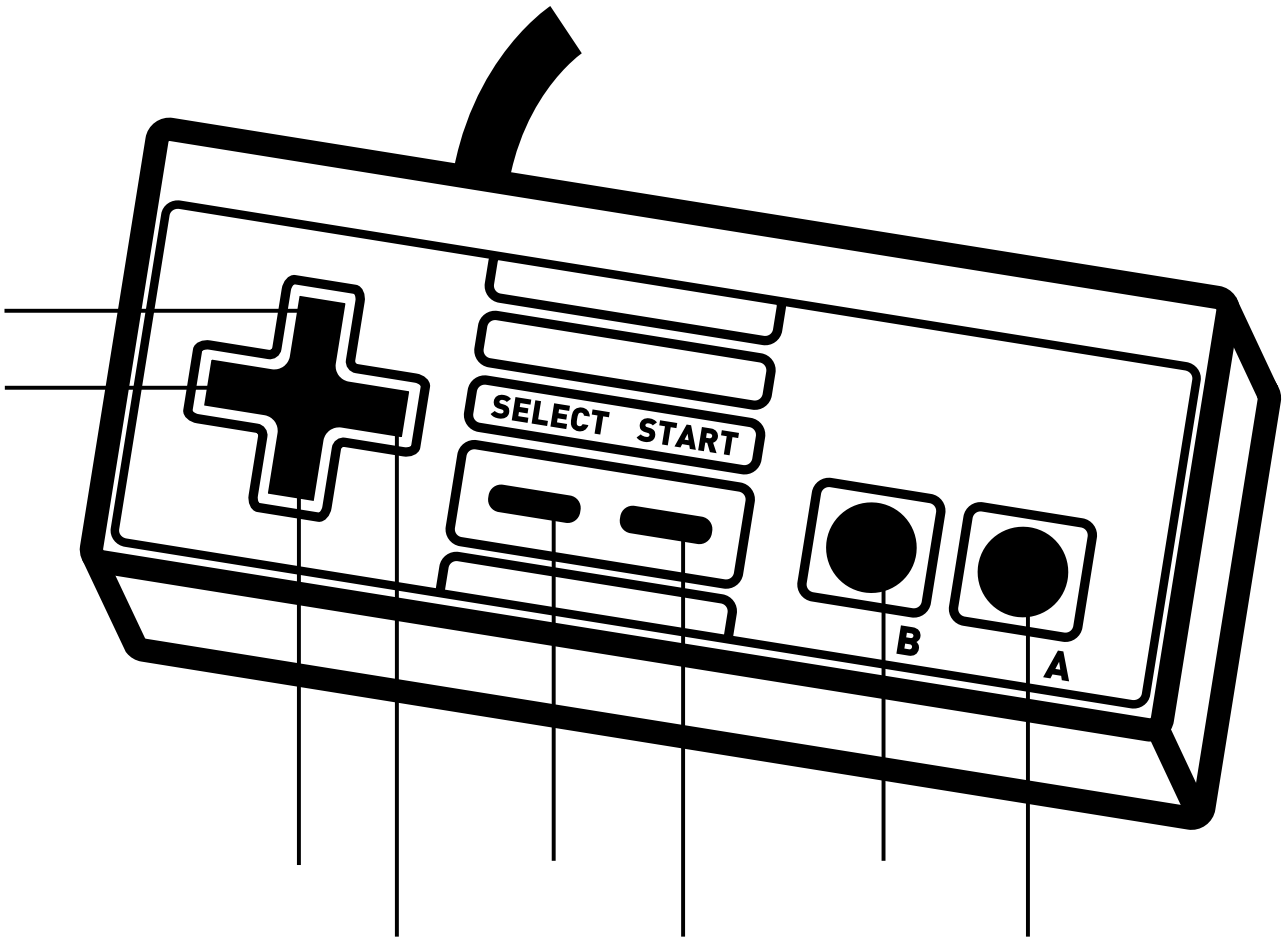


Recycling[®]

ENTERTAINMENT SYSTEM



RESWARE Beta 1.0 by www.recyclism.com
This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 3
Unported License.





