Example Projects
Workshops
Announcements

Actuators
Connections
Power
Sensors
Traces

Circuits and Code Wireless

Conductive Materials
Non-Conductive Materials

Tools

Techniques

Thinking Out Loud

Circuits and Code

- Arduino Overview
- Arduino as Bluetooth HID
- ATtiny: 7-Segment Display
- ATtiny: Capacitive Sensing
- ATtiny: Programming
- ATtiny Serial & Wireless Boards!
- ATtiny: Soft Serial
- ATtiny: Sound
- ATtiny: Soft Fade
- Multiplexed Matrix
- Controlling EL Panel and EL Wire
- EL panel/wire inverter hack
- EMF amplifier
- Heat Controlling Circuit
- LED with Light Sensor
- Lilypad XBee Shield
- pressure matrix code + circuit
- Pull-up Resistors
- RGB Colour Fade
- simple heat circuit
- Solar Powered Motor Circuit
- Sound Amplifying Circuits

- Graphing and Drawing Sensor Values
- Teensy as HID Device
- Transistor Switch
- Volume Detection
- Visualization: 2x2 Matrix
- Visualization: Drawing
- Visualization: Graph
- Visualization: Pressure Sensor Matrix
- Visualization: Touchpad
- Voltage Divider

Support the creation of content on this website through PATREON!

- About
- E-Textile Events
- E-Textile Spaces
- Newsletter
- Print & Publications
- E-Textile Shopping

SEARCH

Search

Content by Mika Satomi and Hannah Perner-Wilson

E-Textile Tailor Shop by KOBAKANT

The following institutions have funded our research and supported our work:

Since 2020, Hannah is guest professor of the Spiel&&Objekt Master's program at the University of Performing Arts Ernst Busch in Berlin

weißensee
eLab

From 2013-2015 Mika was a guest professor at the eLab at Kunsthochschule Berlin-Weissensee

From July - December 2013 Hannah was a researcher at the UdK's Design Research Lab

Smart Textiles Design Lab

From 2010-2012 Mika was a guest researcher in the Smart Textiles Design Lab at The Swedish School of Textiles

high-low tech

From 2009 - 2011 Hannah was a graduate student in the MIT Media Lab's High-Low Tech research group led by Leah Buechley

distance lab

In 2009 Hannah and Mika were both research fellows at the Distance Lab

interfaceculture

Between 2003 - 2009 Hannah and Mika were both students at Interface Cultures

share your fruits
with their seeds

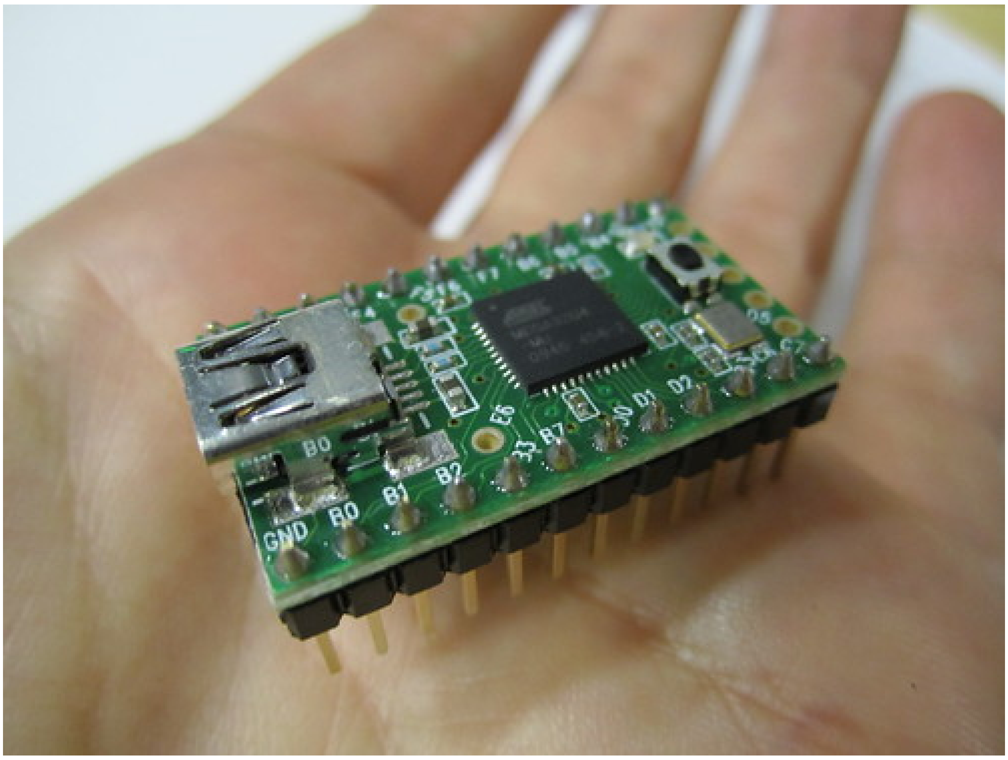
We support the Open Source Hardware movement. All our own designs published on this website are released under the Free Cultural Works definition

