

2016

# Mouse Control Using Arduino Uno R3

Hend Almorsi

Arduino uno dose not have USB HID(human interface device) capability unlike Arduino Leonardo the leonardo based on a USB capable chip that can be used as a USB device a mouse, a keyboard or whatever. In this project we will use Arduino uno as a USB HID and use it as a mouse controller.



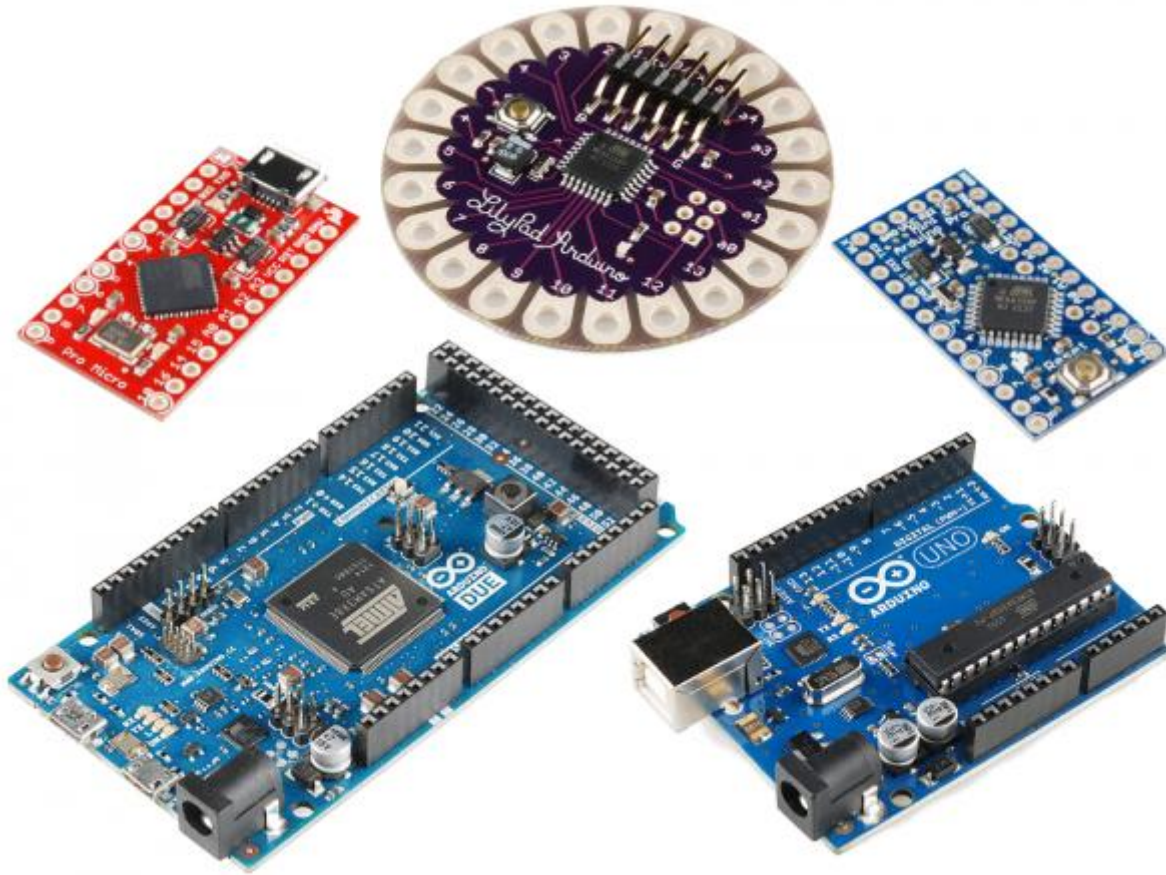
## Table of Contents

Project Idea.....	3
Purpose.....	4
How to do it.....	4
The Code.....	6
The Component.....	8
KEY Button Define.....	9
Important to note that.....	9
References.....	10
For the Arduino and project .....	10
For the Joystick Shield.....	10



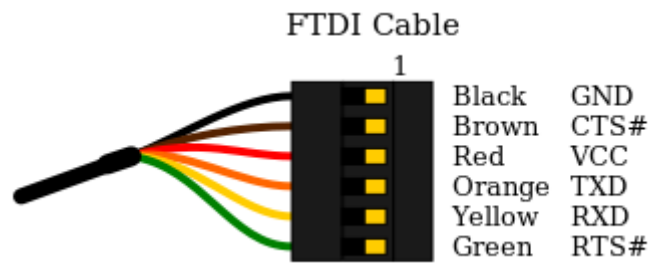
## Project Idea

There are a lot of different Arduino boards out there. Each one has its own capabilities to be used in suitable purposes.



