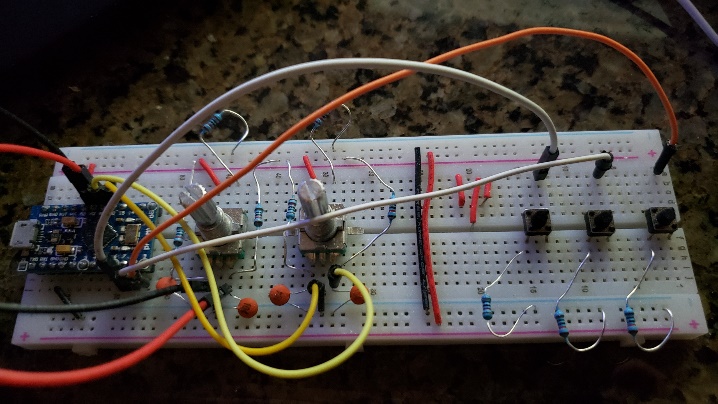
This year I am trying a new thing for a Christmas gift where I can geek out a bit, be crafty and provide something that might be fun and useful at the same time. After many iterations of both whacky and practical gift ideas I settled on a Macro Keypad that was programmable and can adapt to individual needs.

**CHRISTMAS | 2020**

This began way back at the beginning of October and by the end of October, I realized that time was not on my side. The original gift idea was to be a complete design and build of a custom device. Maybe this will be a thing for next year. Who knows what my crazy brain might dream up? Here is a photo of my working prototype just for grins.

I went on the hunt for a macro keypad device that I felt was capable of significant customizations. I then located a case file which was customized just a bit and 3D Printed so the keypad would have a nest to sit inside of. The visual style of the device had to be worthy of making a prominent desk appearance. The final device will work on any computer that can accept a USB Keyboard. I have bundled the device with some preloaded programming to get you started out of the box and some instructions on how you can customize the device for your very own needs. As you venture into creating your very own customizations, fear not because I am here to assist. Just ask if you are uncertain how to make it work the way you want it to work.

The final device has 16 buttons that can be individually customized to do anything you can do with a keyboard or mouse. I hope that you have as much fun configuring and using it as I had building and programming it. Here is a photo of the finished product.

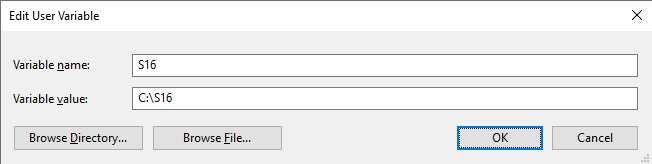
**Merry Christmas 2020**

**-Greg Miller**

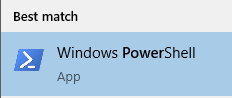
Christmas 2020 Macro Keypad User Guide

To get the device to function out of the box, there are a few steps that are required. This is only for the pre-programmed configuration to function and was done this way more as example. The pre-programmed example is created for a Windows environment but could easily be adapted to a Mac or Linux machine.

To get this device to work with the pre-programmed macros, follow these few easy steps:

* Download the supporting files from my personal GitHub repository and save these to a folder named “C:\S16”. The folder name is very important.  
  <https://github.com/GeekPatrolMiller/Christmas2020>
  + Look for the button and you can download a .ZIP file of the entire repository. Extract the contents of this .ZIP file into the C:\S16\ folder you just created.
* Next, create a new Environment Variable on your computer that is called S16 and has the value of “C:\S16”. To do this, follow these steps
  + Open Control Panel by pressing the Windows button and type Control and pressing Enter.
  + Click on the “System” icon
  + Click on Advanced System Settings
  + This will open the System Properties dialog box. Click on the Advanced Tab, if not already opened to this place
  + Click the button for Environment Variables in the bottom right
  + Click New under User Variables and enter the following exactly as it appears here and click OK:  
    

(You can modify these to anything you wish later if you like but my examples are looking specifically for these names)

