# How to turn the ATMEGA chip of an Arduino Uno and Arduino Mega into a HID keyboard device.

Eward Hage

Interaction design

Hogeschool voor de kunsten Utrecht

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 a special firmware trick. After some research and not supported software I have put in an ArduinoHIDKeyboard.rar file with this PDF for all the requirements needed to perform this action.

**NOTE:** This is also a very basic Arduino coding tutorial, you can make it as advanced as you like

Examples of uses:


https://www.youtube.com/watch?v=Z7Sc4MJ8RPM


https://www.youtube.com/watch?v=NQaogjVaQek

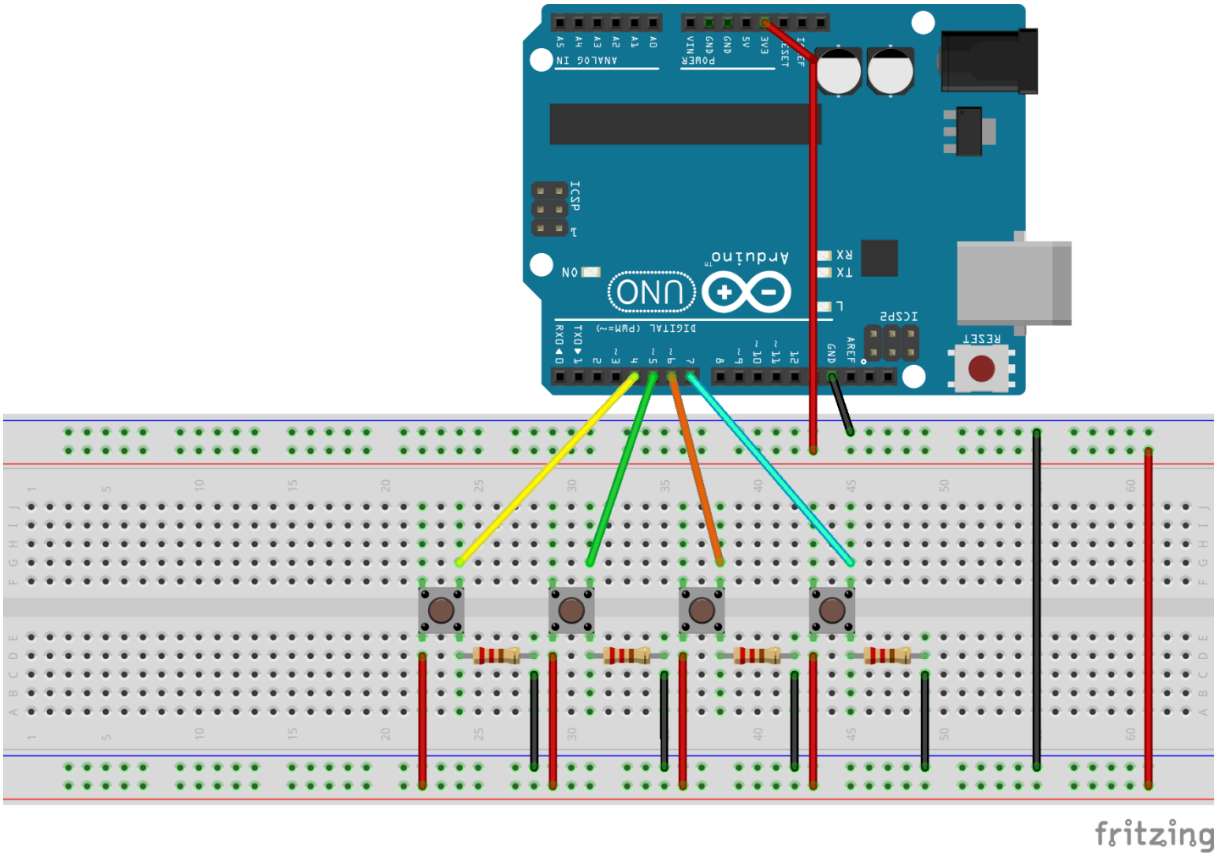
Custom controllers compatible with all games on PC, Xbox or PlayStation if keyboard is supported.

