

Assembly

1. Print the Enclosure

- **Filament Required:** Approximately 50 grams
- **Estimated Print Time:** About 2 hours at 0.2 mm layer height

2. Print the Button

- **Multicolor Version:** Uses around 40 grams of filament and takes about 1.5 hours
- **Single-Color Version:** Generally faster print time than multicolor

3. Wire the Keyboard Switch

1. Solder two wires to the keyboard switch terminals.
2. Connect one wire to Arduino GND and the other wire to Arduino PB0.

4. Install the Mechanical Key Switch & Arduino

1. Snap-fit the switch into the designated slot in the printed enclosure.
2. Slide the Arduino ATtiny85 (Digispark) board into its rail.
3. Insert the 3D-printed springs (add a small amount of glue if they feel loose).
4. Align the cable notch and gently press the 3D-printed button on top.
5. Place the enclosure ring from the top and snap-fit it into place.
6. Refer to any assembly pictures or diagrams for additional guidance.



5. Flash the ATtiny85 → see next chapter

Installing and Setting Up Arduino for Digispark ATtiny85

1. Download and Install Arduino IDE

- Visit the official Arduino website: [Arduino Software](https://www.arduino.cc/en/software)
 - Download the installer for your operating system (Windows, Mac, or Linux).
 - Run the installer and follow the on-screen instructions to complete the installation.
-

2. Install the Digistump Drivers (Windows)

- Locate the Digistump_DRIVER.zip file.
- Extract (unzip) the contents to a folder on your computer.
- Inside the extracted folder, run DPinst64.exe (for 64-bit systems) or DPinst.exe (for 32-bit systems).

Name	Änderungsdatum	Typ	Größe
amd64	12.10.2016 03:07	Dateiordner	
x86	12.10.2016 03:07	Dateiordner	
cdc_digix.cat	08.04.2016 14:21	Sicherheitskatalog	8 KB
ChangeCDCSpeed.vbs	08.04.2016 14:21	VBScript-Skriptdatei	2 KB
digiserial.cat	03.09.2014 23:45	Sicherheitskatalog	9 KB
DigiSerial.inf	03.09.2014 23:42	Setup-Informatio...	3 KB
Digispark_Bootloader.cat	08.04.2016 14:21	Sicherheitskatalog	10 KB
Digispark_Bootloader.inf	08.04.2016 14:21	Setup-Informatio...	9 KB
digiusb.cat	08.04.2016 14:21	Sicherheitskatalog	11 KB
DigiUSB.inf	08.04.2016 14:21	Setup-Informatio...	8 KB
DigiX.inf	08.04.2016 14:21	Setup-Informatio...	4 KB
DPinst.exe	08.04.2016 14:21	Anwendung	901 KB
DPinst64.exe	08.04.2016 14:21	Anwendung	1.023 KB
Install Drivers.exe	08.04.2016 14:21	Anwendung	1.487 KB
launcher.exe	08.04.2016 14:21	Anwendung	1.416 KB
micronucleus.exe	08.04.2016 14:21	Anwendung	82 KB
post_install.bat	08.04.2016 14:21	Windows-Batchda...	1 KB

- Follow any on-screen prompts to install the driver.

Note: On Windows 10 or higher, sometimes the drivers install automatically when you plug in the device. However, if it doesn't install automatically, follow the steps above.

3. Copy the Arduino Sketch to Your Arduino Folder

- Copy the chadGPT_attiny85.ino file into your Arduino sketch folder, typically found at:

C:\Users\YOURUSERNAME\Documents\Arduino

- Ensure the .ino file is in a folder named chadGPT_attiny85 or something descriptive, as the Arduino IDE generally expects the sketch file to be in a folder of the same name.

4. Open the chadGPT_attiny85.ino Sketch

- Launch the Arduino IDE.
- Go to **File > Open**, and navigate to the folder where you placed chadGPT_attiny85.ino.
- Select and open the sketch.

