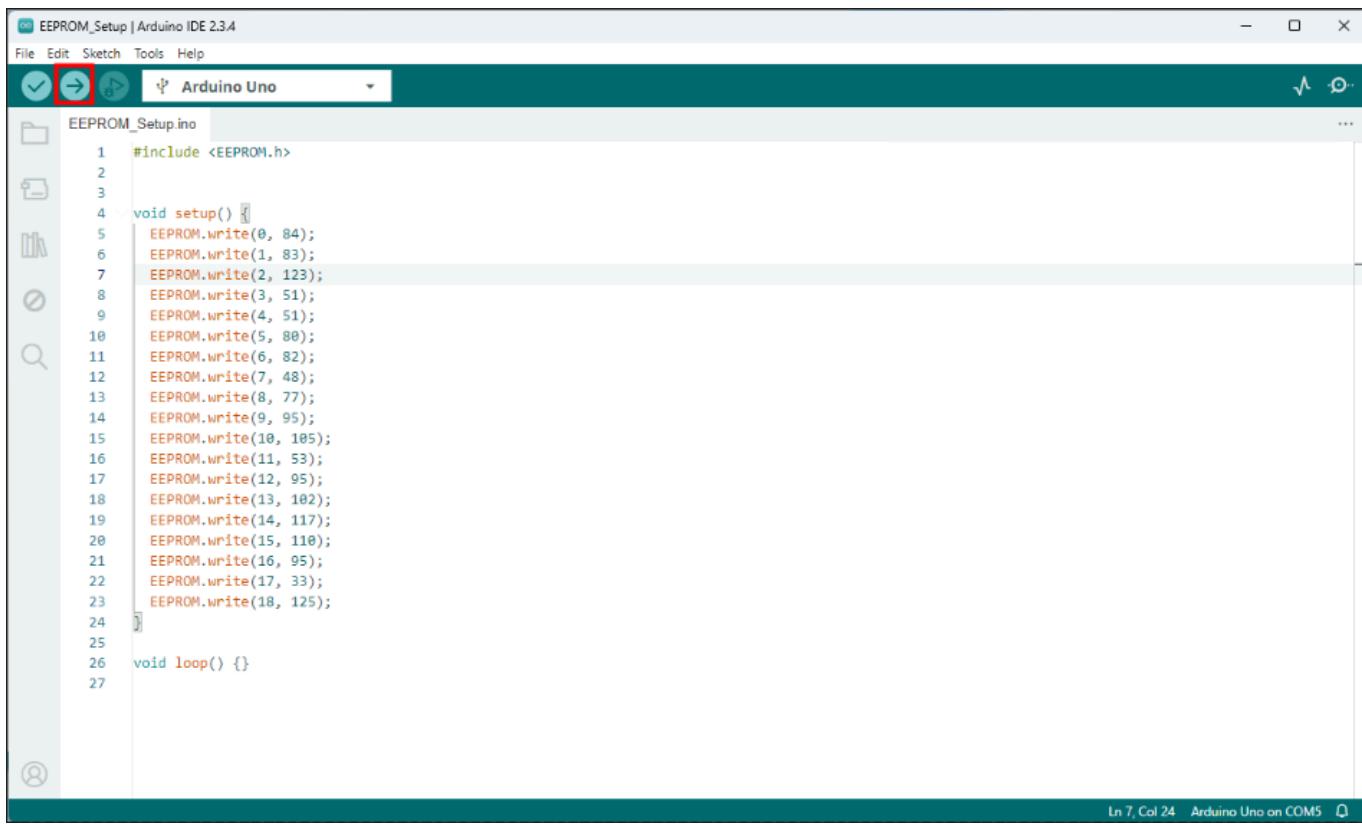
```
1 #include <EEPROM.h>
2
3
4 void setup() {
5     EEPROM.write(0, 84);
6     EEPROM.write(1, 83);
7     EEPROM.write(2, 123);
8     EEPROM.write(3, 51);
9     EEPROM.write(4, 51);
10    EEPROM.write(5, 80);
11    EEPROM.write(6, 82);
12    EEPROM.write(7, 48);
13    EEPROM.write(8, 77);
14    EEPROM.write(9, 95);
15    EEPROM.write(10, 105);
16    EEPROM.write(11, 53);
17    EEPROM.write(12, 95);
18    EEPROM.write(13, 102);
19    EEPROM.write(14, 117);
20    EEPROM.write(15, 110);
21    EEPROM.write(16, 95);
22    EEPROM.write(17, 33);
23    EEPROM.write(18, 125);
24
25
26 void loop() {}
```