But basically you can make all sorts of controller stuff with this trick!

# Contents

## Setup of the Arduino

The setup of the Arduino hardware is the same as we are used to developing with the Arduino platform. For this example we use 4 push buttons as seen in Pic 1.




Pic 1. 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 pushbuttons onto the Digital-pins. The resistors used for this example are 1k ohm resistors.

# How to code a key press onto the Arduino

Coding is done using the Arduino IDE, there are no extra special requirements needed except for the Output requirements of a HID device.

## Step 1) Initialize a buffer

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

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

## Step 2) Define pins

```
#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
```

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

## Step 3) Void Setup() and Baud rates

```
void setup() {
  Serial.begin(9600); // Setup Serial communication

  pinMode(PIN_W, INPUT);
  pinMode(PIN_A, INPUT);
  pinMode(PIN_S, INPUT);
  pinMode(PIN_D, INPUT);
}
```

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.

The Baud rate can vary between (insert baud rates) this depends on how quick you want your bits to be send. More info on Baud rate this reference link:
https://www.arduino.cc/reference/en/language/functions/communication/serial/begin/

## Step 4) Write a key release function.

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

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.

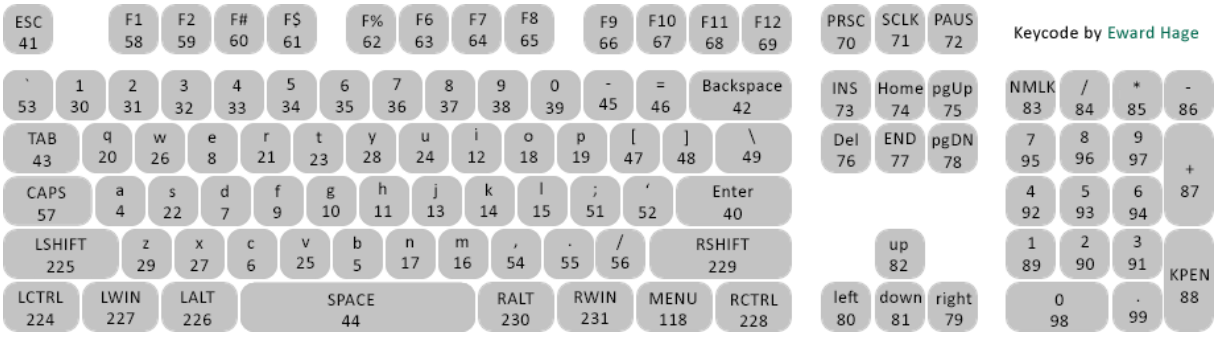
**Note:** Without this step you're HID keyboard can start sending button inputs but it will never stop sending bits until the HID keyboard (Arduino) USB is unplugged.

## Step 5) Writing a key press in the Loop()

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();
  }
}
```

To send a key press we need to access the keyboard buffer, and put in our key code value. See Pic 2 for Key codes.



Pic 2 Key code image.

## Useful tip for Debugging your device

For debugging purposes I recommend putting a Serial.println("W pressed") after the Serial.write(buf,8) Before flashing the Arduino with FLIP. This way, with the serial monitor of the Arduino IDE you can debug in the Arduino environment.

**NOTE:** Remove the Serial.println("W pressed") before updating the firmware to the keyboard. Serial.println function can cause interference with the Serial.write thus making the Serial communication not working.

## Example code

