

---

# **The Macro Project**

***Release 0.1***

**Aniket Katkar**

**Mar 22, 2023**



# CONTENTS

- 1 Introduction 3**
- 2 MacroPod One 5**
  - 2.1 MacroPod One Hardware . . . . . 5
  - 2.2 MacroPod One Software . . . . . 7
  - 2.3 MacroPod One Examples . . . . . 8
- 3 MacroPod Lite 9**
- 4 MacroPod Pro 11**
- 5 MacroPod Pro (Wifi) 13**
- 6 Use Cases 15**
  - 6.1 Macro Pod Lite Use Cases . . . . . 15
  - 6.2 Macro Pod Pro Use Cases . . . . . 15
  - 6.3 Macro Pod Pro (Wifi) Use Cases . . . . . 16



**The Macro Project** is a open source project to help increase productivity of day to day tasks and workflows with the use of Macro Keyboards. The ultimate open-source macro keyboard - designed for the power user who demands flexibility and customization. With its open-source firmware and programming capabilities, this keyboard allows you to fully customize and personalize your typing experience.

Unlike proprietary macro keyboards, an open-source macro keyboard gives you the freedom to modify and improve the firmware to your exact needs. You can create custom macros, remap keys, and even add new functionality. The open-source community also provides a wealth of resources, including tutorials, FAQs, and forums where you can share your work and learn from others.

With its durable design, customizable layout, and advanced programming capabilities, the open-source macro keyboard is perfect for gamers, programmers, and power users who want to take their productivity to the next level. Plus, you can be sure that your keyboard will continue to work and be supported with new updates and features.

For more information, visit [The Macro Project website](#)

Checkout our social media to get the latest updates on all the development and updates. [Website](#) [Instagram](#) [LinkedIn](#) [Twitter](#) [Reddit](#) [Linktree](#)

Check out our blog on [Medium](#)

Support us on our Patreon

<https://patreon.com/themacroproject>

---

**Note:** This project is under active development.

---

### Overview

- *Welcome to The Macro Project documentation!*



## INTRODUCTION

The following are pre-requisites to design your own custom Macro keyboard. We have provided all the guides and resources to various topics such as Arduino coding, Hardware, PCB design.

### Summary:

An Arduino microcontroller can be used to build a custom macro keyboard by programming it to recognize specific keystrokes and perform a corresponding action, such as executing a macro or launching a program. This allows for automation and streamlining of repetitive tasks, as well as customization and personalization of the keyboard. We will mostly be using Arduino or Arduino IDE supported boards as they are highly customizable and is supported by a large community.

### Arduino:

[Arduino](https://en.wikipedia.org/wiki/Microcontroller) is an open-source electronics platform based on simple, easy-to-use hardware and software. It consists of a *microcontroller* <<https://en.wikipedia.org/wiki/Microcontroller>>\_ board and a development environment for writing software, which can be used to control various devices and sensors. Arduino boards are commonly used for projects such as home automation, robotics, and Internet of Things (IoT) applications due to their versatility, low cost, and ease of use. With a basic understanding of electronics and programming, anyone can use an Arduino to build and control a wide range of interactive devices.

**Arduino IDE:** The Arduino Integrated Development Environment (IDE) is a software application used for programming and uploading code to an Arduino microcontroller board. The Arduino IDE provides a simple and user-friendly interface for writing, compiling, and uploading code to the board. It supports multiple programming languages, including C and C++, and includes libraries and examples to help users get started quickly. The Arduino IDE also allows for debugging and testing of code, making it a valuable tool for beginners and experienced users alike. Overall, the Arduino IDE provides a streamlined and accessible way for users to develop and control projects using an Arduino board. Download Link - [Arduino IDE](#)

To learn more about how to use Arduino, check out the following links.

[Arduino Docs](#),

[Getting Started](#),

[Tutorials](#).

### Boards:

Any board that is supported by Arduino can be used for this project. The pins and connections with the mechanical switches would change accordingly. To check which boards we can use with the Arduino IDE, please refer [Supported Boards](#)

Check out our Github for more details - [Github](#)

---

**Note:** This project is under active development.

---





## MACROPOD ONE

**MacroPod One:** The MacroPod One is a single programmable button that can be customized to perform a specific task or execute a series of commands with a single press. It is typically used to automate repetitive actions, such as opening a frequently used program, entering a complex string of characters, or triggering a specific function in a game. The key can be programmed using software provided by the manufacturer or with third-party tools that allow for more advanced customization. These keyboards are popular among gamers, programmers, and power users who want to streamline their workflow and increase productivity.

One key macro keyboards can be especially useful for gamers who want to optimize their gameplay experience. For example, a gamer may program a key to activate a specific combo of moves in a fighting game, or to switch weapons quickly in a first-person shooter. This can help reduce reaction time and increase precision, giving the player a competitive edge.

