



instructables

How to Make a Arduino HID Keyboard



by Eward Hage

Controller by picklesandicecream

I also have made a youtube Video for the coding part.

In this tutorial I shall explain how you can turn your Arduino chip into a HID keyboard device.

This is done by updating the Firmware on your chip with FLIP.

[//www.youtube.com/embed/j05vj8zRP1o](https://www.youtube.com/embed/j05vj8zRP1o)

The cool thing about this trick is that you can make your own game controller or macro keyboard with the power of the Arduino coding method, so possibility are endless for great prototypes.

Supplies:

Arduino UNO or Mega

4 push buttons

4 resistors 1k ohm

Jumper cables

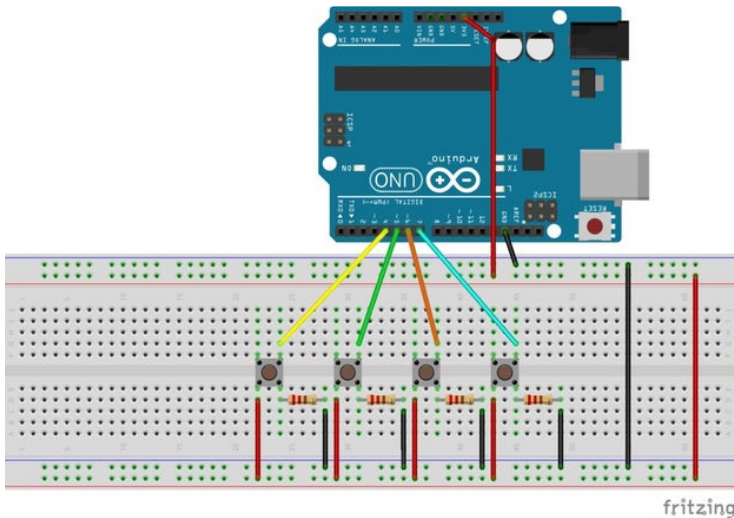
I have put all files into a Google Drive map for you to download.

link: https://drive.google.com/open?id=1jfOJmFf9C6QCb8I_...



Step 1: Setup of the Arduino

The basic setup of the Arduino for this example. (This can also be done with the Arduino Mega if more pin inputs are required) Note that we use the 3.3v of the Arduino as the input current of the push buttons onto the Digital-pins. The resistors used for this example are 1k ohm resistors.



Step 2: CODE - How to Code a Key Press Onto the Arduino

Video link: <https://www.youtube.com/watch?v=j05vj8zRP1o>

Step 3: CODE - Write a Keyboard Buffer

First we initialize a keyboard buffer, this is required for the Arduino to send a bit register as a HID Keyboard.

```
uint8_t buf[8] = { 0 }; //Keyboard report buffer
```

Step 4: CODE - Define the Pins for Input Into the Arduino

Define the pins of the Arduino, you can also use (const int PIN_W = 4;) but I prefer the #define method.

```
#define PIN_W 4 // Pin for w  
#define PIN_A 5 // Pin for a  
#define PIN_S 6 // Pin for s  
#define PIN_D 7 // Pin for d
```

Step 5: CODE - Write the Setup Function

Write the setup function, as with every Arduino code project the setup is pretty much the same except that setting the Baud rate and serial communication with the (Serial.begin(9600)) function is required for the Arduino to communicate with your computer.

```
void setup() {  
  Serial.begin(9600); // Setup Serial communication  
  
  pinMode(PIN_W, INPUT);  
  pinMode(PIN_A, INPUT);  
  pinMode(PIN_S, INPUT);  
  pinMode(PIN_D, INPUT);  
}
```

Step 6: CODE - Write a Button Release

Write an end buffer method to send a bit when button is released, this step is required to end the data stream of the HID keyboard. For this example we send the key code through buf[2] so we need to reset them to 0 when the button on the Arduino is released. Note: Without this step you're HID keyboard can start sending button inputs but it will never stop sending bits until the HID keyboard (Arduino) USB is unplugged.

```
void releaseKey()  
{  
  buf[0] = 0;  
  buf[2] = 0;  
  Serial.write(buf, 8); // Release key  
}
```

Step 7: CODE - Write a Keypress

This step is done in the Loop() function to be updated every cycle the Arduino get through checking if a button is pressed.

```
void loop() {  
  if (digitalRead(PIN_W) == HIGH) {  
    buf[2] = 26; // W keycode  
    Serial.write(buf, 8); // Send keypress  
    releaseKey();  
  }  
}
```