Circuits and Code

Teensy as HID Device

Use the Teensy board to fake Human Interface Device (HID) input. Human interface devices are recognized by your computer without need for drivers or custom software. They include computer mice, keyboards and joysticks. You can program the Teensy using arduino!!!

Also see:
>> Sewable fabric break-out board for the Teensy
>> Example: Mouse in a Hole



Using the Teensy for HID

The Teensy is a tiny microcontroller development system, similar to the Arduino. You can use the Arduino programming environment to program it, all you need is a mini USB cable.
>> http://www.pjrc.com/teensy/index.html

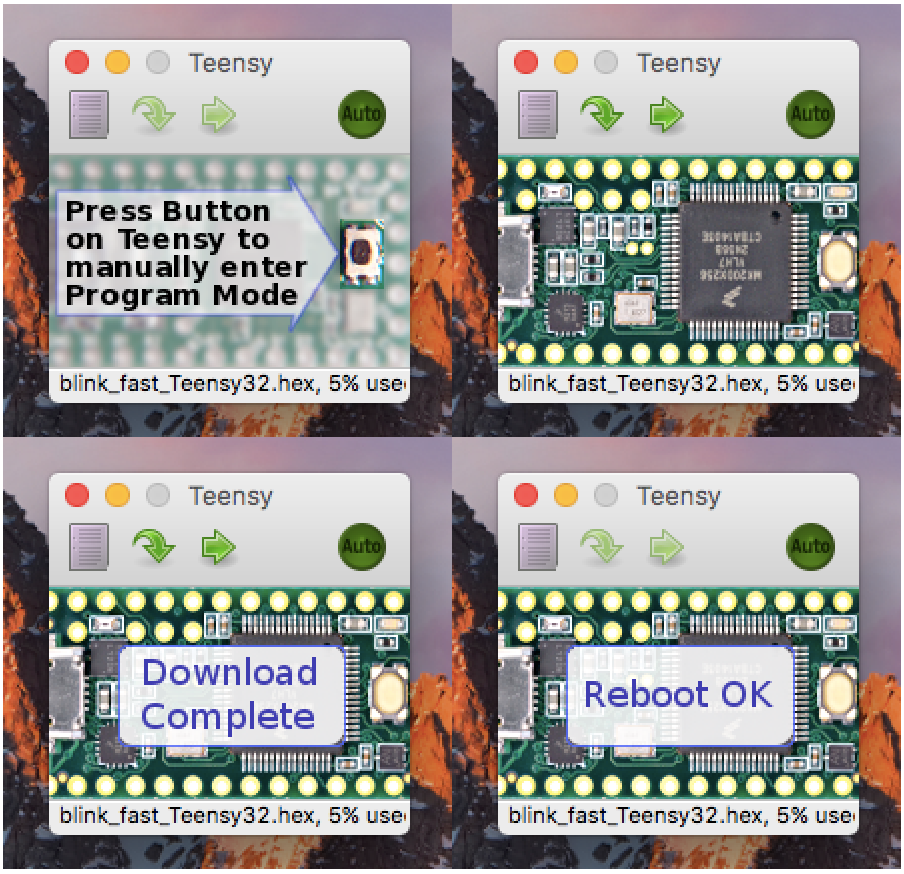
HID stands for Human Interface Device and refers to the USB-HID specification which is a defined protocol that allows devices such as computer mice, keyboards and joysticks to connect to your computer without having to download additional software.
>> http://en.wikipedia.org/wiki/Human_interface_device

The Teensy can be programmed by writing Arduino code in the Arduino programming environment and then uploading it to your board with thelp of the Teensy Loader and Teensyduino plugin. Allowing you to create Personal Interface Devices that can replace (for example) standard hard-shell computer mice.