5. Add the Additional Board Manager URL

- In the Arduino IDE, go to **File > Preferences**.
- In the **Additional Board Manager URLs** field, add the following URL:

http://drazzy.com/package_drazzy.com_index.json

- Click **OK** to save.

5.1 Alternative Method (If the URL Is Inactive or Not Working)

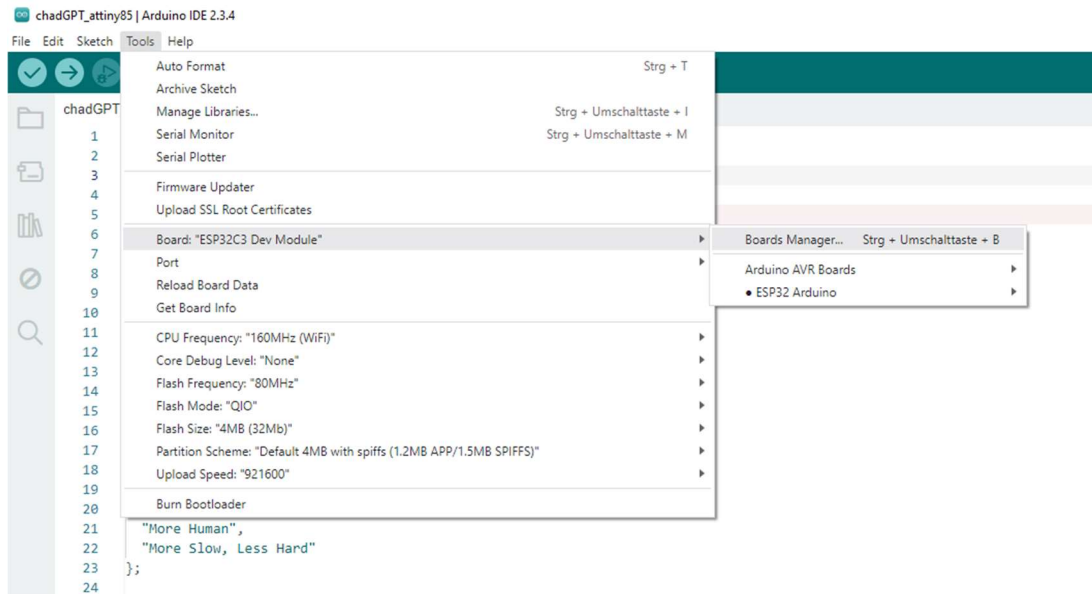
- If for any reason the above URL is no longer active:
 - Locate your downloaded ATTinyCore_BOARDLIBRARY.zip.
 - Extract (unzip) it.
 - Copy the unzipped ATTinyCore folder into:

C:\Users\YOURUSERNAME\AppData\Local\Arduino15\packages

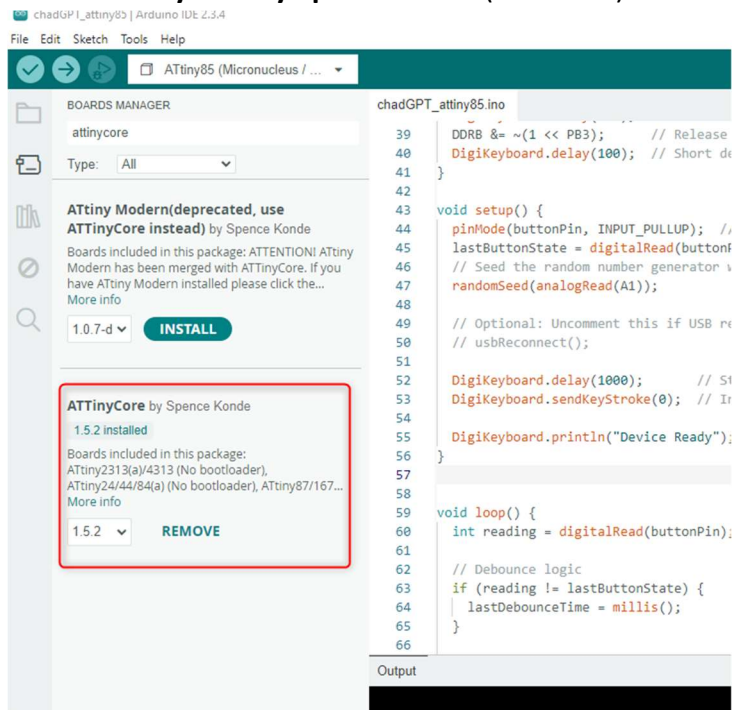
- This manually installs the ATTinyCore package without using the online Boards Manager URL.
-

