



Programmable Keypad

by [jordantallent](#) on June 17, 2013

Table of Contents

Programmable Keypad	1
Intro: Programmable Keypad	2
Step 1: Materials	2
Step 2: Setup	3
Step 3: Arduino Code	3
File Downloads	4
Step 4: Flip	4
Step 5: Enclosing	5
Step 6: Reprogramming	5
Related Instructables	5
Advertisements	5

Intro: Programmable Keypad

I recently got an internship with a company that is a distributor of industrial products such as hydraulics, pneumatics and various other mechanical and electrical devices. My most recent assignment was definitely work for an intern. They have an online store with tens of thousands of products and lucky for me, approximately half of these items did not have an associated picture or datasheet. My job, to take this giant spreadsheet of pictureless items and search for them online and then save the pictures in one folder and the datasheets in a separate folder.

Now after the first day of constant copying and pasting, and having to change the folder I was using, my hand hurt and I was tired of the repetition so I got the idea. I had an idea earlier in the year to make a little keyboard that I could use for symbols when writing a lab report, like the ohm symbol, micro symbol, etc. Well I never did do that but I had all the parts for it so I decided to finally make this project and use it to help me at my job.



Step 1: Materials

Arduino Uno(<https://www.sparkfun.com/products/11021>)

Keypad(<https://www.sparkfun.com/products/8653>)

Sparkfun enclosure(<https://www.sparkfun.com/products/8601>)

Small pushbutton, NO(already had)

USB cable(already had)

Ribbon cable with headers or jumper wires

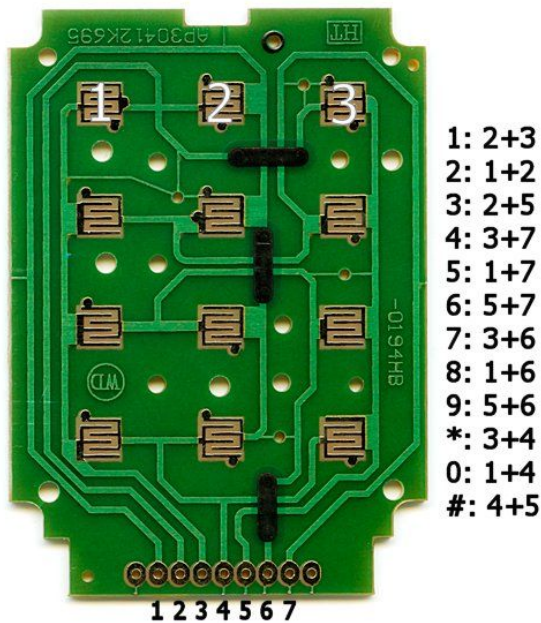


Image Notes

1. Arduino Uno
2. SparkFun Enclosure
3. Sparkfun Keypad
4. Sparkfun Keypad
5. Small push button

Step 2: Setup

To get started, I looked at the pictures of the keypad on the Sparkfun website, specifically the picture with the buttons removed. This allowed me to trace the tracks and figure out which pin corresponds to which row or column of the keypad. After doing this, I loaded up the example keypad sketch from the Arduino website ([here](#)). I loaded this code just as a base and used it to make sure that all of the pins were assigned correctly using the serial terminal to confirm.



Step 3: Arduino Code

After I had the right pins assigned in the basic keypad code, I started to add the necessary code to use an Arduino Uno as an HID keyboard. [Here](#) is all the reference for the mouse and keyboard libraries that are used on Arduino. This is a good learning spot to see what everything that will be in the code means. After downloading the library and adding it to my "Libraries" folder in Arduino, it's time to start coding! Attached is my final code, the most recent one is the 'LH_keyboard' that I made for work, you'll notice that I only used 4 of the available 12 buttons, this is simply because I only needed four macros for what I was doing. This is easily extendable to fit any of your needs. This code is pretty self explanatory and simple. All it does is wait for a key to be pressed, then based on which one was pressed, types out a certain keystroke. The four buttons used on my keypad were programmed to copy, paste, and do a save/save as in two different locations. To get the keyboard library to recognize certain special characters, such as slashes or quote marks, you must use the ASCII representation of the symbol and send it using "keyboard.pressKey(ASCII);", where you replace 'ASCII' with the corresponding number found on an ASCII table.