1. Download and install the Teensy Loader application

The Teensy Loader allows you to upload new programs to your Teensy board.
Teensy Loader >> http://pjrc.com/teensy/td_download.html

Detailed instructions on how to use the Teensy Loader application can be found on the PJRC website (click on the operating system you use!):
>> http://pjrc.com/teensy/loader.html



Press Button on Teensy to manually enter Program Mode

blink_fast_Teensy32.hex, 5% use

Download Complete

Reboot OK

2. Download and install Teensyduino

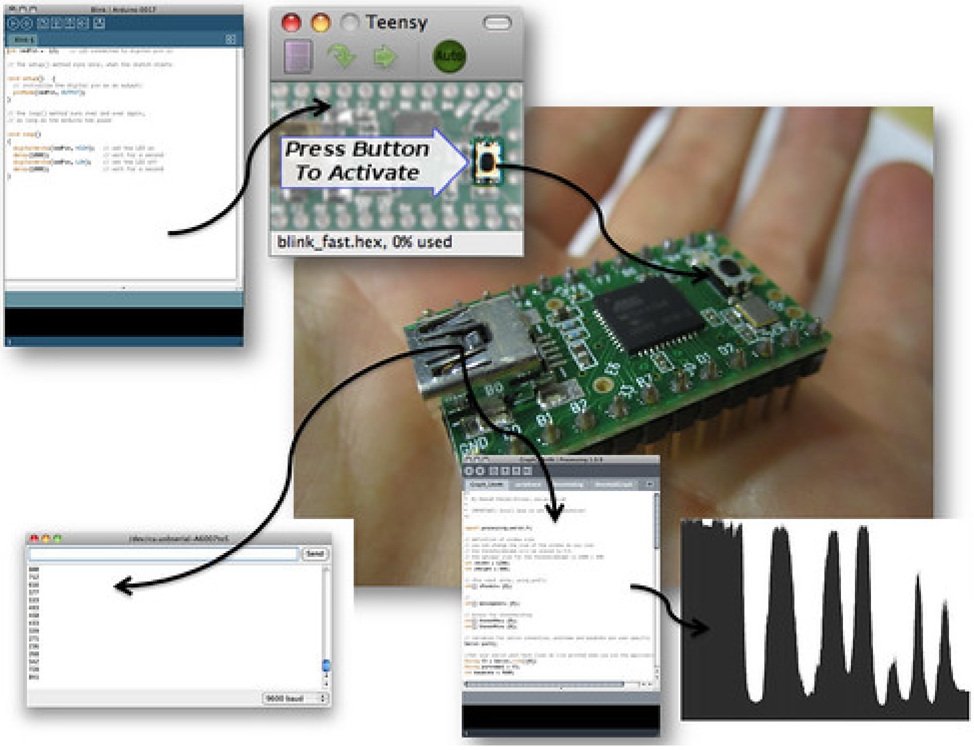
If you don't already have the Arduino software installed, you will need to download this and install it first: http://arduino.cc/en/Main/Software

Teensyduino is a software add-on for the Arduino software that allows you to select your Teensy from the menu as well as use the mouse and keyboard functions.
Teensyduino >> http://pjrc.com/teensy/td_download.html

3. Getting started with Teensy Loader

>> http://www.pjrc.com/teensy/first_use.html

Work-flow



4. Writing Arduino code for the Teensy

If you have never written Arduino code before you will want to take a moment to familiarize yourself with the environment, the code and the structure.
Getting started with Arduino >> http://arduino.cc/en/Guide/HomePage

You can basically write Arduino code as you are used to (if you are) to:
** set in- and outputs:* ***pinMode(pinNumber, INPUT/OUTPUT);***

** read analog and digital values:* ***analogRead(pinNumber); digitalRead(pinNumber);***

** set internal [pull-up resistors](#):* ***digitalWrite(pinNumber, HIGH);***

** use incoming values to trigger events:* ***if(analogRead(pinNumber) >> 100) { digitalWrite(otherPinNumber, HIGH); else digitalWrite(otherPinNumber, LOW); }***

** print incoming values out over the serial port:* ***Serial.println(analogRead(pinNumber));***

But you can also turn your Teensy into a Human Interface Device, so that after you have programmed it it will be automatically recognized as a mouse/keyboard/joystick and be able to control your mouse