```
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
}
```

# Installing and Setting up FLIP 3.4.7

Flip is a firmware hex file pusher developed by ATMEL for their chipsets. They do not support flip anymore but it works like a charm.

## Step 1) Install and open FLIP

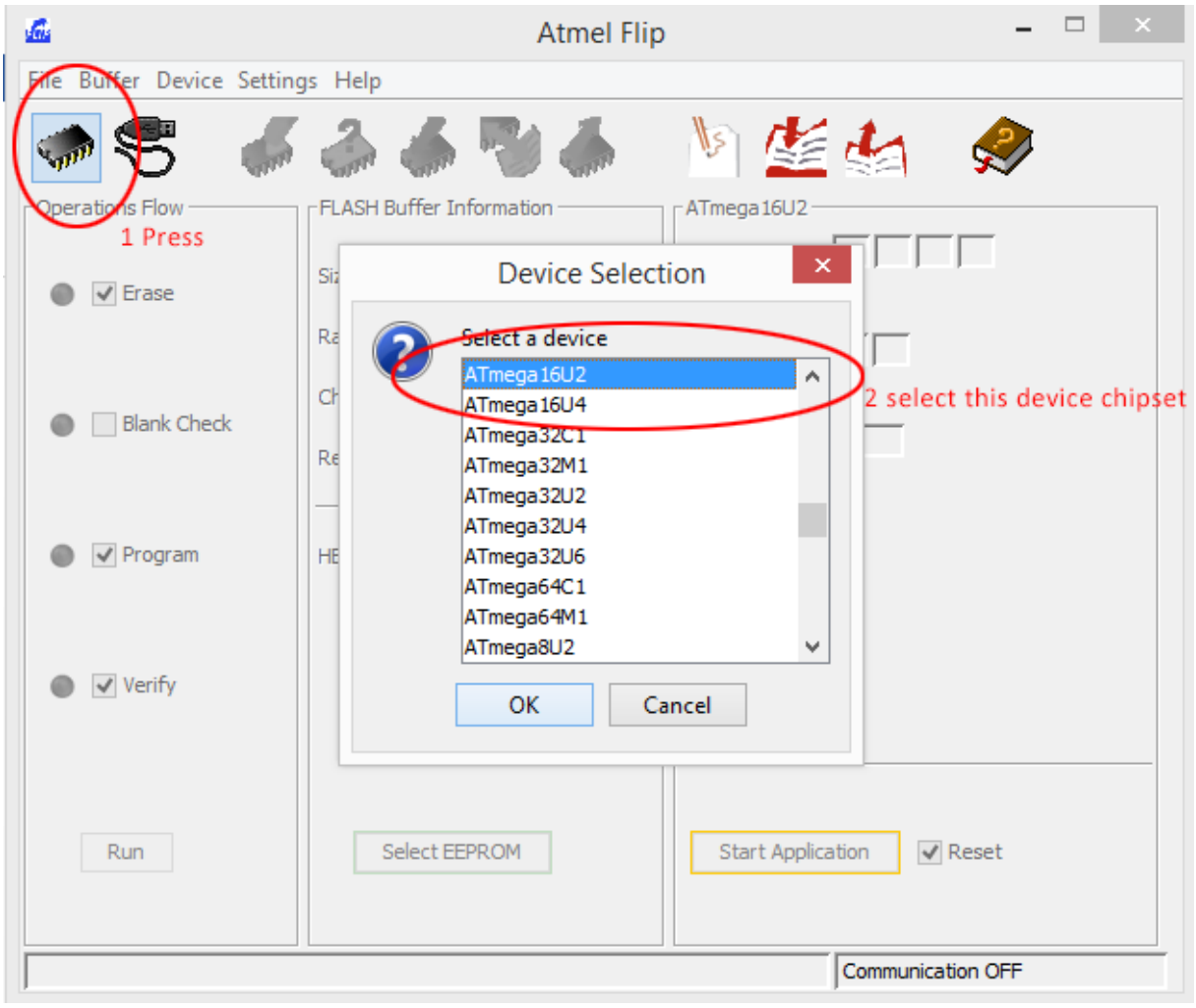
To install FLIP please install JRE - Flip Installer - 3.4.7.112.exe provided in the ArduinoHIDKeyboard.rar

Then start it up.

**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=KQ9BjKjGnIc

Open up FLIP

## Step 2) Chose the device Chipset



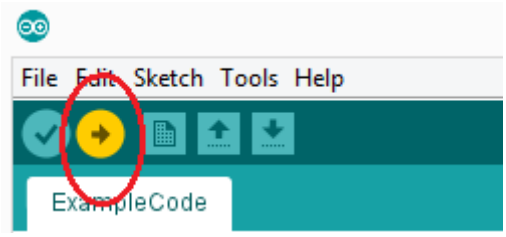