6. Install ATTinyCore Via the Boards Manager

- In the Arduino IDE, go to **Tools > Board > Boards Manager**.



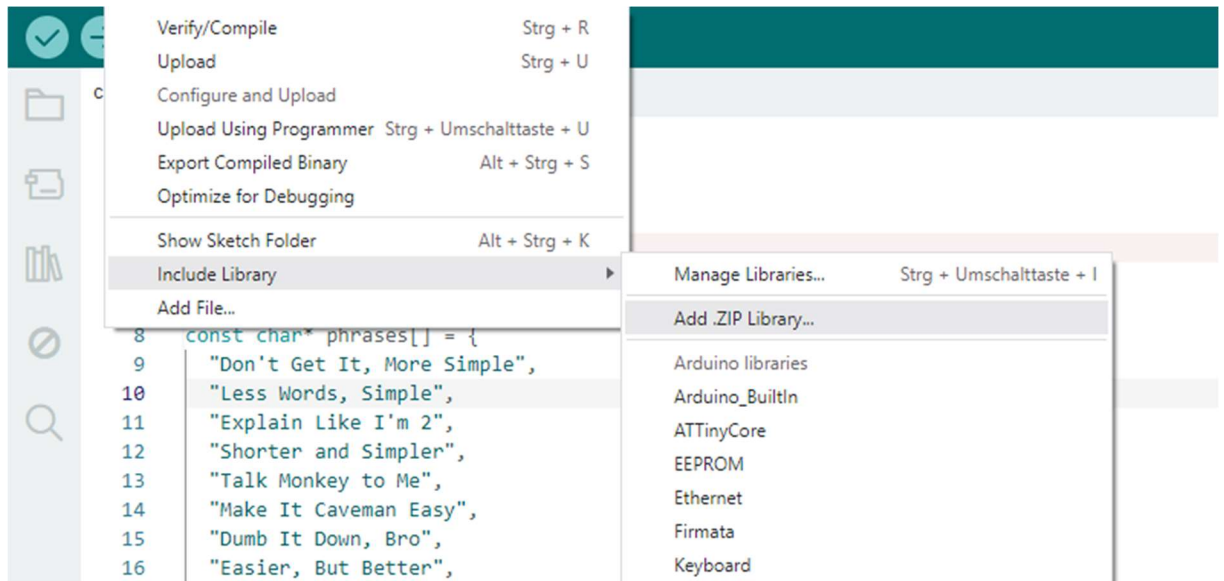
- In the search bar, type **ATTinyCore**.
- Locate **ATTinyCore by SpenceKonde** (or similar) in the search results.



- Click **Install** for version **1.5.2** (or the latest version available).

7. Install the DigisparkKeyboard Library

- In the Arduino IDE, go to **Sketch > Include Library > Add .ZIP Library...**



- Navigate to the DigisparkKeyboard-master_LIBRARY.zip file and select it.
- The library will install automatically.