They also differ in their programming interfaces(Which is how you hook up the Arduino board to your computer for programming). Some boards have a USB jack on-board so that all you need to do is plug them into a USB cable. Others have a header available so that you can plug in an FTDI Basic breakout or FTDI Cable. Other boards, like the Mini, break out the serial pins for programming but aren't pin-compatible with the FTDI header. Any Arduino board that has a USB jack on-board also has some other hardware that enables the serial to USB conversion. Some boards, however, don't need additional hardware because their microcontrollers have built-in support for USB.





Here in our study we use Arduino Uno R3 board. It does not use the *FTDI USB-to-serial* drive chip. Instead, it features the **ATMega16U2** programmed as a **USB-to-serial converter**. Unlike the Arduino Leonardo board which uses the Native USB. The **ATMega32U4** offers build-in USB communication, eliminating the need for a secondary processor. This allows it to appear as a mouse and keyboard, in addition to being recognized as a virtual (CDC) serial / COM port.

## Purpose

Here our problem arise as we want to use the Arduino Uno as a mouse instead of buying new Arduino board(Arduino Leonardo). We are going to change the firmware on the **USB-to-serial converter** of the Uno to work also as a HID simulator.

## How to do it

To do that we will first have to get the HID library and the hoodloader bootloader. I have attached Here:



Hootloader and HID library.rar

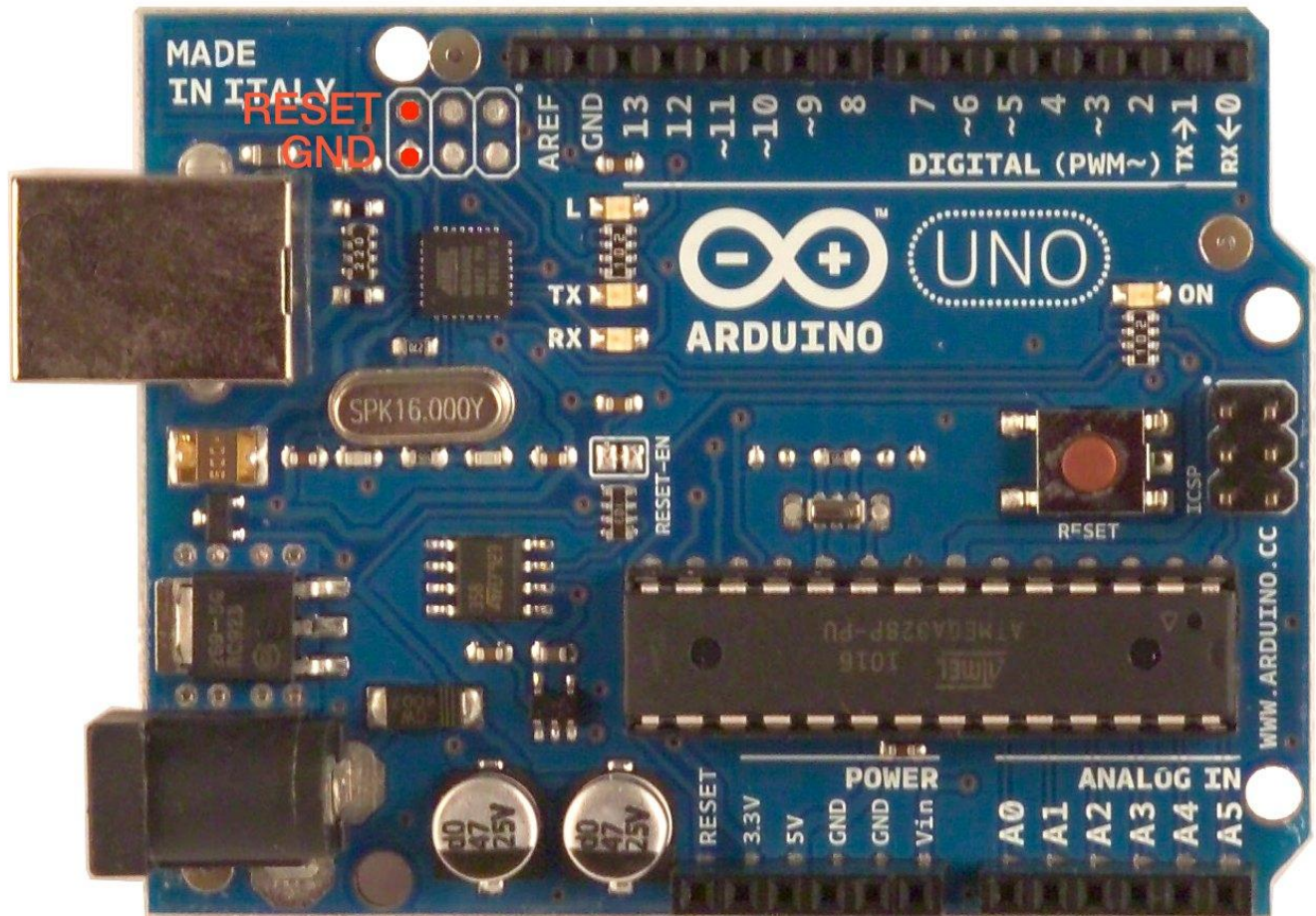
Then we have to download the Atmel's FLIP programmer from:

<http://www.atmel.com/tools/flip.aspx>



Now we get the Arduino program part and Arduino runs a software called firmware(so named because you couldn't change it once it had been programmed on the chip) that can be updated through a special USB protocol called DFU(Device Firmware Update).

To enter the Arduino into the DFU mode or to reset it we connect the RESET pin to the GROUND as shown in the picture below for a second or two.



Then we open the Atmel FLIP and select device as ATmega16U2:

Click on the USB cable symbol (Ctrl+U) and select the "USB" option. It should detect your device and now click on file and load hex file. Locate to the hoodloader folder from the rar I have attached previously and select hoodloaderX\_X.hex and click run. It should burn the flash on the chip. To check unplug the Arduino and replug it in device manager you should see hoodloader beta or something similar. Update the driver, for that select browse my





computer for device software and browse to hoodloader folder(It should prompt a security warning(or not) just accept and install).

Our final steps is to connect our Arduino and component and upload our code.

## The Code

Here we include the **HID library** the second file inside the rar attached previously.

```
// include HID library
#include <HID.h>

// set pin numbers
const int mouseButton = 3;    // input pin for the mouse pushButton
const int rightButton = 4;

const int xAxis = A0;         // joystick X axis
const int yAxis = A1;         // joystick Y axis

// parameters for reading the joystick:
int range = 5;                // output range of X or Y movement
int responseDelay = 2;        // response delay of the mouse, in ms
int threshold = range / 4;    // resting threshold(the velocity required for acceleration to become effective)
int center = range / 2;       // resting position value

void setup() {

  pinMode(rightButton, INPUT); // Right click pin
  pinMode(mouseButton, INPUT); // Left click pin

  // Starts Serial at baud 115200 otherwise HID wont work on Uno/Mega.
  // This is not needed for Leonardo/(Pro)Micro but make sure to activate desired USB functions in HID.h
  Serial.begin(SERIAL_HID_BAUD);

  // take control of the mouse:
  Mouse.begin();
}

void loop() {

  // read and scale the two axes:
  int xReading = readAxis(0);
  int yReading = readAxis(1);

  // move command
  Mouse.move(xReading, -yReading, 0); // as y axis gets inverted

  // read the mouse button and click or not click:
  // if the mouse button is pressed:
```

```
if (digitalRead(mouseButton) == LOW) {
  // if the mouse is not pressed, press it:
  if (!Mouse.isPressed(MOUSE_LEFT)) {
    Mouse.press(MOUSE_LEFT);
  }
}
// else the mouse button is not pressed:
else {
  // if the mouse is pressed, release it:
  if (Mouse.isPressed(MOUSE_LEFT)) {
    Mouse.release(MOUSE_LEFT);
  }
}
// if the right mouse button is pressed:
if (digitalRead(rightButton) == LOW) {
  // if the mouse is not pressed, press it:
  if (!Mouse.isPressed(MOUSE_RIGHT)) {
    Mouse.press(MOUSE_RIGHT);
  }
}
// else the mouse button is not pressed:
else {
  // if the mouse is pressed, release it:
  if (Mouse.isPressed(MOUSE_RIGHT)) {
    Mouse.release(MOUSE_RIGHT);
  }
}
}

delay(responseDelay);
}

/*
  reads an axis (0 or 1 for x or y) and scales the
  analog input range to a range from 0 to <range>
*/
int readAxis(int thisAxis) {
  // read the analog input:
  int reading = analogRead(thisAxis);

  // map the reading from the analog input range to the output range
  reading = map(reading, 0, 1023, 0, range);

  // if the output reading is outside from the
  // rest position threshold, use it
  int distance = reading - center;

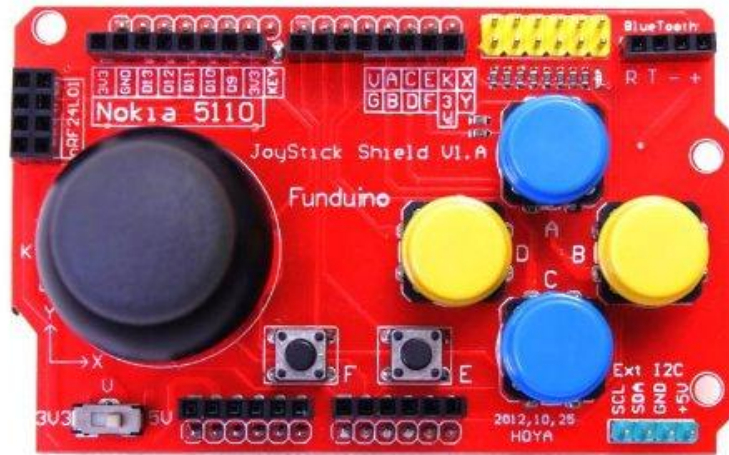
  if (abs(distance) < threshold) {
    distance = 0;
  }

  // return the distance for this axis
  return distance;
}
```