**Press** the Highlighted chip button, then **Select** ATmega16u2 and then the button OK. As this is the chipset for the Arduino Uno R3 and the Arduino Mega R3.

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

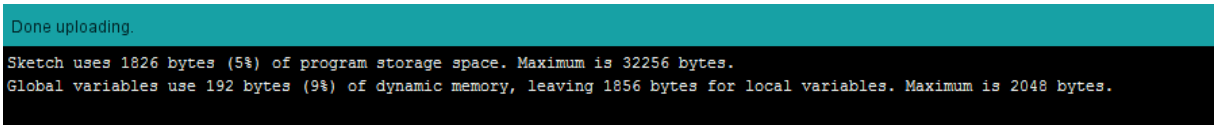
# How to flash the Arduino using FLIP 3.4.7

## Step 1) Upload your Arduino code

Upload your code or the examplecode given by pressing the upload button in the Arduino IDE




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**.

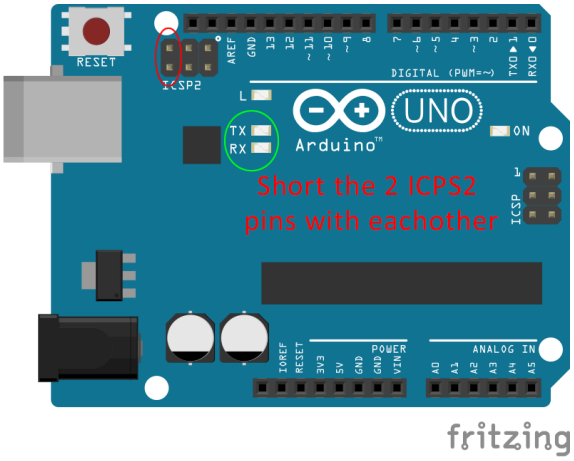