7.1 Alternative Method (If the ZIP Install Fails)

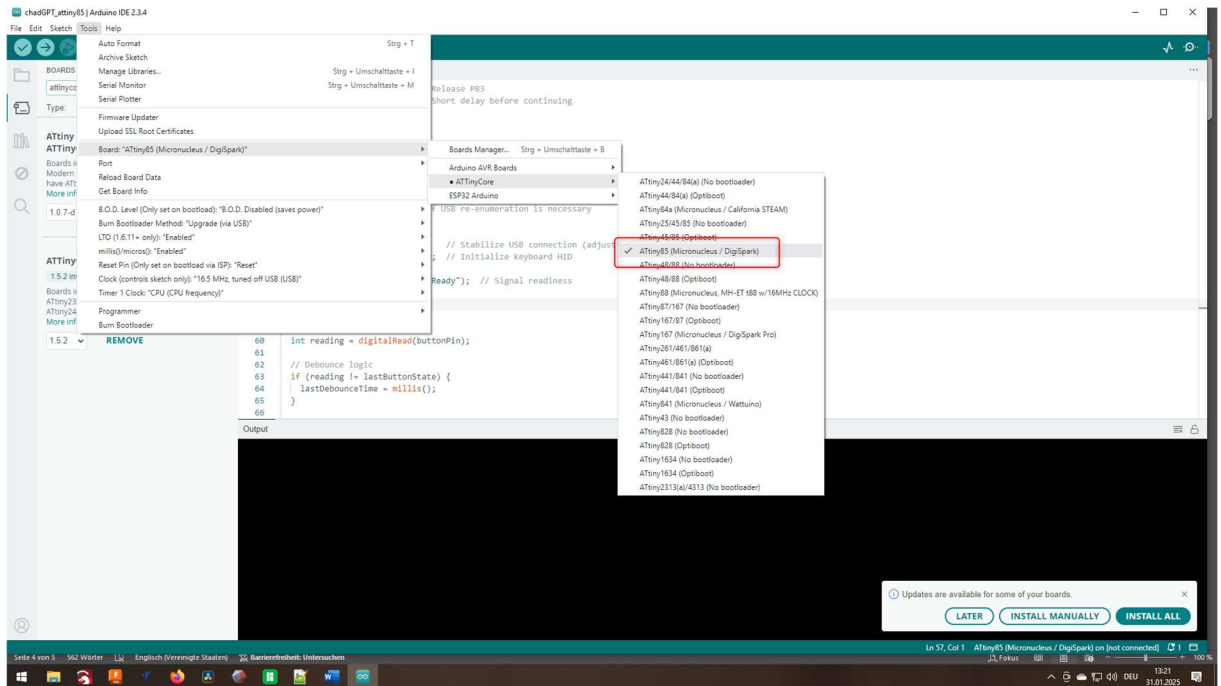
- If the Arduino IDE fails to install from the ZIP:
 - Unzip the contents of DigisparkKeyboard-master_LIBRARY.zip.
 - Copy the extracted folder into:

C:\Users\YOURUSERNAME\Documents\Arduino\libraries\DigisparkKeyboard-master

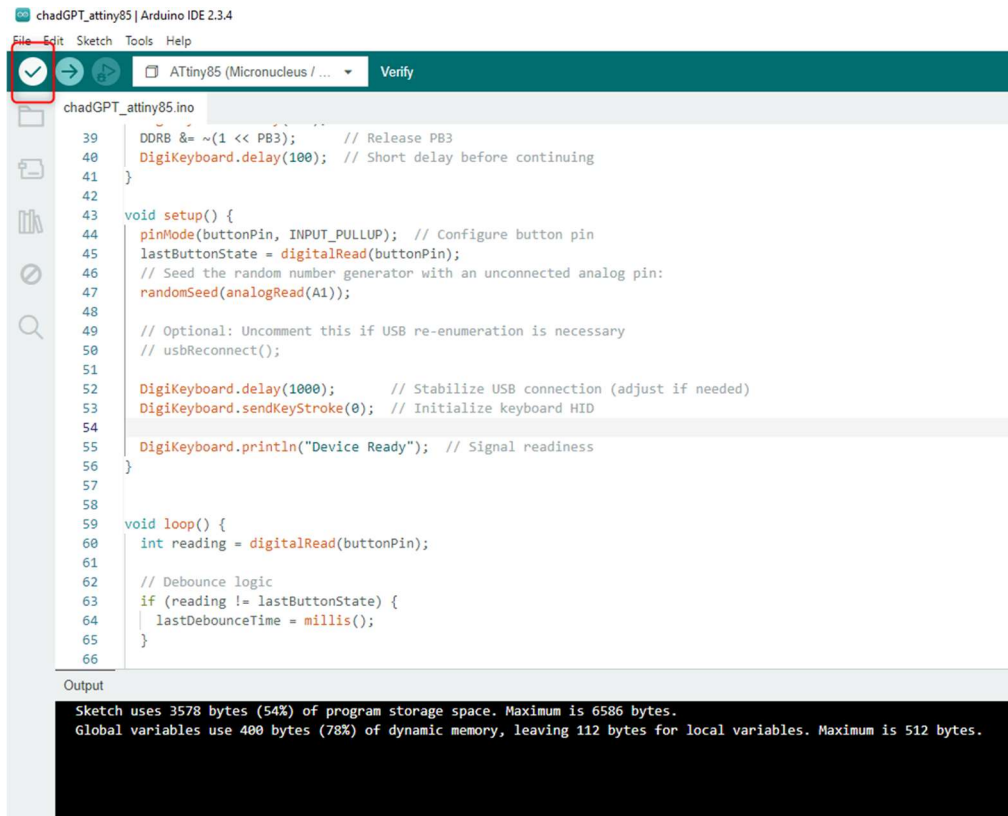
- Restart the Arduino IDE if necessary, and the library should be recognized.

8. Compile the Sketch

- In the Arduino IDE, ensure the correct board is selected under **Tools > Board**.



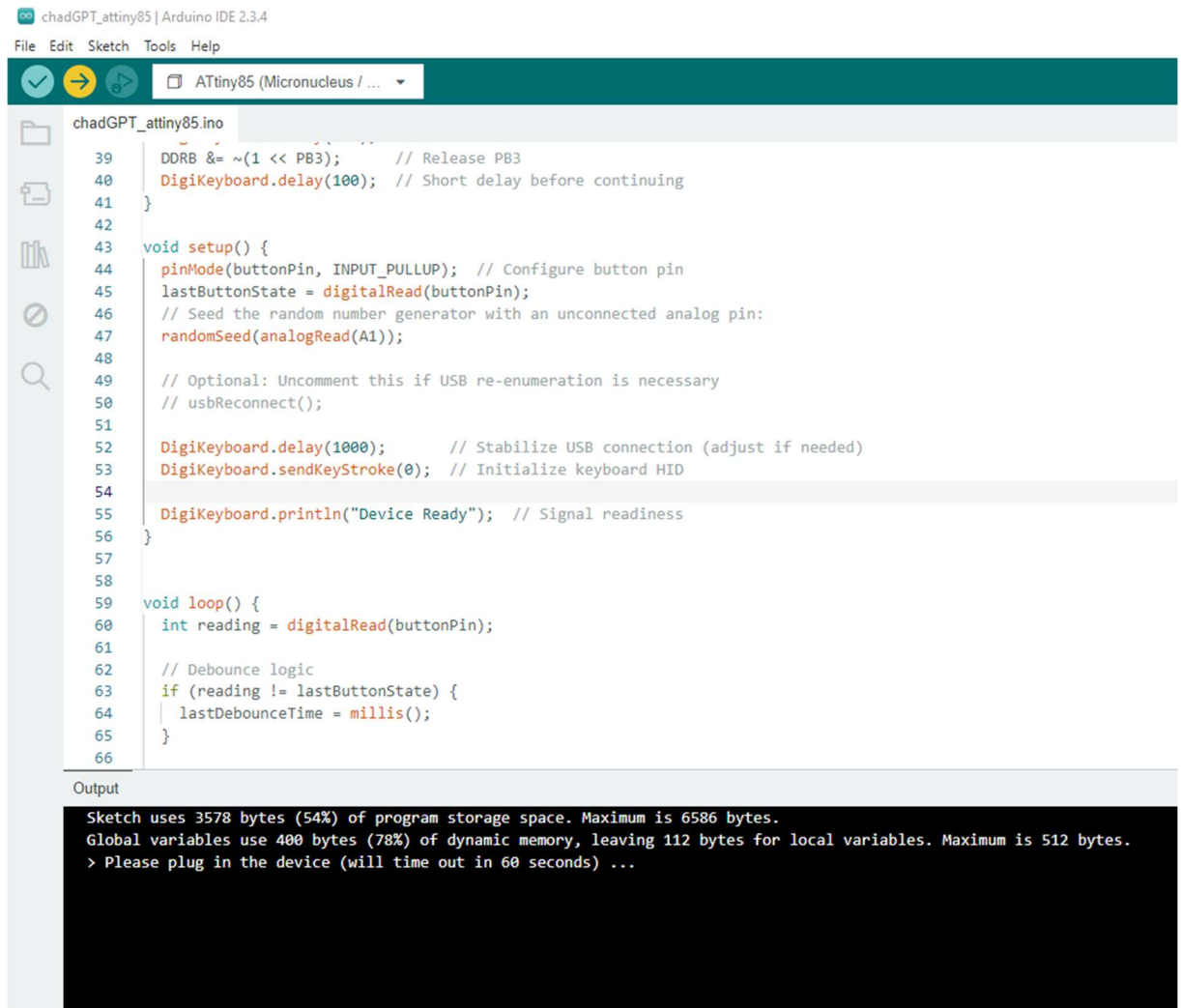
- Since you're using **Digispark/ATtiny85**, select the appropriate ATtiny85 settings provided by ATtinyCore or the Digispark profile.
- Click the **Verify (✓)** button (or go to **Sketch > Verify/Compile**) to compile your sketch.