Step 8: Complete Code for Reference

```
uint8_t buf[8] = { 0 }; //Keyboard report buffer

#define PIN_W 4 // Pin for w

#define PIN_A 5 // Pin for a

#define PIN_S 6 // Pin for s

#define PIN_D 7 // Pin for d

void setup() {

  Serial.begin(9600); // Setup Serial communication

  //Set pinmode of Input pins

  pinMode(PIN_W, INPUT);

  pinMode(PIN_A, INPUT);

  pinMode(PIN_S, INPUT);

  pinMode(PIN_D, INPUT);

}

void loop() {

  //When button representing W is pressed

  if (digitalRead(PIN_W) == HIGH) {

    buf[2] = 26; // W keycode

    Serial.write(buf, 8); // Send keypress

    releaseKey();

  }

  //When button representing A is pressed

  if (digitalRead(PIN_A) == HIGH) { buf[2] = 4; // A
  keycode

  Serial.write(buf, 8); // Send keypress

  releaseKey();

}

//When button representing S is pressed

if (digitalRead(PIN_S) == HIGH) { buf[2] = 22; // S
keycode

  Serial.write(buf, 8); // Send keypress

  releaseKey();

}

//When button representing D is pressed

if (digitalRead(PIN_D) == HIGH) { buf[2] = 7; // D
keycode

  Serial.write(buf, 8); // Send keypress

  releaseKey();

}

// Function for Key Release

void releaseKey() {

  buf[0] = 0;

  buf[2] = 0;

  Serial.write(buf, 8); // Send Release key

}
```

Step 9: Install Flip

To install FLIP please install JRE - Flip Installer - 3.4.7.112.exe provided in the Google Drive link

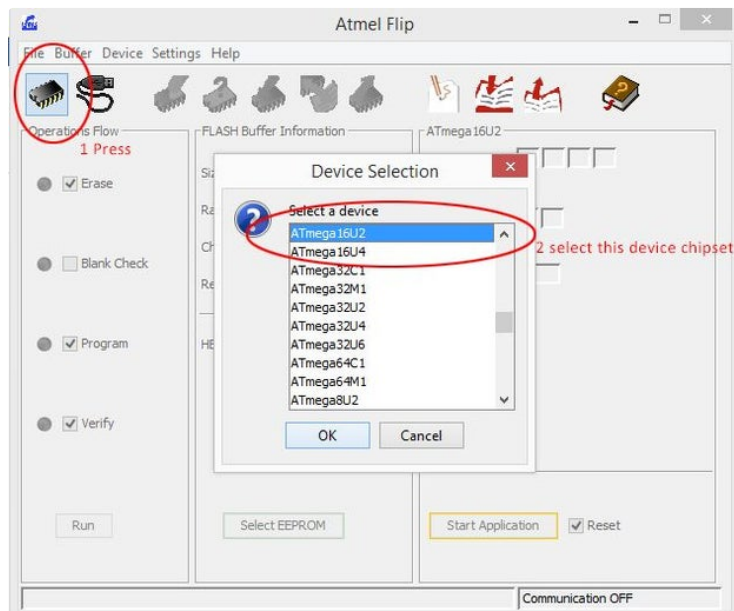
Note: When starting up you get the error "AtLibUsbDfu.dll not found" you have to install a driver.

Here is a solution made by MDGrein link: <https://www.youtube.com/watch?v=KQ9BjKjGnlc>

Step 10: FLIP - Device Selection

Press the Highlighted chip button, then Select ATmega16u2 and then the button OK. As this is the chipset for the Arduino Uno R3 and the Arduino Mega R3.

And now you are done with setting up flip and we can go to the Next step of Flashing the Arduino.

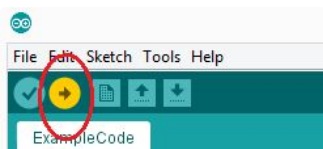


Step 11: Upload Your Arduino Code

You will get a message at the bottom from the Arduino IDE saying done Uploading

Note: if uploading fails

Check 1 COM port under Tools>port if none is available unplug and replug USB or check device manager. Check 2 if you have the correct board selected in Tools>board.