ENTERTAINMENT SYSTEM

2011 RES Mini User Manual

Table of Contents

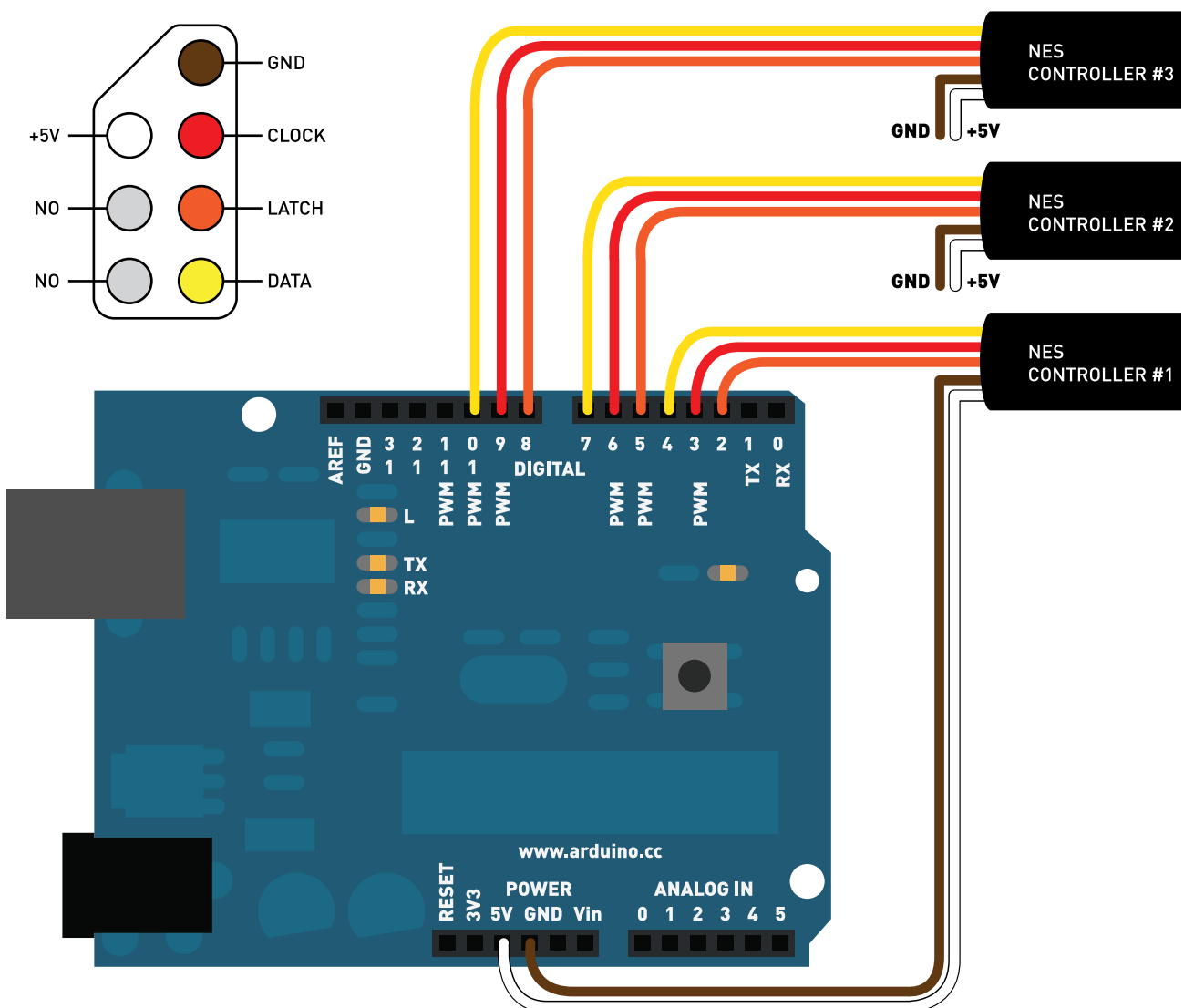
Page 3..... Hack the NES Controller
Page 4..... Arduino Code
Page 12..... RESWARE Beta 1.0 Settings
Page 16..... Additional Information

RESWARE Beta 1.0 by www.recyclism.com
This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 3
Unported License.



Hack the NES Controller

The NES Controller contains a chip (MC 14021 BCP) also called a shift register. This means with only 3 wires you can read the 8 buttons of the controller. To connect to an Arduino Board simply follow the drawing below:



RESWARE Beta 1.0 by www.recyclism.com
This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 3
Unported License.