- Watch the status window for any errors. If it compiles successfully, move on to the upload step.

9. Upload the Code to the Digispark

- Click the **Upload (→) button** in the Arduino IDE.
- Wait for the IDE to display a message instructing you to **“Plug in device now...”** or similar.



The screenshot shows the Arduino IDE interface. The top bar indicates the board is set to ATtiny85 (Micronucleus / ...). The main editor displays the code for chadGPT_attiny85.ino, which includes setup and loop functions for a button and a DigiKeyboard. The output window at the bottom shows the compilation status and a prompt to plug in the device.

```
39  DDRB &= ~(1 << PB3);    // Release PB3
40  DigiKeyboard.delay(100); // Short delay before continuing
41  }
42
43  void setup() {
44    pinMode(buttonPin, INPUT_PULLUP); // Configure button pin
45    lastButtonState = digitalRead(buttonPin);
46    // Seed the random number generator with an unconnected analog pin:
47    randomSeed(analogRead(A1));
48
49    // Optional: Uncomment this if USB re-enumeration is necessary
50    // usbReconnect();
51
52    DigiKeyboard.delay(1000);    // Stabilize USB connection (adjust if needed)
53    DigiKeyboard.sendKeyStroke(0); // Initialize keyboard HID
54
55    DigiKeyboard.println("Device Ready"); // Signal readiness
56  }
57
58
59  void loop() {
60    int reading = digitalRead(buttonPin);
61
62    // Debounce logic
63    if (reading != lastButtonState) {
64      lastDebounceTime = millis();
65    }
66  }
```

Output

```
Sketch uses 3578 bytes (54%) of program storage space. Maximum is 6586 bytes.
Global variables use 400 bytes (78%) of dynamic memory, leaving 112 bytes for local variables. Maximum is 512 bytes.
> Please plug in the device (will time out in 60 seconds) ...
```

- Plug your Digispark ATtiny85 board into a USB port.