The status bar at the bottom right shows "Ln 7, Col 24 Arduino Uno on COM5".

## Step 3: Select the Correct COM Port



## Step 4: Compile and Upload EEPROM\_Setup.ino

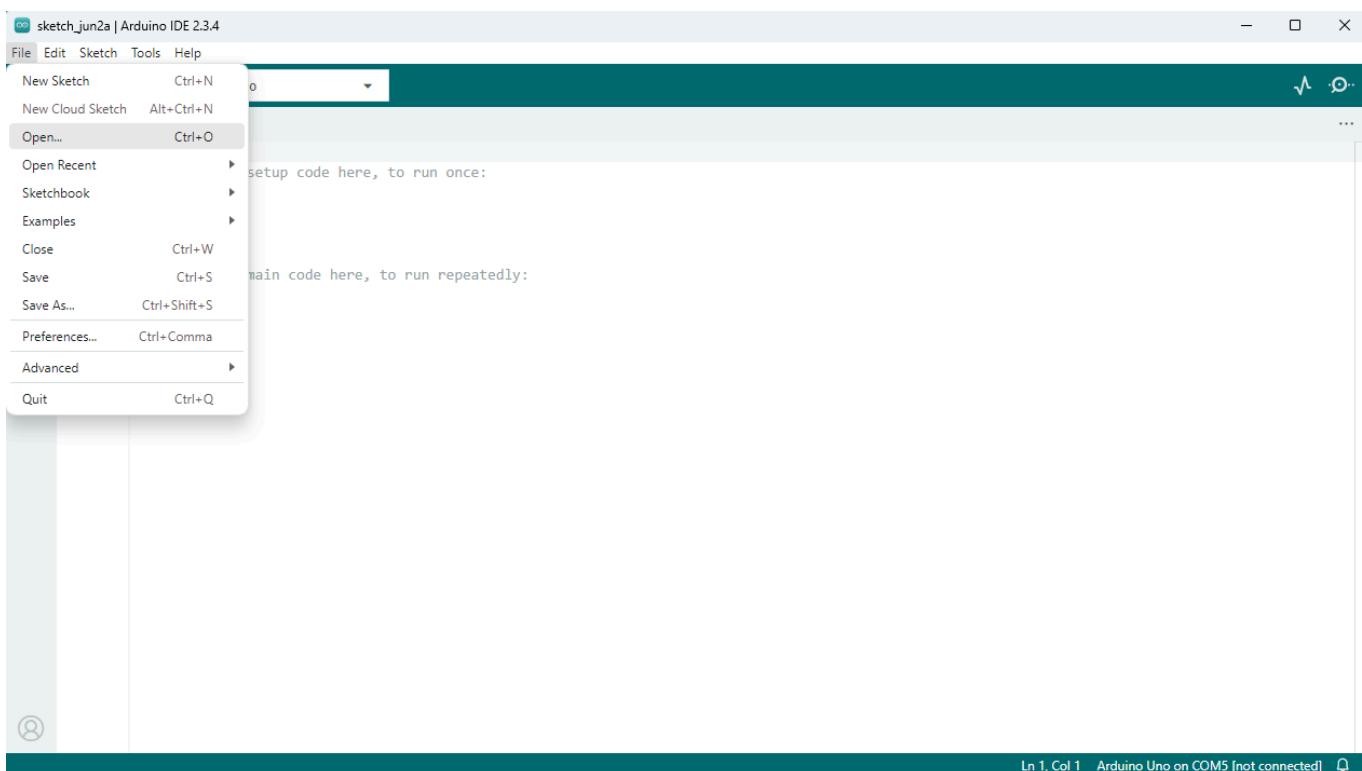


The screenshot shows the Arduino IDE interface with the title bar "EEPROM\_Setup | Arduino IDE 2.3.4". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for file operations and a play button, with the play button highlighted by a red box. The board selection dropdown shows "Arduino Uno". The code editor displays "EEPROM\_Setup.ino" with the following content:

```
1 #include <EEPROM.h>
2
3
4 void setup() {
5     EEPROM.write(0, 84);
6     EEPROM.write(1, 83);
7     EEPROM.write(2, 123);
8     EEPROM.write(3, 51);
9     EEPROM.write(4, 51);
10    EEPROM.write(5, 80);
11    EEPROM.write(6, 82);
12    EEPROM.write(7, 48);
13    EEPROM.write(8, 77);
14    EEPROM.write(9, 95);
15    EEPROM.write(10, 185);
16    EEPROM.write(11, 53);
17    EEPROM.write(12, 95);
18    EEPROM.write(13, 182);
19    EEPROM.write(14, 117);
20    EEPROM.write(15, 110);
21    EEPROM.write(16, 95);
22    EEPROM.write(17, 33);
23    EEPROM.write(18, 125);
24 }
25
26 void loop() {}
27
```

The status bar at the bottom right shows "Ln 7, Col 24 Arduino Uno on COM5".

## Step 5: Open PwnPad.ino



sketch\_jun2a | Arduino IDE 2.3.4

File Edit Sketch Tools Help

New Sketch Ctrl+N  
New Cloud Sketch Alt+Ctrl+N  
Open... Ctrl+O  
Open Recent ▾  
Sketchbook ▾  
Examples ▾  
Close Ctrl+W  
Save Ctrl+S  
Save As... Ctrl+Shift+S  
Preferences... Ctrl+Comma  
Advanced ▾  
Quit Ctrl+Q

setup code here, to run once:

main code here, to run repeatedly:

PwnPad | Arduino IDE 2.3.4

File Edit Sketch Tools Help

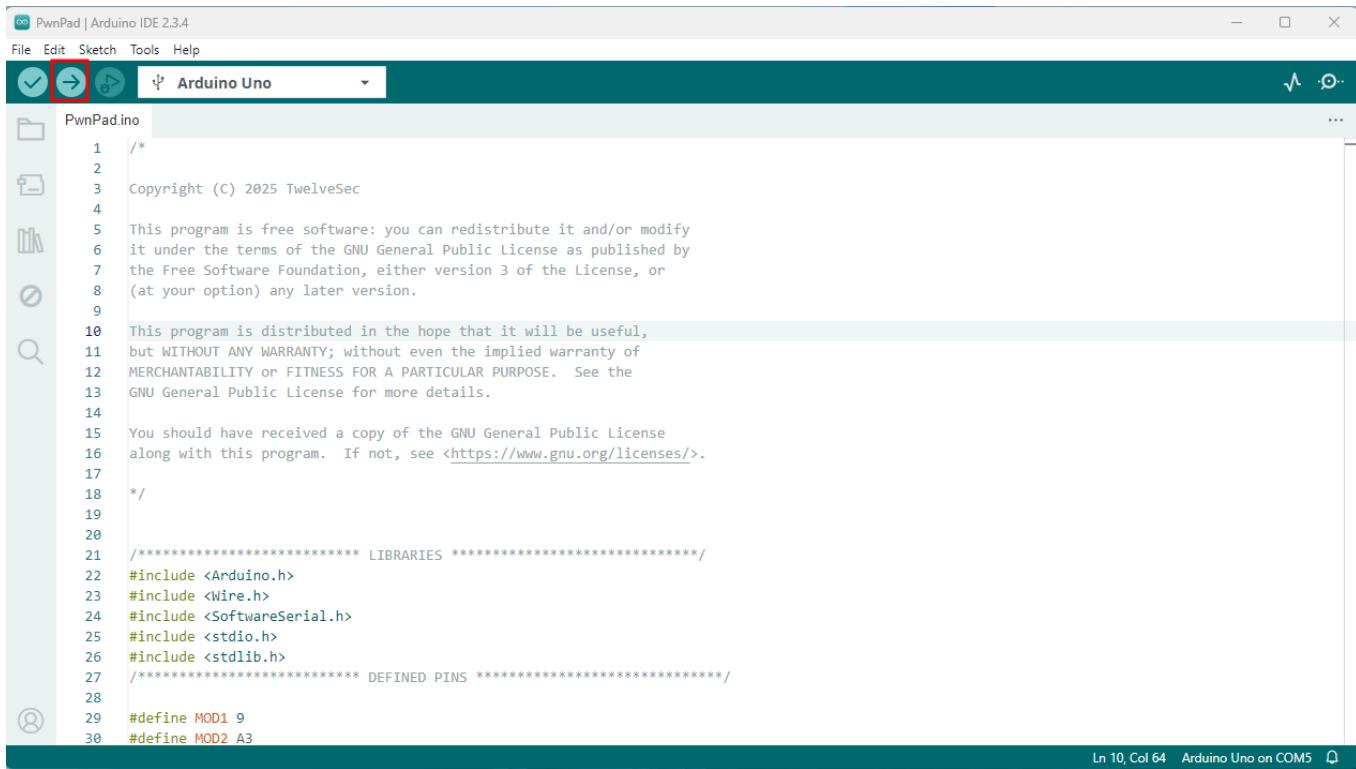
Arduino Uno

PwnPad.ino

```
1  /*
2
3  Copyright (C) 2025 TwelveSec
4
5  This program is free software: you can redistribute it and/or modify
6  it under the terms of the GNU General Public License as published by
7  the Free Software Foundation, either version 3 of the License, or
8  (at your option) any later version.
9
10 This program is distributed in the hope that it will be useful,
11 but WITHOUT ANY WARRANTY; without even the implied warranty of
12 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 GNU General Public License for more details.
14
15 You should have received a copy of the GNU General Public License
16 along with this program. If not, see <https://www.gnu.org/licenses/>.
17
18 */
19
20
21 //***** LIBRARIES *****/
22 #include <Arduino.h>
23 #include <Wire.h>
24 #include <SoftwareSerial.h>
25 #include <stdio.h>
26 #include <stdlib.h>
27 //***** DEFINED PINS *****/
28
29 #define MOD1 9
30 #define MOD2 A3
```

Ln 1, Col 1 Arduino Uno on COM5 [not connected]

## Step 6: Compile and Upload PwnPad.ino



```
PwnPad | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Arduino Uno
PwnPad.ino
1 /*
2
3 Copyright (C) 2025 TwelveSec
4