In addition to gaming, one key macro keyboards can also be useful in other contexts, such as programming or data entry. Programmers may program a key to automatically insert frequently used code snippets, while data entry specialists may use a key to input common phrases or symbols.

The hardware and software sections give a detailed information on how to make your own one key macropod along with the code, circuit design and examples.

---

**Note:** This project is under active development.

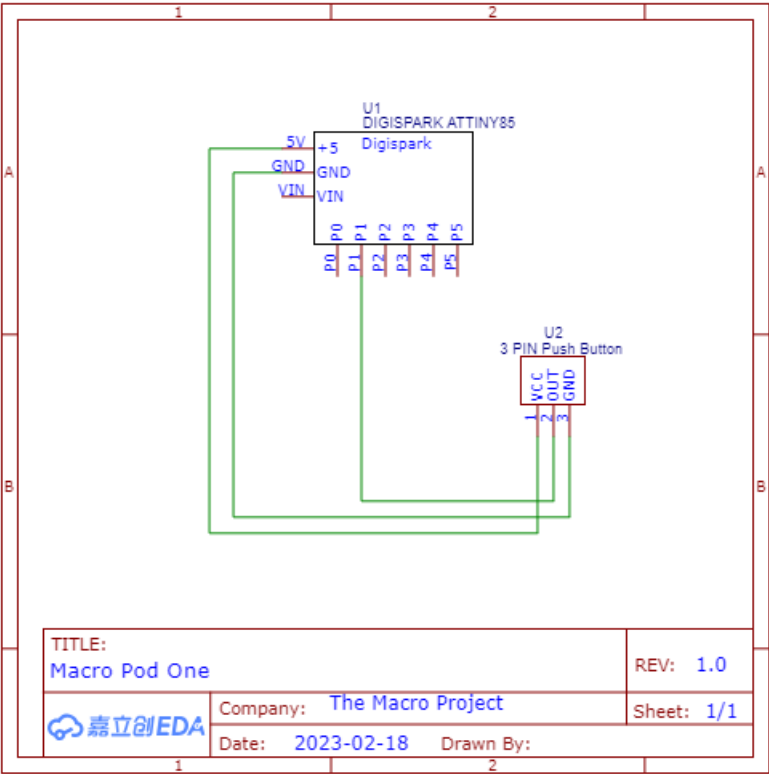
---

### 2.1 MacroPod One Hardware

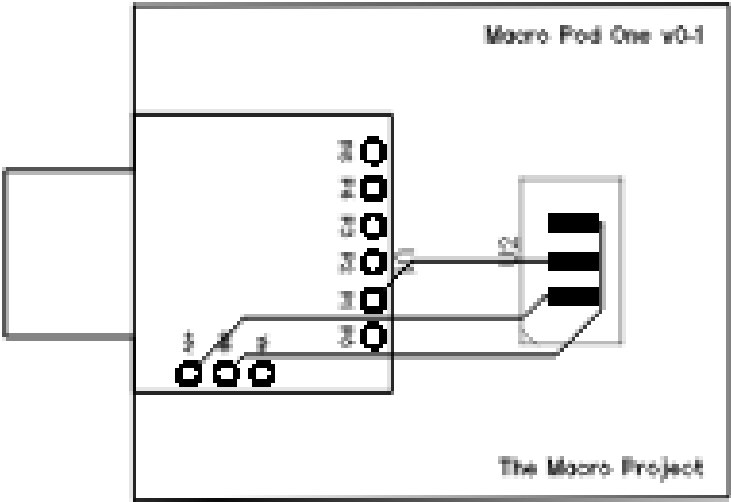
#### Hardware:

The Digispark Attiny85 is a small, low-cost microcontroller board that is compatible with the Arduino IDE. It is based on the ATtiny85 chip and comes with a USB connector, which allows it to be easily programmed and powered. Despite its small size, the Digispark Attiny85 is capable of running a wide range of programs and can be used for a variety of projects, such as controlling LEDs, motors, and sensors. It also has a number of built-in features, such as PWM output, analog-to-digital conversion, and support for I2C and SPI communication protocols. Its compact size makes it an ideal choice for projects where space is limited or portability is required. You can learn more about Attiny85 [here](#). Below are the schematic, PCB and Gerber files which we made using EasyEDA. You can check the tutorials on how to design in Easy EDA [HERE](#)

#### Schematic:



PCB:



Gerber File: [File](#)