## Step 2) Put Arduino onto DFU mode

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

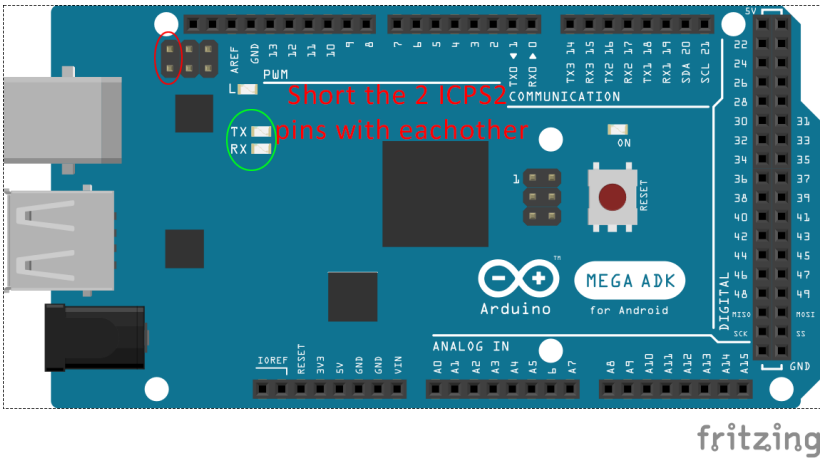
More info on this reference link: [https://www.arduino.cc/en/Hacking/DFUProgramming8U2](https://www.arduino.cc/en/Hacking/DFUProgramming8U2)

To do this short the 2 ICSP2 pins as done in the image

Example for the Arduino Uno



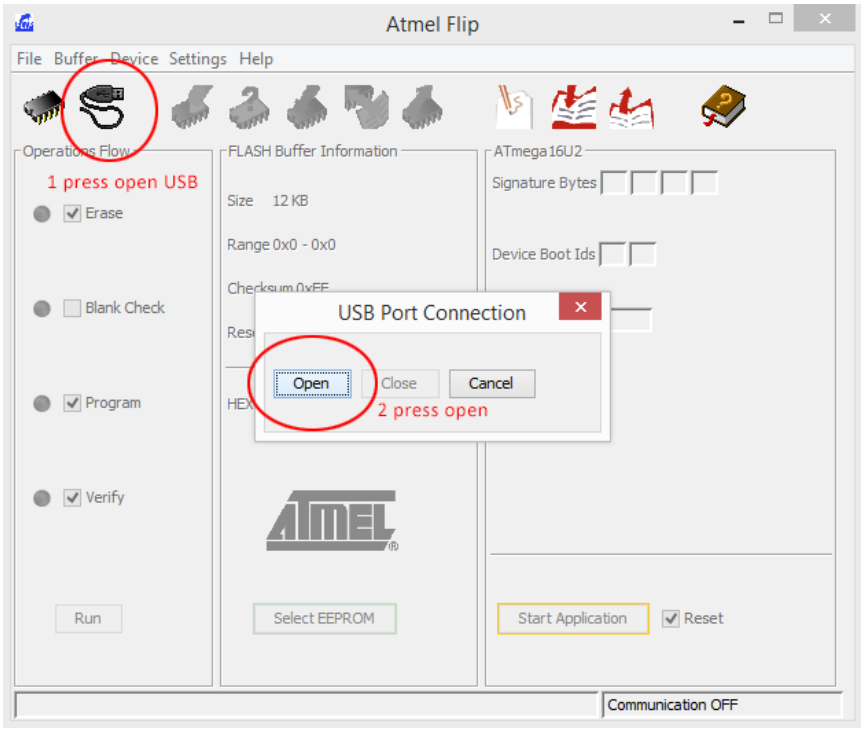
Example for the Arduino Mega




In both case the TX and RX led in the green circle will flash and you will hear a USB disconnect sound,