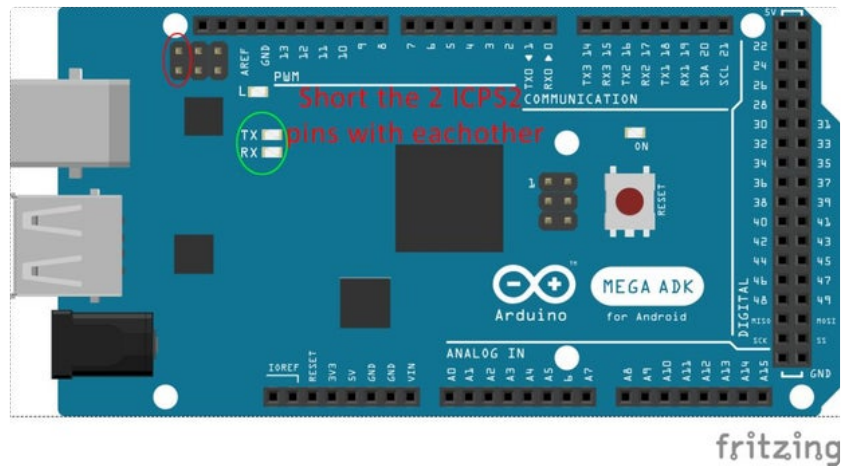
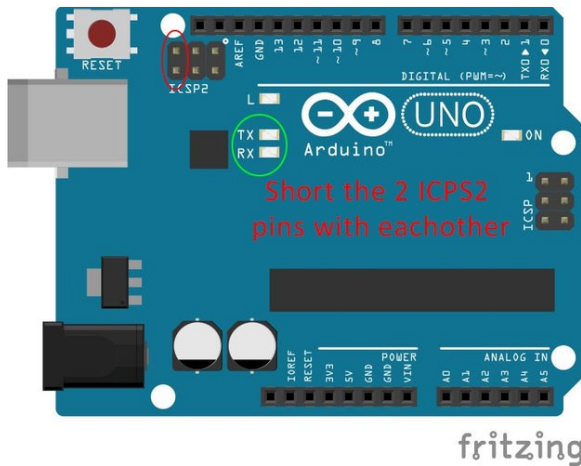


Step 12: Arduino Into DFU Mode

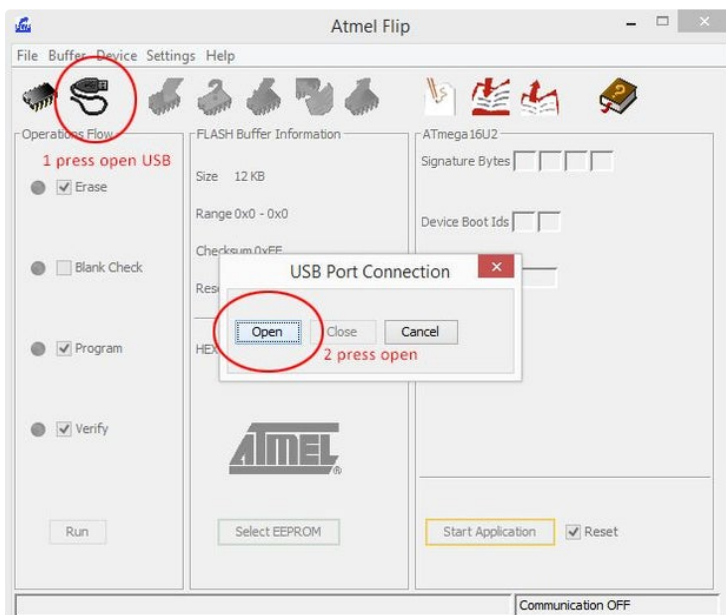
short the 2 ICSP2 pins as shown in the image

To put the Arduino in the DFU (Device Firmware Update) mode, so that FLIP can access the Firmware

on the Arduino Chip. More info on this reference link: <https://www.arduino.cc/en/Hacking/DFUProgramming8...>



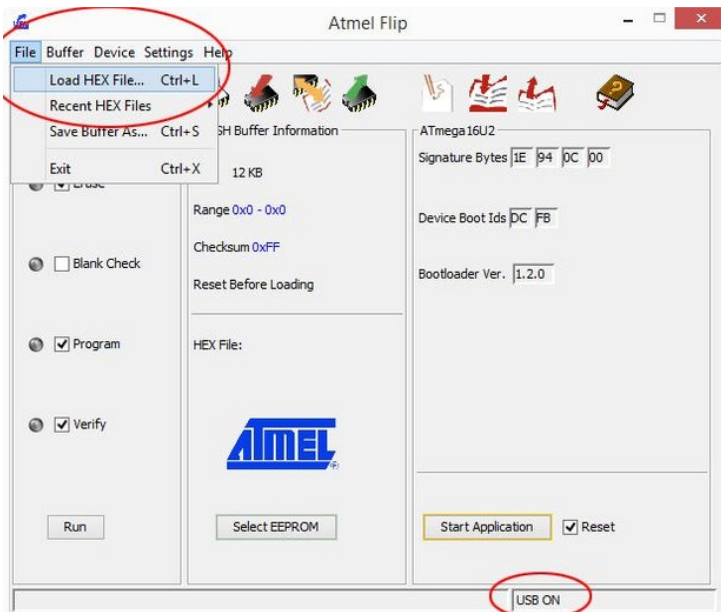
Step 13: FLIP - Press the USB Icon in FLIP and Then Open