- The IDE should detect the Digispark and begin the upload process automatically.

chadGPT_attiny85 | Arduino IDE 2.3.4

File Edit Sketch Tools Help

ATtiny85 (Micronucleus / ...

chadGPT_attiny85.ino

```
39 DDRB &= ~(1 << PB3); // Release PB3
40 DigiKeyboard.delay(100); // Short delay before continuing
41 }
42
43 void setup() {
44   pinMode(buttonPin, INPUT_PULLUP); // Configure button pin
45   lastButtonState = digitalRead(buttonPin);
46   // Seed the random number generator with an unconnected analog pin:
47   randomSeed(analogRead(A1));
48
49   // Optional: Uncomment this if USB re-enumeration is necessary
50   // usbReconnect();
51
52   DigiKeyboard.delay(1000); // Stabilize USB connection (adjust if needed)
53   DigiKeyboard.sendKeyStroke(0); // Initialize keyboard HID
54
55   DigiKeyboard.println("Device Ready"); // Signal readiness
56 }
57
58
59 void loop() {
60   int reading = digitalRead(buttonPin);
61
62   // Debounce logic
63   if (reading != lastButtonState) {
64     lastDebounceTime = millis();
65   }
66 }
```

Output

```
> Available space for user applications: 6586 bytes
> Suggested sleep time between sending pages: 7ms
> Whole page count: 103 page size: 64
> Erase function sleep duration: 721ms
parsing: 50% complete
> Erasing the memory ...
erasing: 55% complete
erasing: 60% complete
erasing: 65% complete
> Starting to upload ...
writing: 70% complete
writing: 75% complete
writing: 80% complete
> Starting the user app ...
running: 100% complete
>> Micronucleus done. Thank you!
```

Note: Digispark typically has a bootloader that waits a few seconds for an upload upon being powered/plugged in.

10. Wait for “Device Ready”

- After a few seconds, if everything goes correctly, the Arduino will write via USB:

Device ready

- This indicates a successful upload.

Troubleshooting Guide

1. Computer Doesn't Recognize the Device

- **Check USB Cable:** Ensure you're using a data-capable micro USB cable (many cables are charge-only).
 - **Re-seat the Board:** Confirm the ATtiny85 (Digispark) board is firmly plugged into its USB port or adapter. Try using different USB Ports -> connect directly without any adapters or hubs
 - **Install Drivers:** For some operating systems, you may need additional drivers (e.g., Digistump drivers on Windows).
-

2. No Response or Keystrokes

- **Verify Wiring:** Make sure the keyboard switch wires are correctly soldered (one to GND, one to PB0).
 - **Inspect Solder Joints:** Cold or weak solder joints can break the connection; reflow if necessary.
 - **Confirm Firmware:** Double-check you've uploaded the correct .ino file and selected the right board in your IDE.
-

3. Wrong Characters or Key Layout

- **Keyboard Layout Definition:** In the .ino file, ensure you've commented out the correct layout line, e.g.:
 - `#define KEYBOARD_LAYOUT_GERMAN`
 - `// #define KEYBOARD_LAYOUT_ENGLISH`Switch them if you need English instead of German (or vice versa).
 - **OS Keyboard Settings:** Verify your computer's language/keyboard input settings aren't overriding the device's output.
-

4. "Device Ready" Message Never Appears

- **Re-flash Firmware:** The upload may have failed. Try flashing again.
 - **Power Cycle:** Unplug the device, wait a few seconds, then plug it back in.
-

5. Button Feels Stuck or Doesn't Spring Back

- **3D-Printed Springs:** Make sure the springs are inserted correctly and are not obstructed.
 - **Alignment:** Check that the button aligns properly with the housing (watch for cable notches).
 - **Glue Overuse:** Excess glue can restrict movement. Clean up any excess if necessary.
-

6. Other Unlisted Problems

- **Review the PDF:** Refer to the detailed documentation for more in-depth steps.
- **Check GitHub Issues:** See if others have encountered the same problem; solutions might already be posted.
- **Ask the Community:** If all else fails, open a GitHub issue or seek help on relevant forums.