**NOTE:** if not unplug and replug the Arduino USB and repeat the process of shorting the 2 ICSP2 pins.

## Step 3) Open FLIP USB

Press the USB icon in FLIP and then open.

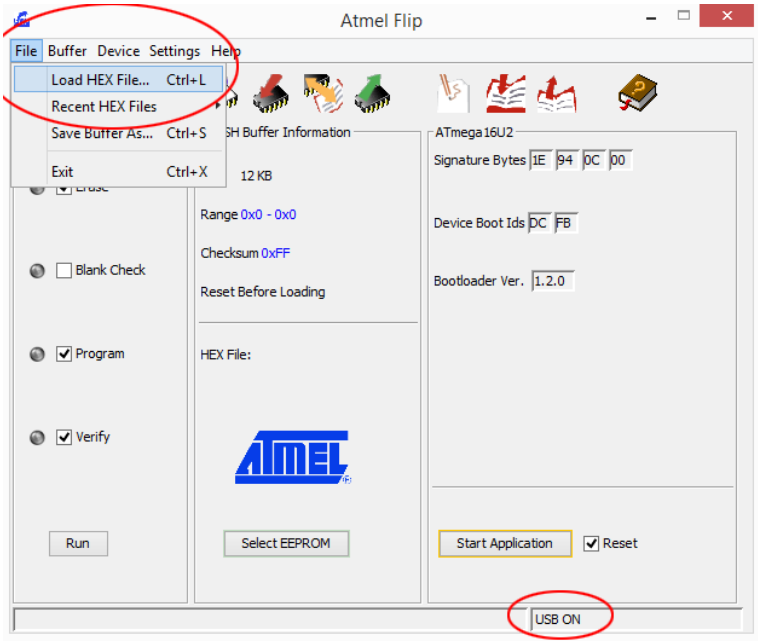


## Step 4) Parse the firmware HEX

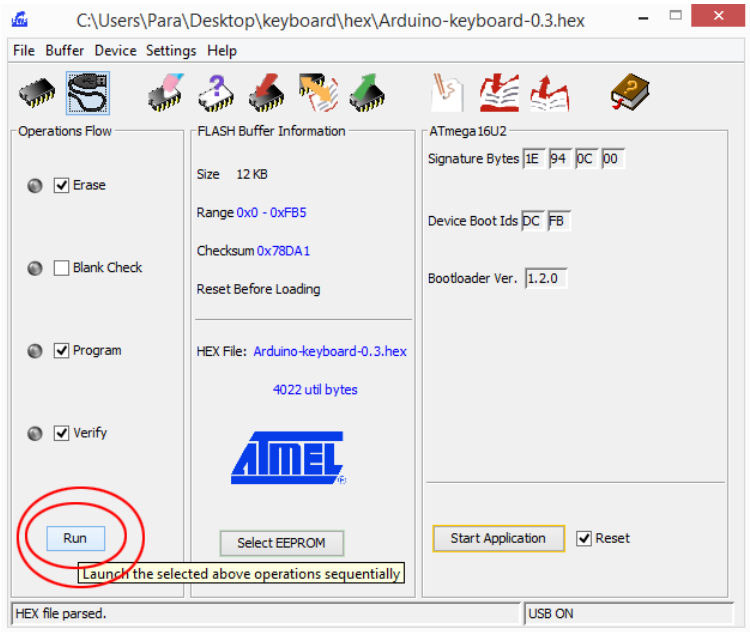
You will see at the bottom USB ON, this is that the Arduino is connected through DFU Mode.

Select the Arduino-keyboard.hex file,

or if you want your Arduino Back select Arduino-usbserial-uno.hex

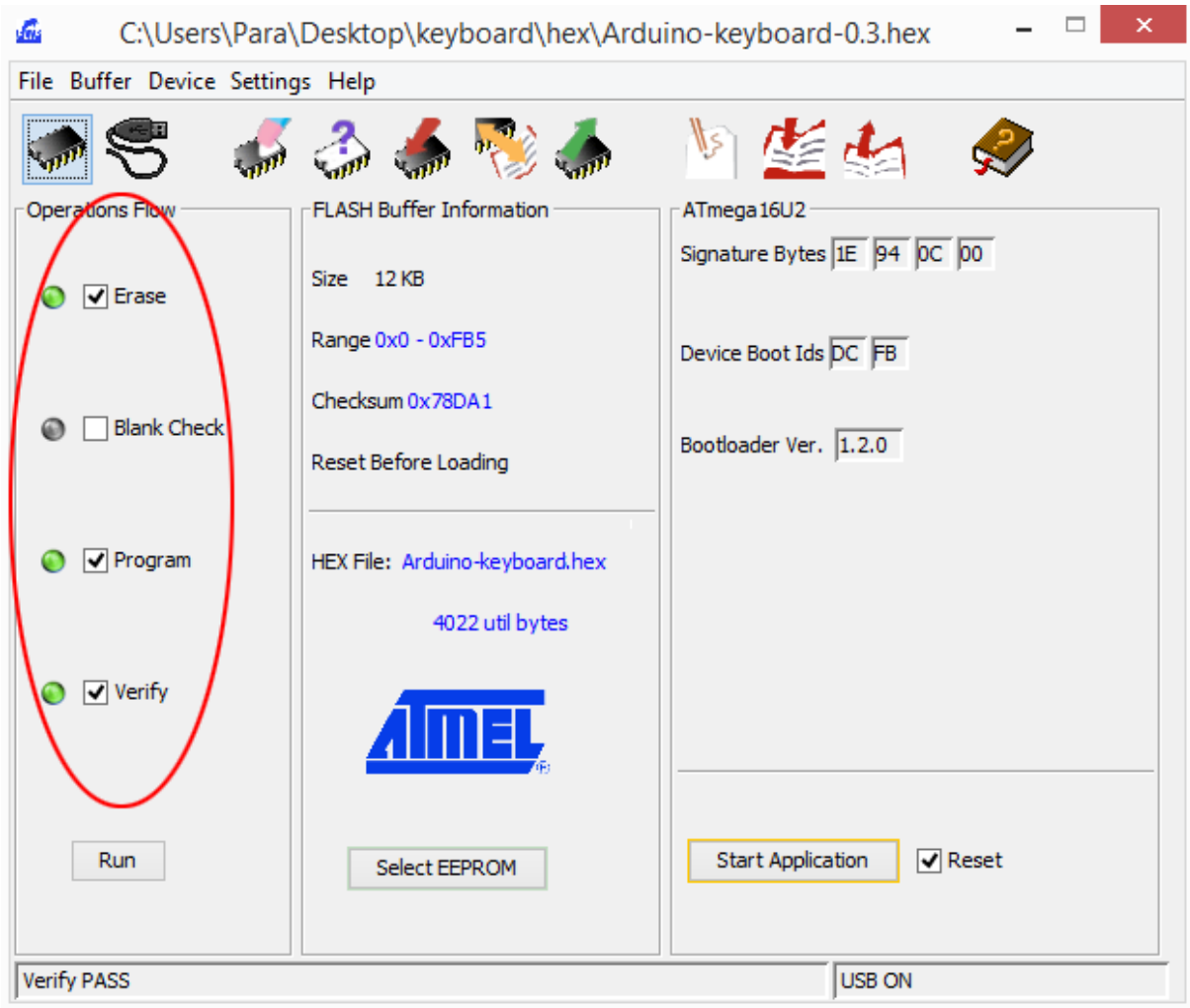



And press Run (NOT START APPLICATION)!



## Step 5) Unplug and replug USB

You will see this all in green if not close FLIP and return to step 3 Open FLIP USB