**Note:** This project is under active development.

## 2.2 MacroPod One Software

### How to program Digispark Attiny85 using Arduino IDE

1. Install the Digispark board definitions in Arduino IDE.

#. Open the Arduino IDE and go to File > Preferences. In the Additional Boards Manager URLs field, add the following URL: [https://raw.githubusercontent.com/digistump/arduino-boards-index/master/package\\_digistump\\_index.json](https://raw.githubusercontent.com/digistump/arduino-boards-index/master/package_digistump_index.json) Click “OK” to close the Preferences window. Install the necessary USB drivers:

#. Download the appropriate driver for your operating system from the Digispark website: <https://digistump.com/wiki/digispark/tutorials/connecting> Follow the installation instructions for the driver. Connect the Digispark to your computer:

#. Plug the Digispark into an available USB port on your computer using a micro USB cable. Configure the Arduino IDE to use the Digispark board:

#. Go to Tools > Board and select “Digispark (Default - 16.5mhz)” from the list. Write your code in the Arduino IDE:

#. Write your code in the Arduino IDE as you normally would. Upload your code to the Digispark:

#. Click the Upload button in the Arduino IDE. The IDE will compile your code and then upload it to the Digispark.

#. Test your code: Once the code is uploaded, the Digispark will automatically run it. Test your code to make sure it works as expected.

#### The Code:

This code is designed to control a button press that sends a keyboard shortcut to a computer. It uses the DigiKeyboard library to allow an Arduino board to act as a USB keyboard.

Include the DigiKeyboard library.

```
#include "DigiKeyboard.h";
```

Constant keyPin1 is defined as 1, which indicates that the button is connected to pin 1 on the board.

```
const int keyPin1 = 1;
```

An integer variable keyState1 is declared and initialized to 0. This variable will store the state of the button.

```
int keyState1 = 0;
```

In the setup() function, the keyPin1 is set as an input pin using the pinMode() function.

```
void setup() {
  pinMode(keyPin1, INPUT);
}
```

In the loop() function, the DigiKeyboard.sendKeyStroke(0) function sends a null key to the computer to prevent missing the first character after a delay.

```
DigiKeyboard.sendKeyStroke(0);
```

The digitalRead() function reads the state of the button and stores it in the keyState1 variable.

```
keyState1 = digitalRead(keyPin1);
```

Then, an if statement checks if the button is pressed by comparing `keyState1` to `HIGH`. If the button is pressed, the `DigiKeyboard.sendKeyStroke()` function is called to send a keyboard shortcut to the computer. In this case, it sends the Windows shortcut to show the desktop using the `KEY_D` and `MOD_GUI_LEFT` parameters.

Finally, the `DigiKeyboard.delay()` function is used to wait for 500 milliseconds before repeating the loop. This function is preferred over the `delay()` function used in regular Arduino programming because it keeps the computer informed that the keyboard is still alive and connected.

For more information, check the MacroPod One [Github](#)

---

**Note:** This project is under active development.

---

## 2.3 MacroPod One Examples

### Examples

There are two ways to program the MacroPod One- to insert characters and/or to send keystrokes (single or combined).

1. To send any characters, use the following in the if block of the `loop()` function.

```
DigiKeyboard.print("ABCDEFGH");
```

Check out the code [HERE](#)

2. To send a keystroke or a combination of keys, use the following in the if block of the `loop()` function.

```
DigiKeyboard.sendKeyStroke(KEY_D, MOD_GUI_LEFT);
```

Check out the code [HERE](#)

Here, 'KEY\_D' is the character D and 'MOD\_GUI\_LEFT' is the windows key.

Check out the list of predefined keys that you can use in this [PDF](#)

For more information, visit [The Macro Project website](#)

---

**Note:** This project is under active development.

---

## MACROPOD LITE

**MacroPod Lite:**

#TODO

---

**Note:** This project is under active development.

---



## MACROPOD PRO

**MacroPod Pro:**

#TODO

---

**Note:** This project is under active development.

---





## MACROPOD PRO (WIFI)

**MacroPod Pro (Wifi):**

#TODO

---

**Note:** This project is under active development.

---



## USE CASES

Here are basic snippets of code that you can use to understand and extend your own logic to add features or new definitions to your buttons.

---

**Note:** This project is under active development.

---

### Use Cases

- *Use Cases*

## 6.1 Macro Pod Lite Use Cases

### Use Cases

#TODO

For more information, visit *The Macro Project website* <<https://theproject.github.io/>>

---

**Note:** This project is under active development.

---

## 6.2 Macro Pod Pro Use Cases

### Use Cases

#TODO

For more information, visit *The Macro Project website* <<https://theproject.github.io/>>

---

**Note:** This project is under active development.

---

## 6.3 Macro Pod Pro (Wifi) Use Cases

### Use Cases

#TODO

For more information, visit *The Macro Project website* <<https://theproject.github.io/>>

---

**Note:** This project is under active development.

---