Anh Nguyen

Assistant Professor of Computer Science, Auburn University

Hi! I'm Anh. I completed my Ph.D. at the University of Wyoming, working with Jeff Clune and Jason Yosinski. I've been an Assistant Professor at Auburn University since 8/2017. My research focuses on improving the robustness and interpretability of neural networks.

Turning your Arduino Uno R3 into an USB mouse

by **Anh** on January 30, 2014 in **Arduino**, **Hardware**, **User Interface**

Tonight I've just turned my Arduino Uno R3, which runs the Atmega16U2 chip, into an USB mouse on my Ubuntu 12.04. I thought this guide might be helpful to those who are looking for a solution with similar setup as mine. Basically, this method uses the special USB protocol called Device Firmware Update (DFU) to program the Atmega16U2 chip.

In this demo, we will flash the generic USB HID mouse firmware by Darran. And first of all, you need to download the two firmware files:

- *Arduino-usbserial-uno.hex*
- *Arduino-mouse-0.1.hex*

from here.

Next step is to install *dfu-programmer*, the tool to reset the firmware for Atmega16U2, on your system. Download and install the dfu-programmer 0.6.2. (This version is confirmed to support Atmega16U2). Or you can try using *apt-get* but it did not give me the correct version.

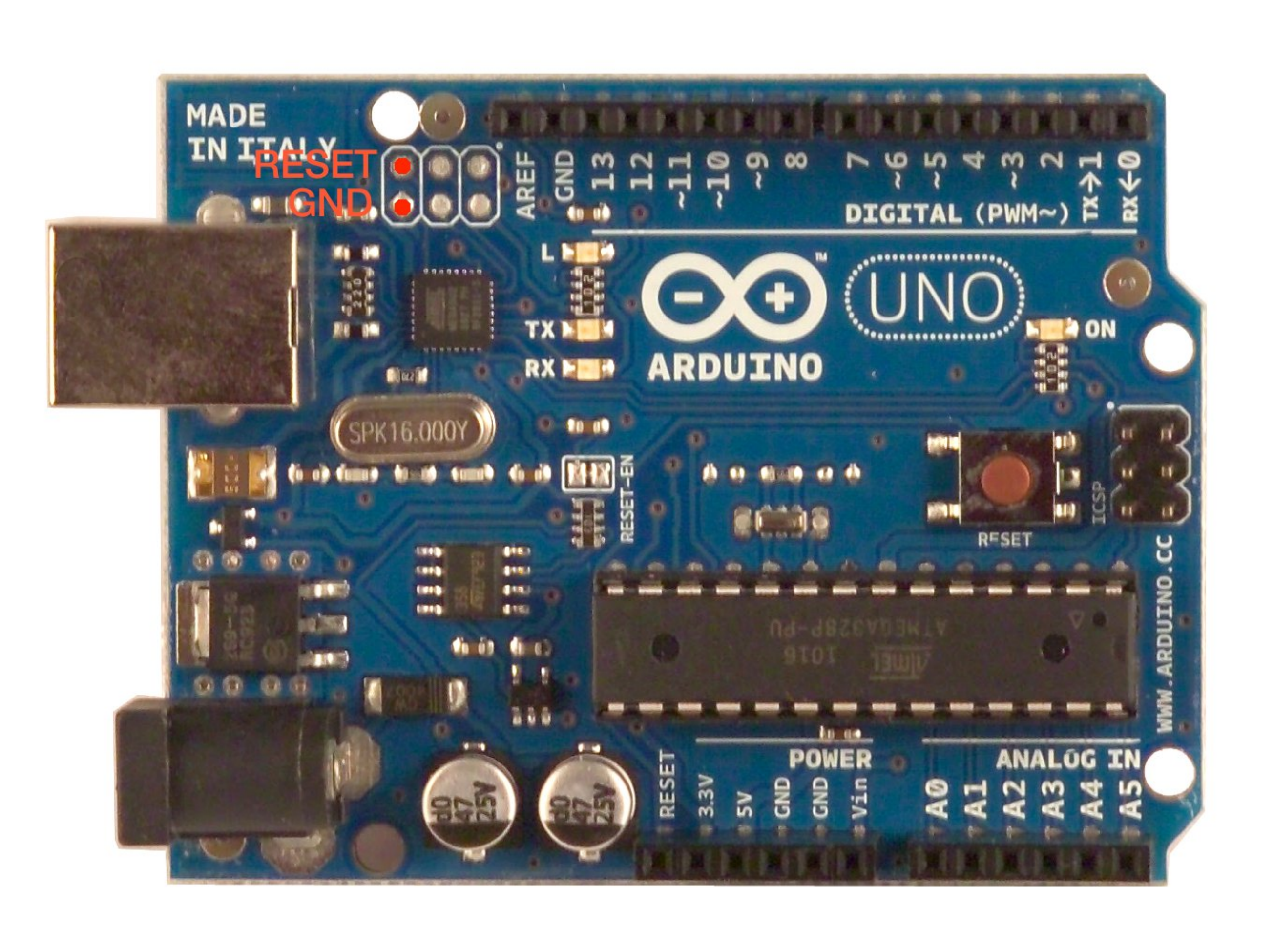
```
sudo apt-get install dfu-programmer dfu-util
```