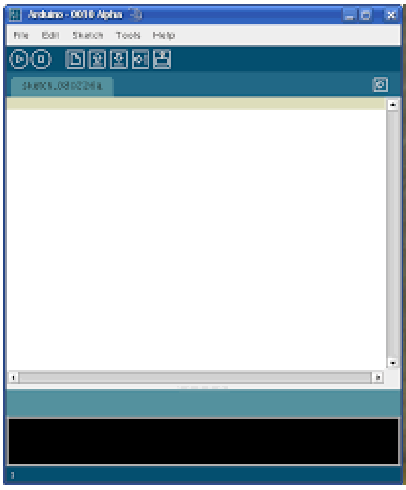
** cursor:* ***Mouse.move(x,y);***

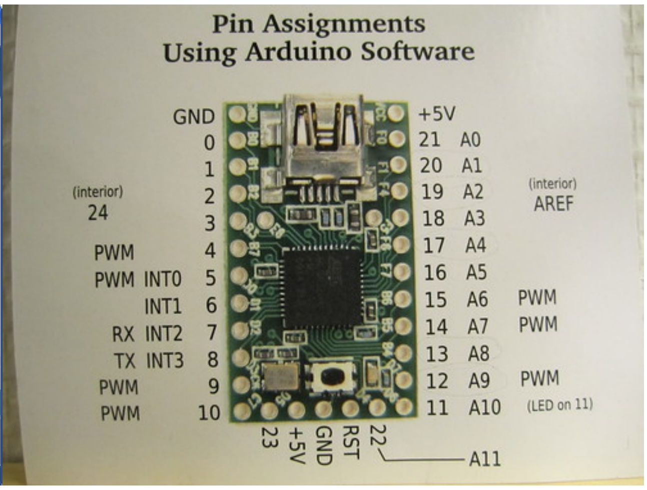
** your mouse click:* ***Mouse.click();***

** your scroll wheel:* ***Mouse.scroll(scrollSpeed);***

** your keyboard strokes:* ***Keyboard.print(”whatever”);***

!!! Here is the link to the Teensy Keyboard/Mouse library documentation:
>> http://www.pjrc.com/teensy/td_keyboard.html





5. Reading and Printing Sensor Input – Arduino Serial Monitor

Before starting to write code that will turn your Teensy project into and Human Interface Device you should work on getting smooth values from your sensors. Begin by reading the value of one of your sensors.

```
// The following code is an example of reading the analog input
// from analog pin 1 on the Teensy and prints the value over the serial port.
// Open the Arduino serial monitor to see incoming values

int pin1 = 1; //variable for storing value of first pin
int incomingValue = 0; //variable for storing incoming value

void setup() {
digitalWrite(20, HIGH); //set internal pull-up resistor
Serial.begin(9600); //begin serial communication at 9600 bauderate
}

void loop() {
incomingValue = analogRead(pin1);
Serial.println(incomingValue);
delay(100);
}
```

In order to upload this code to your Teensy you'll need to follow the following steps:

* Copy and paste the above code into your Arduino sketch

* Select your Teensy board from the **Tools** –> **Serial Port** menu: /dev/tty.usbmodem######

* Select your Teensy board from the **Tools** –> **Board** menu (Teensy 2.0 (USB Serial))

* Compile the Arduino sketch (press the play button or apple+r)

* Run the Teensy Loader application

* Press the black button on your Teensy board

* In order to see the incoming analog values open the Arduino serial monitor (apple+shift+m)

6. Mapping Sensor Input to Mouse X

!!! Before uploading this code to your Teensy using the above described steps you first need to select the HID version of your Teensy board from the Arduino **Tools** –> **Board** menu: (Teensy 2.0 (USB Keyboard/Mouse))
Upload the application and instead of printing the values out over the serial port, the Teensy will now be competing with your computer mouse or laptop touchpad. If your mouse cursor is going crazy, you can simply unplug your Teensy from the USB.

```
// The following code is an example of mapping analog input
// from analog pin 1 on the Teensy to the X value of the mouse cursor.

int pin1 = 1; //variable for storing value of first pin
int incomingValue = 0; //variable for storing incoming value

void setup() {
digitalWrite(20, HIGH); //set internal pull-up resistor
}

void loop() {
incomingValue = analogRead(pin1);
Mouse.move(incomingValue, 0);
delay(100);
}
```

7. Smoothing Sensor Input

You mouse input may be erratic and hard to controll. This is why we need to smooth the sensor input before we map it to the X position.
In order to get a feeling for the range and behavior of our sensor we need to look at the incoming values. So we want to go back to installing the first Serial.println(); application. But it can be hard to get a feeling for the sensor values from the fast moving numbers on the screen. The next step explains how you can run a Processing application that will graph your incoming sensor values so that you can look at them in more visual representation.