## The Component

The only component we used is the **Joystick Shield Funduino**



The connection is very easy all the pins go into the arduino board.





The Joystick has a buttons are connected as follows:

### KEY Button Define

Pin	#	KEY
D0(RX)	0	NULL
D1(TX)	1	NULL
D2	2	KEY A
D3	3	KEY B
D4	4	KEY C
D5	5	KEY D
D6	6	KEY E
D7	7	KEY F
D8	8	KEY Down

### Important to note that

When coding I found out that the push buttons on the Joystick shield is active low after searching I found it because:

If you've used a pushbutton switch before you may have noticed a resistor is normally required in order to detect a known voltage when the button is not pressed. To reduce the number of parts required this shield has been designed not to require resistors on the shield itself.

It turns out your Arduino actually has internal resistors connected to the pins inside the microcontroller. In order to use the internal resistors we need to "*enable the internal pull-up resistors*".

When a "*pull-up*" resistor is connected to a push button it means that the voltage level when the button is not pressed will be HIGH because the resistors "*pulls the voltage level up*" to HIGH when the button is not pressed. On a typical Arduino a pin that is HIGH will be at 5 volts. When the push button is pressed the pin will read as LOW because there is less resistance between the pin and ground than there is between the pin and 5 volts.



To enable a pin's pull-up resistor you first set the pin as an input and then enable the pull-up:

```
pinMode(PIN_BUTTON_RIGHT, INPUT);  
digitalWrite(PIN_BUTTON_RIGHT, HIGH);
```

In the project code I used the easy way and just change the check from HIGH to LOW

## References

### **For the Arduino and project**

- <http://www.instructables.com/id/Make-arduino-uno-work-like-leonardo/?ALLSTEPS>
- <https://www.arduino.cc/en/Hacking/DFUProgramming8U2>
- <http://anhnguyen.me/2014/01/turn-arduino-uno-r3-into-a-mouse/>
- <http://www.kevindemarco.com/2014/01/02/arduino-uno-as-a-usb-hid-interface/>
- <https://learn.sparkfun.com/tutorials/arduino-comparison-guide>
- <https://www.pubnub.com/blog/2015-04-17-arduino-board-comparisons-picking-the-right-board/>

### **For the Joystick Shield**

- [http://www.electronics.com/wiki/index.php?title=Joystick\\_Shield](http://www.electronics.com/wiki/index.php?title=Joystick_Shield)
- <https://www.sparkfun.com/tutorials/171>