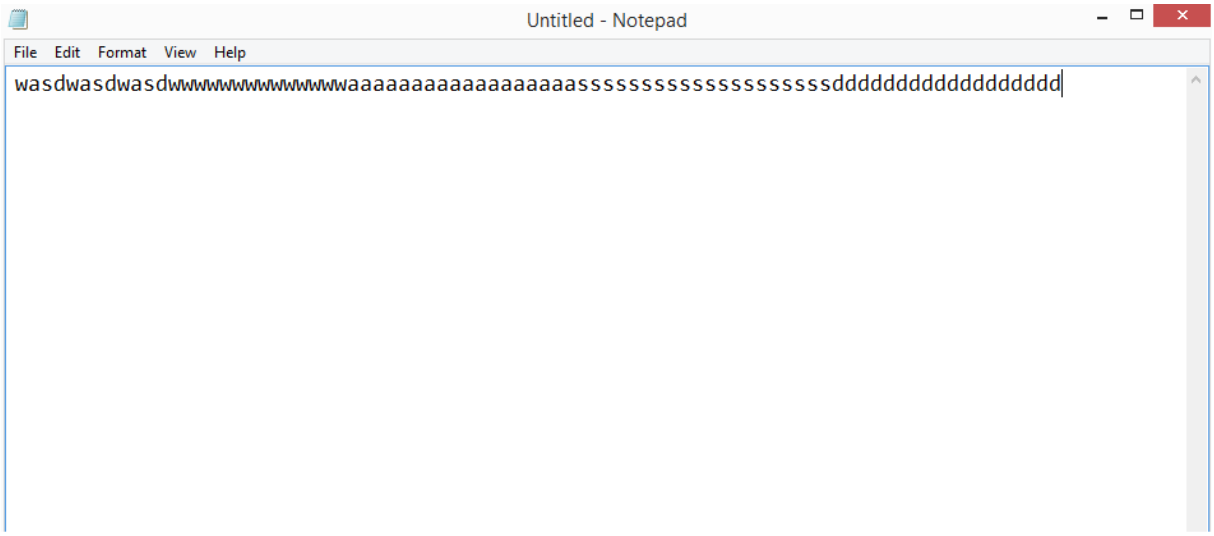



Then unplug and replug your new Keyboard device.

## 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.




You can also play games with your keyboard or put volume up and down with macro's you name it you have the power to do it.

## How to flash the HID Keyboard into a Arduino using FLIP 3.4.7

But Eward I want my Arduino back! Ok ok calm down.

You can do that by repeating the steps of **chapter how to flash you Arduino,** only then in **step 4** chose Arduino-usbserial-uno.hex instead of arduino-keyboard.hex