* For Mouse.click(); trigger you want to decide on a trigger value that when exceeded (below or above) will click the mouse and keep it clicked until that same value (or another) is reached.

* For Mouse.move(x,y); values you also want to decide on a trigger value that when exceeded calls the move function and either moves the cursor at a steady speed (see bellow example) or moves the cursor at a speed proportional to the incoming values (see next example).

```
// The following code is an example of mapping analog input
// from analog pin 1 on the Teensy to the X value of the mouse cursor.
// It sets a trigger threshold, bellow which the function is called
// When the move function is called X steadily increases by the value of speed.
// You can not accelerate the speed of the mouse in this example.

int pin1 = 1; //variable for storing value of first pin
int incomingValue = 0; //variable for storing incoming value
int speedValue = 1;
```

```
void setup() {
digitalWrite(20, HIGH);
}

void loop() {
incomingValue = analogRead(pin1);
if(incomingValue < 700) { //setting threshold Mouse.move(speedValue,0); //scaling values } }
```

* For Mouse.move(x,y); in order to accelerate you cursor speed you want to find a scale value that nicely scales your sensor values to a range that controls the speed of movement of the cursor.

```
// The following code is an example of mapping analog input
// from analog pin 1 on the Teensy to the X value of the mouse cursor.
// It sets a trigger threshold and scales the values in order
// to control the speed of movement.

int pin1 = 1; //variable for storing value of first pin
int incomingValue = 0; //variable for storing incoming value
int scale = 300;

void setup() {
digitalWrite(20, HIGH);
}

void loop() {
incomingValue = analogRead(pin1);
if(incomingValue < 700) { //setting threshold Mouse.move(incomingValue/scale, 0); //scaling values } }
```

8. Graphing Sensor Input – Processing Application

It can be easier to read the serial input if you graph it visually. The following explains how to run a Processing sketch that draws a graph based on your incoming sensor values.

You will want to download the Processing if you don't already have it installed: http://processing.org/download/

In order to read sensor input from the Teensy into Processing you have two options. You can either have the Teensy be a mouse/keyboard and then have Processing read mouse/keyboard values. Or you can have the Teensy be a serial port. In this case you need to be sure to have programmed the Teensy in the USB/serial mode, otherwise Processing will not recognize it as a serial port!

In Arduino go to: **File –> Examples –> Communication –> Graph**

The graph sketch contains Arduino code that reads the first analog input, which you will want to change to be 1 instead of 0 (for the Teensy fabric breakout board) and sends this out the serial port. In the commented-out part of this sketch there is code for both Processing and Max/MSP to read this serial input and graph it visually. Copy and paste the Processing part of the code into an empty Processing sketch and run it by pressing the play symbol (or press: apple+r). If all goes well, then it should work.
Also see: http://www.arduino.cc/en/Tutorial/Graph

If you want to send more than one value from Arduino to Processing you will either have to give your individuals data bits a prefix or put them in a format that you de-translate in Processing. Tom Igoe has a nice example of doing this that lets you select exactly how many analog inputs you have connected, both in the Arduino and Processing