File Downloads



Arduino UNO keyboard.zip (20 MB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Arduino UNO keyboard.zip']

Step 4: Flip

In order to have your computer recognize the Arduino Uno as a keyboard, we need to reprogram the Atmega16U2. This is the small chip that handles all of the USB controls. We can program hex files onto it using Atmel's Flip software, [here](#) . You will also want to download the appropriate hex file from [here](#) . The preloaded hex file can be found in your Arduino folder, \hardware\arduino\avr\firmwares. To get the USB chip in programming mode, you must short the reset pin to ground. This can be simply done by taking a header jumper and temporarily connecting it across the two pins that are closest to the RESET button on the Arduino Uno. Not the pins in the female headers, but the pins on the male headers shown in the picture.

After you have Flip installed, run it and have your Arduino plugged into your computer. Reset the Atmega 16u2. Click on you USB cable icon in Flip and select USB. Select the button that looks like an IC and choose the correct device. Now you're ready to load the hex file! Click File, Load HEX File and navigate to the keyboard hex file downloaded from above. After loading it simply press Run and wait for the process to finish. This usually only takes a few seconds. After the chip has been reprogrammed, kill power to the Arduino and start it back up. Now when you plug your Arduino Keypad into your computer it will be seen as a keyboard!

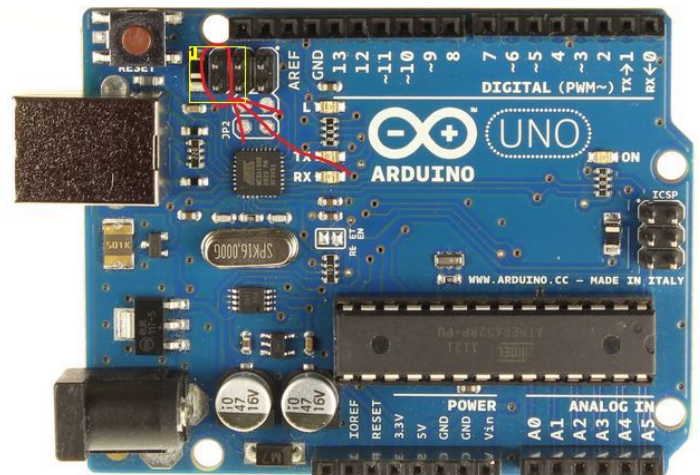


Image Notes

1. pins used to set chip in program mode

Step 5: Enclosing

To make an enclosure for my keypad I just used a SparkFun enclosure and a small push button. The push button is connected to the reset pin of the 16u2. This allows for easy reprogramming without having to take out the Arduino. The enclosure from SparkFun is very easy to use. I started by just cutting out a big hole in the top that the keypad would fit through. Then just mounted the Arduino so that the USB port was right up against the backing plate so that I could cut out a hole for that to make an easy way to connect the USB cable. I also drilled a small hole in the back plate to put the push button.

Another enclosure idea would be to make a custom case if you have access to a 3D printer.



Step 6: Reprogramming

To reprogram the buttons on your keypad, you need to reload the original hex file onto the 16u2. Then re-upload your Arduino code, load the keyboard hex file and you're good to go!

If you use an Aduino Leonardo or Micro I am pretty sure that the need to reprogram the 16u2 is obsolete, as they are only single chip boards.

ENJOY!!!!!!

Related Instructables



Arduino Resistor-less keypad access control (Photos)
by donromeo



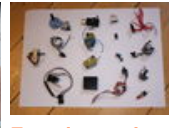
Password Lock with Arduino by razvan_iycdi



Password access with arduino by razvan_iycdi



DinoCalc Version 1.0 by weaseljones



Extra inputs for Arduino with a keyboard by 02JanDal



Access control with Arduino + Keypad 4x4 + Servo by camilo.n1012