5 This program is free software: you can redistribute it and/or modify
6 it under the terms of the GNU General Public License as published by
7 the Free Software Foundation, either version 3 of the License, or
8 (at your option) any later version.
9
10 This program is distributed in the hope that it will be useful,
11 but WITHOUT ANY WARRANTY; without even the implied warranty of
12 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 GNU General Public License for more details.
14
15 You should have received a copy of the GNU General Public License
16 along with this program. If not, see <https://www.gnu.org/licenses/>.
17
18 */
19
20
21 //***** LIBRARIES *****/
22 #include <Arduino.h>
23 #include <Wire.h>
24 #include <SoftwareSerial.h>
25 #include <stdio.h>
26 #include <stdlib.h>
27 //***** DEFINED PINS *****/
28
29 #define MOD1 9
30 #define MOD2 A3
Ln 10, Col 64 Arduino Uno on COM5
```

## Step 7: Final Microcontroller Transfer

1. Disconnect the Arduino UNO from power.
2. Using tweezers, **carefully remove the microcontroller** from the Arduino.
3. Insert the now-programmed microcontroller into your assembled PwnPad board.
4. Reinstall the original microcontroller back into the Arduino UNO.

---

Congratulations! 🎉 Your **PwnPad** is now built and ready for use in your Hardware CTF challenges.