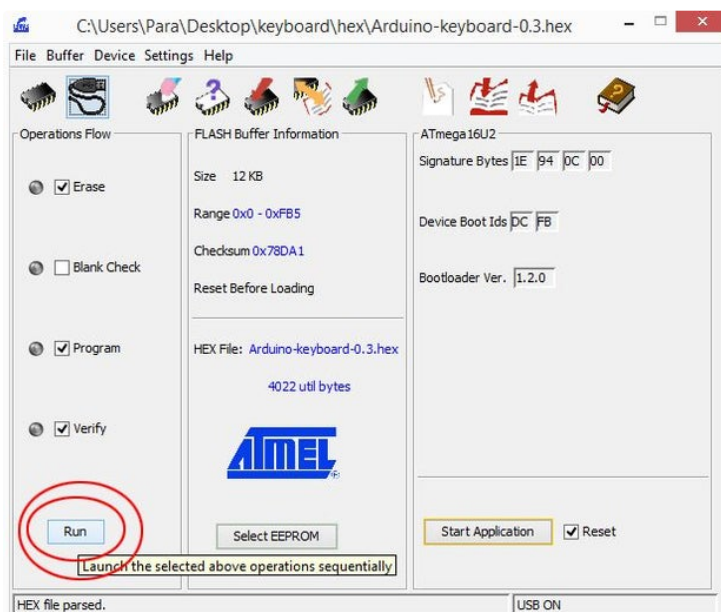
Step 14: FLIP - Parse the Firmware HEX

Load the **arduino-keyboard.hex** into FLIP

You will see at the bottom USB ON, this is that the Arduino is connected through DFU Mode.

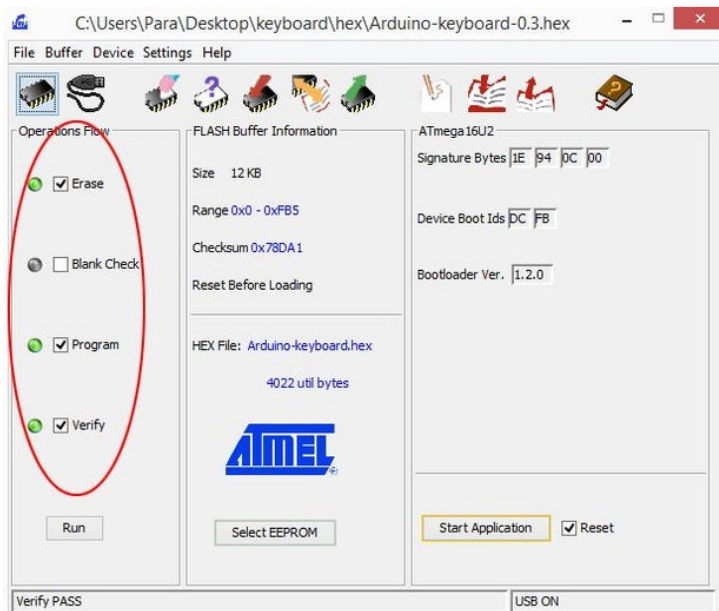


Step 15: FLIP - Press RUN Not Start Application



Step 16: FLIP - Unplug and Replug USB

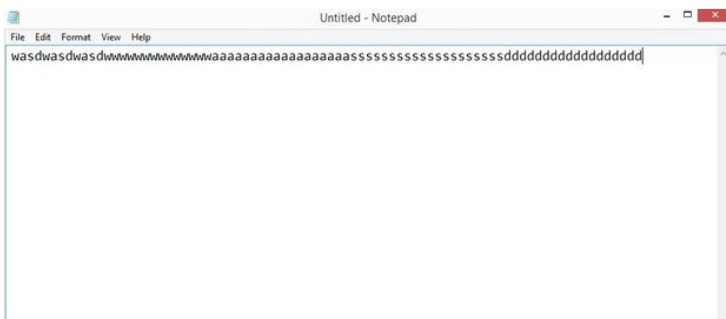
You will see this all in green if not close FLIP and return to **step 12** Arduino into DFU mode.



Step 17: How to Check If the Flash Worked.

From now on the Arduino is a HID Keyboard Congratulations!

The only thing now is that the Arduino IDE does not support your Arduino anymore. So to check if your HID keyboard works we open Notepad, or any other text editor.



Step 18: Want Your Arduino Back :(

You can do that by repeating the steps after **Step 12** Arduino into DFU mode, only then chose **Arduino-usbserial-uno.hex** instead of arduino-keyboard.hex



Nice job! Did you build that whole control panel?



Thank you Penology,

No that control panel was made by picklesandicecream. (hé posted it on Reddit)

Right now I am planning a new instructable as a follow up of this one, for making a control box! :) (also planning to put in some lasercut or cnc files for every body to use).

Cheers,

Eward



Sounds cool! I look forward to seeing it :)