* + Close all the windows you opened to get here to continue.
* Open a PowerShell console window by pressing the Windows button and typing PowerShell and pressing Enter. There might be multiple versions of PowerShell on your computer so open this one
* In the PowerShell console type the following command: **$Env:S16**  
  and press Enter. You should get a response of: **C:\S16**
  + If you do not get this response, doublecheck your creation of the Environment Variable in the previous step.
  + If this checks out, close the PowerShell console window to continue
* Now plug in the Macro Keypad Device and wait for it to boot up. After a few seconds you will see the internal LED lights turn on.
* Using this chart as your guide, press a key and you should see it working:

|  |  |  |  |
| --- | --- | --- | --- |
| **Encompass** | **Chrome** | **Firefox** |  |
| **Outlook** | **Word** | **Excel** | **PowerPoint** |
|  |  |  |  |
| **Shutdown** |  |  | **RGB Mode** |

( Blank buttons are not defined yet and will indicate this via onscreen message )

* Here is a cool feature, if you press the button to launch the application but that application is already running, the controlling code will bring that application to the foreground instead of launching a new instance. This allows you to change quickly between applications with a single button press.
* There are two special buttons I have placed on the bottom row which do not use PowerShell to provide the control.
  + **Shutdown** – This will send the ACPI signal to the computer. This is equivalent to pressing the power button one time. The computer will begin performing a shutdown routine.
  + **RGB Mode** – This button will allow you to change the LED effects that are shown. There are many variations to play with.

How to make this device my own?

There are several web-based tools you can use to program this device to do other things. Here are some sites and tools that I recommend but you will find others that you may prefer better. Use the tools that you feel the most comfortable using.

**QMK Configurator** – [https://config.qmk.fm/](https://config.qmk.fm/#/1upkeyboards/sweet16/v1/LAYOUT_ortho_4x4) - This is a web-based configuration tool that allows you to setup many basic key layouts and generate firmware.

**Keyboard Firmware Builder** - <https://kbfirmware.com/> - This is a more feature rich web-based configurator that allows you to do all the same functions as QMK but also supports many more complex configurations. My example configuration was created with this tool.

**QMK Toolbox** – <https://github.com/qmk/qmk_toolbox/releases/> - This is the preferred tool to install the configuration firmware files into the Macro Keypad. There are many tools like this one and they all work well. Use the tool that fits your needs. I like this one because it is very basic and simple to use.

Steps Required to Customize…

There are preconfigured firmware files that I have built in GitHub, including the one which is already installed on the device. You can always install one of these for a known working configuration.

I am only going to give detailed instructions for the QMK Configurator site as it is easier to use and has a more user-friendly interface. Once you master the QMK Configurator, the KFB site should be simple to understand.

* Open the QMK Configurator site
* In the **KEYBOARD** dropdown select “**1upkeyboards/sweet16/v1**” if it is not already selected. It is very, very important that you have the right keyboard selected else you can damage the device permanently.
* In the **LAYOUT** dropdown, select “**LAYOUT\_ortho\_4x4**” if it is not already selected.
* Enter an optional **KEYMAP NAME** which will be used in naming your customizations for later use.
* In the middle section, make sure the **LAYER** selection is set to layer 0.
* The KEYMAP selection is optional and only changes the key colors on screen. This has no effect on any functionality. Set to your favorite if you want to.

You are now ready to configure the keypad. To do this, look at the key layout of the keypad on screen. If you saved now, this would be how your keypad would be configured after installing the firmware. You can change this layout in a few ways. The simplest way is to click on the on-screen key you want to change and then press the key you want it to become, on your keyboard. You can also click on the key within the on-screen keyboard as well. Go through all the devices keys and assign then how you want.

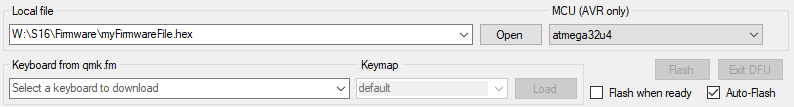
Once this is completed, you are ready to save your configuration file. Always save your configuration files as they can be imported back into the configurator to make small changes without having to fully redefine the entire keypad from scratch. To save the configuration you need to click on **Export QMK Keymap JSON file** on the left side of the button. The default name for the file will be a combination of the **KEYBOARD** and **LAYOUT** names you chose at the start of this process. After you download this file, it is strongly suggested that you rename the file with a more friendly name so you can recall what it does.