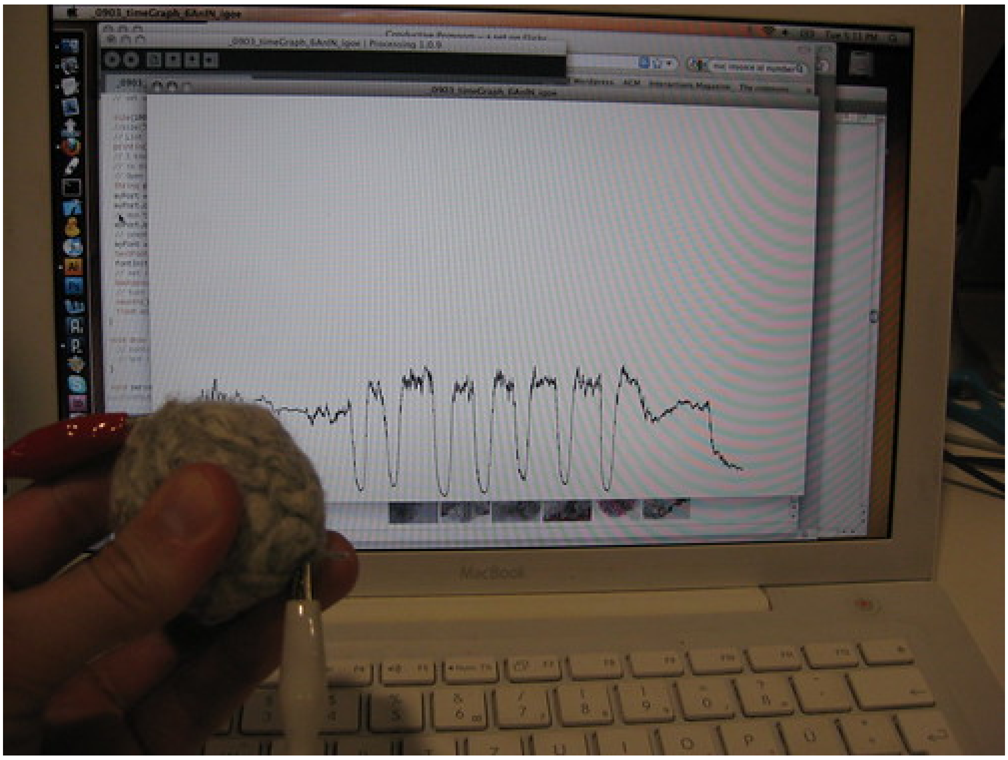
sketches. The code for this can be copied and pasted from the very bottom of this interview page:
>> http://wiki.processing.org/w/Tom_Igoe_Interview

The Arduino code from the above link needs to be modified in order to work with the Teensy fabric breakout board because we are not using inputs 0-5 but odd numbers. See the following modification for a quick solution:

```
// define the total number of analog sensors that you want to read (max 5)
#define numberOfSensors 5
// define the numbers of analog sensors that you want to read
int sensorPin[5] = {
1,3,5,7,9};

void setup() {
// initialize the serial port:
Serial.begin(9600);
//set internal pull-ups for all inputs:
digitalWrite(20, HIGH);
digitalWrite(18, HIGH);
digitalWrite(16, HIGH);
digitalWrite(14, HIGH);
digitalWrite(12, HIGH);
}

void loop() {
// loop over the sensors:
for (int thisSensor = 0; thisSensor < numberOfSensors; thisSensor++) { // read each sensor int
sensorReading = analogRead(sensorPin[thisSensor]); // print its value out as an ASCII numeric string
Serial.print(sensorReading, DEC); // if this isn't the last sensor to read, // then print a comma after it if
(thisSensor < numberOfSensors -1) { Serial.print(","); } } // after all the sensors have been read, // print a
newline and carriage return Serial.println(); delay(50); }
```



Downloads

The following code example is designed for the sewable breakoutboard we made. It set all analog and digital pins to be inputs. Reads all analog and digital inputs. And sets all the pull-up resistors to HIGH.
Fabric-breakout Code >>

Where to buy a Teensy

>> PJRC
>> Adafruit Industries

4 Comments so far

1. High-Low Tech on August 23rd, 2010

   […] >> Download: Powerpoint Presentation (17MB) >> Teensy: Workflow >> Teensy: Sewable Fabric Breakout >> Example: Mouse in a Hole >> How To Get What You Want >> […]

2. HDI | wer bastelt mit on July 17th, 2011

   […] Teensy: http://oe3xht.at/node/137 http://www.pjrc.com/teensy/ Working with Teensy: http://www.kobakant.at/DIY/?p=2497 […]

3. Lord Rybec on September 8th, 2011

   No, not joystick. You are correct that USB joysticks are part of the HID specifications that do not need a driver in modern OSs, but the Teensy does not currently emulate joysticks. As of today, it will only emulate keyboard and mouse.

   A few months ago I contacted Paul and he told me that work was being done to add joystick emulation, but there is not currently a timeline for when this will be finished and joystick support still does not exist in Teensyduino.

   Lord Rybec

4. Chicken soup on April 29th, 2013

   Hi to every body, it’s my first go to see of this webpage; this weblog carries remarkable and genuinely good stuff in support of readers.

Leave a comment

name (required)

email (will not be shown) (required)

website

□

Submit Comment

four − one = □ ⟳