Now to program the firmware for Atmega16U2, we need to follow the procedure below:

1. Flash Arduino-usbserial-uno.ino bootloader with dfu-programmer (erase/flash/reset)
2. Plug cycle the Arduino (eg. Unplug the Arduino USB cable; wait 1-2 seconds; and plug it back in)
3. Flash firmware with your own sketch using Arduino IDE (eg. Upload an Arduino sketch as usual. However, this sketch controls the mouse as you desire)
4. Plug cycle the Arduino
5. Flash Arduino-mouse-0.1.hex bootloader with *dfu-programmer* (erase/flash/reset)
6. Done! Now plug cycle the Arduino and it should control the mouse as written in the sketch in step (3)

In details:

Step 1: Reset your chip by bridging the reset pin with the ground as guided here.



Bridging reset pin and the ground pin

Then go the the folder where you extracted the firmware files above, and perform erase/flash/reset as following:

```
sudo dfu-programmer atmega16u2 erase
sudo dfu-programmer atmega16u2 flash --debug 1 Arduino-usbserial-uno
sudo dfu-programmer atmega16u2 reset
```

Step 2: plug cycle your Arduino.

Step 3: upload the following sketch using Arduino IDE. What the below sketch does is only move the mouse in a small circle in a 1-second interval. This should be the starting sketch, and you should change it later as desired.

An example of what this sketch could do is like the Gyration Air Mouse which reads the values reported from its Inertial Measurement Unit, and moves the mouse on the screen accordingly.

```
/* Arduino USB Mouse HID demo */

/* Author: Darran Hunt
 * Release into the public domain.
 */
```

```
struct {
    uint8_t buttons;
    int8_t x;
    int8_t y;
    int8_t wheel;   /* Not yet implemented */
} mouseReport;

uint8_t nullReport[4] = { 0, 0, 0, 0 };

void setup();
void loop();

void setup()
{
    Serial.begin(9600);
    delay(200);
}

/* Move the mouse in a clockwise square every 5 seconds */
void loop()
{
    int ind;
    delay(1000);

    mouseReport.buttons = 0;
    mouseReport.x = 0;
    mouseReport.y = 0;
    mouseReport.wheel = 0;

    mouseReport.x = -2;
    for (ind=0; ind<20; ind++) {
    Serial.write((uint8_t *)&mouseReport, 4);
    Serial.write((uint8_t *)&nullReport, 4);
    }

    mouseReport.x = 0;
    mouseReport.y = -2;
    for (ind=0; ind<20; ind++) {
    Serial.write((uint8_t *)&mouseReport, 4);
    Serial.write((uint8_t *)&nullReport, 4);
    }

    mouseReport.x = 2;
    mouseReport.y = 0;
    for (ind=0; ind<20; ind++) {
    Serial.write((uint8_t *)&mouseReport, 4);
    Serial.write((uint8_t *)&nullReport, 4);
    }

    mouseReport.x = 0;
    mouseReport.y = 2;
    for (ind=0; ind<20; ind++) {
    Serial.write((uint8_t *)&mouseReport, 4);
    Serial.write((uint8_t *)&nullReport, 4);
    }
}
```

Step 4: plug cycle your Arduino.

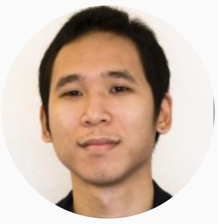
Step 5: Reset your chip again as in the step 1, and and perform erase/flash/reset as following:

```
sudo dfu-programmer atmega16u2 erase
sudo dfu-programmer atmega16u2 flash --debug 1 Arduino-mouse-0.1.hex
sudo dfu-programmer atmega16u2 reset
```

Finally: plug cycle your Arduino, and test your 'new mouse' ! It should move around, and around,.. and around in circle. :)

Cheers,

Anh



About Anh

View all posts by Anh →

‹ How to use Bullet Physics engine in Vrui applications

A modular wearable 3D input device ›