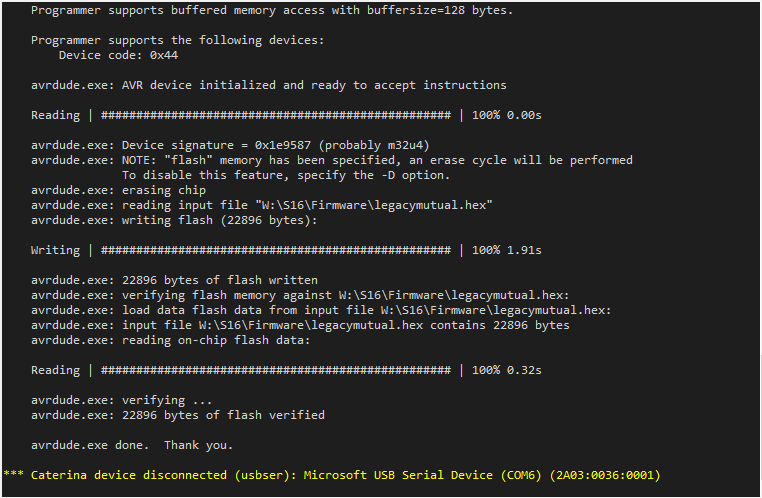
Now you are ready to create your Firmware that will control the behavior of your device. Click the  button and you will see the Baking animation while your Firmware is being compiled. After a moment or two, the baking animation will go away, and you can now download your firmware for flashing to your device. Click the  button to save your file. The default name for the file will be a combination of the **KEYBOARD** and **LAYOUT** names you chose at the start of this process. After you download this file, it is strongly suggested that you rename the file with a more friendly name so you can recall what it does, because there is no way to determine.

Flashing the Device!

We are on the final step to customizing your device. You need to download the **QMK Toolbox** from the link provided previously. This is the tool that will be used to install the firmware you just created onto the device which will modify its behavior. Follow these steps to perform the firmware installation:

* Plug in your Macro Keypad Device to an available USB port
* Launch **QMK Toolbox**
* Set the MCU (AVR Only) dropdown to **atmega32u4**
* Check the Auto-Flash checkbox
* Click the Open button and browse to the location you saved your firmware file, select the file, and click the Open button.

Your screen should resemble this:  


* With the device still plugged into the USB, turn the device over and using a pen or some other object that will fit into the square hole on the bottom, quickly depress the reset button two times.
  + The double reset will put the device into a state called Bootloader Mode for about 8 seconds.
  + If you do not double press the reset button fast enough, nothing will appear to happen other than the LED lights may restart whatever sequence they are currently set to display.
  + If you do successfully double press the reset button, on a windows machine, you will hear the USB unplug device sound.
* If successful in the double tap on the reset button, you will see the dark text area in the center of the application begin to show progress information.   
  If there are error messages on this screen, start over at the beginning of this section and try again. This is the trickiest part of this entire operation by far.
* **Test your new functions…!**

Additional Tips to Share

On the QMK Configurator site, you may have noticed the wizard hat in the upper right corner. If you click on it you are presented with a nice video that will provide some additional features of the Configurator site that you may find super useful.

As you start to get creative with your device and want to get into some of the more advanced features, here is a link to the KeyCodes that are supported.  
<https://docs.qmk.fm/#/keycodes?id=keycodes-overview>

Explore both of the Configurator websites for much more content that I could just not give enough time to discuss here. These are amazing free tools and they are only the tip of the iceburg. If you look for others, you will find so, so many.

If you are wanting to take it even further and you have some C/C++ programming experience, you can download the entire source code for the QMK and KFB configurations. From there you can create new feature that are not currently available through the standard web-based configurator.

The QMK Configurator allows for basic key assignments with some advanced functionality but if you want to get a bit more advanced and use Macros, you will want to learn more about the Keyboard Firmware Builder (KFB) Configurator. This tool allows for easy access to create many more complex functions without needing to know how to program in C/C++.

I have found that loading a JSON file to edit it and then save it again get a little tedious. If you want to dig into the JSON file, you can edit these files using a simple text editor. The only downside to this is that the JSON files that you save from the configurators are flattened, which means they have no layout format and appear to be a single line of text. This is very hard to decipher without a text editor that can format the look so it is easier to read. This is sometimes called a pretty view or beautified. A nice example of an online editor which works fairly well is: <https://jsoneditoronline.org/> There are many of these types of tools out there so exercize your Google-Fu and find one that you like.

Final Thoughts…

This is the first time I have taken up a project such as this and while my original dream was not realized, I do plan on continuing my own custom design and build for the hopefully near future. I learned a lot along the way and will use that knowledge to extend the scope and functionality of projects in the future. I hope that you enjoy this gift as much as I enjoyed building it.