Arduino Code

To read your controller(s) and send the state of each button to Pure Data you can use the following Code (Original code by: Sebastian Tomczak, 21 July 2007, Adelaide, Australia. Modified by: Joshua de Haan, 21 June 2009, Landgraaf, The Netherlands. Modified by: Benjamin Gaulon December 2010. Dublin, Ireland). You can also download the code directly: <http://www.recyclism.com/resmini/arduino.zip>

```
/* INITIALISATION */

// Controller 1
int latch = 2; // set the latch pin
int clock = 3; // set the clock pin
int datin = 4; // set the data in pin
char controller_data[8];

// Controller 2
int latch2 = 5; // set the latch pin
int clock2 = 6; // set the clock pin
int datin2 = 7; // set the data in pin
char controller_data2[8];

// Controller 3
int latch3 = 8; // set the latch pin
```

RESWARE Beta 1.0 by www.recyclism.com
This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 3
Unported License.



```
int clock3 = 9; // set the clock pin
int datin3 = 10; // set the data in pin
char controller_data3[8];

/* SETUP */
void setup()
{
  Serial.begin(9600);

  // controller 1
  pinMode(latch,OUTPUT);
  pinMode(clock,OUTPUT);
  pinMode(datin,INPUT);

  // controller 2
  pinMode(latch2,OUTPUT);
  pinMode(clock2,OUTPUT);
  pinMode(datin2,INPUT);

  // controller 3
  pinMode(latch3,OUTPUT);
  pinMode(clock3,OUTPUT);
  pinMode(datin3,INPUT);
}
```



```

/* CONTROLLER 1 READ */
void controllerRead()
{
    digitalWrite(latch,LOW);
    digitalWrite(clock,LOW);
    digitalWrite(latch,HIGH);
    delayMicroseconds(200);
    digitalWrite(latch,LOW);
    if (digitalRead(datin) == 1)
    {
        controller_data[0] = '1';
    }
    else
    {
        controller_data[0] = '0';
    }
    delayMicroseconds(200);
    for (int i = 1; i <= 7; i ++)
    {
        digitalWrite(clock,HIGH);
        delayMicroseconds(200);
        if (digitalRead(datin) == 1)
        {
            controller_data[i] = '1';
        }
    }
}

```



```

else
{
    controller_data[i] = '0';
}
digitalWrite(clock,LOW);
delayMicroseconds(200);
}
}

```

```

/* CONTROLLER 2 READ */
void controllerRead2()
{
    digitalWrite(latch2,LOW);
    digitalWrite(clock2,LOW);
    digitalWrite(latch2,HIGH);
    delayMicroseconds(200);
    digitalWrite(latch2,LOW);
    if (digitalRead(datin2) == 1)
    {
        controller_data2[0] = '1';
    }
else
{

```



```

    controller_data2[0] = '0';
}
delayMicroseconds(200);
for (int i = 1; i <= 7; i ++)
{
    digitalWrite(clock2,HIGH);
    delayMicroseconds(200);
    if (digitalRead(datin2) == 1)
    {
        controller_data2[i] = '1';
    }
    else
    {
        controller_data2[i] = '0';
    }
    digitalWrite(clock2,LOW);
    delayMicroseconds(200);
}

}

/* CONTROLLER 3 READ */
void controllerRead3()
{

```

RESWARE Beta 1.0 by www.recyclism.com
This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 3
Unported License.



```

digitalWrite(latch3,LOW);
digitalWrite(clock3,LOW);
digitalWrite(latch3,HIGH);
delayMicroseconds(200);
digitalWrite(latch3,LOW);
if (digitalRead(datin3) == 1)
{
    controller_data3[0] = '1';
}
else
{
    controller_data3[0] = '0';
}
delayMicroseconds(200);
for (int i = 1; i <= 7; i ++)
{
    digitalWrite(clock3,HIGH);
    delayMicroseconds(200);
    if (digitalRead(datin3) == 1)
    {
        controller_data3[i] = '1';
    }
    else
    {
        controller_data3[i] = '0';
    }
}

```



```

    }
    digitalWrite(clock3,LOW);
    delayMicroseconds(200);
}

}

/* PROGRAM */
void loop()
{
    controllerRead();
    for (int i = 0; i <= 7; i++){
        Serial.print(controller_data[i]);
        delayMicroseconds(200);
    }

    Serial.print(" ");
    delayMicroseconds(200);

    controllerRead2();
    for (int i = 0; i <= 7; i++){
        Serial.print(controller_data2[i]);
        delayMicroseconds(200);
    }
}

```



```
Serial.print(" ");
delayMicroseconds(200);

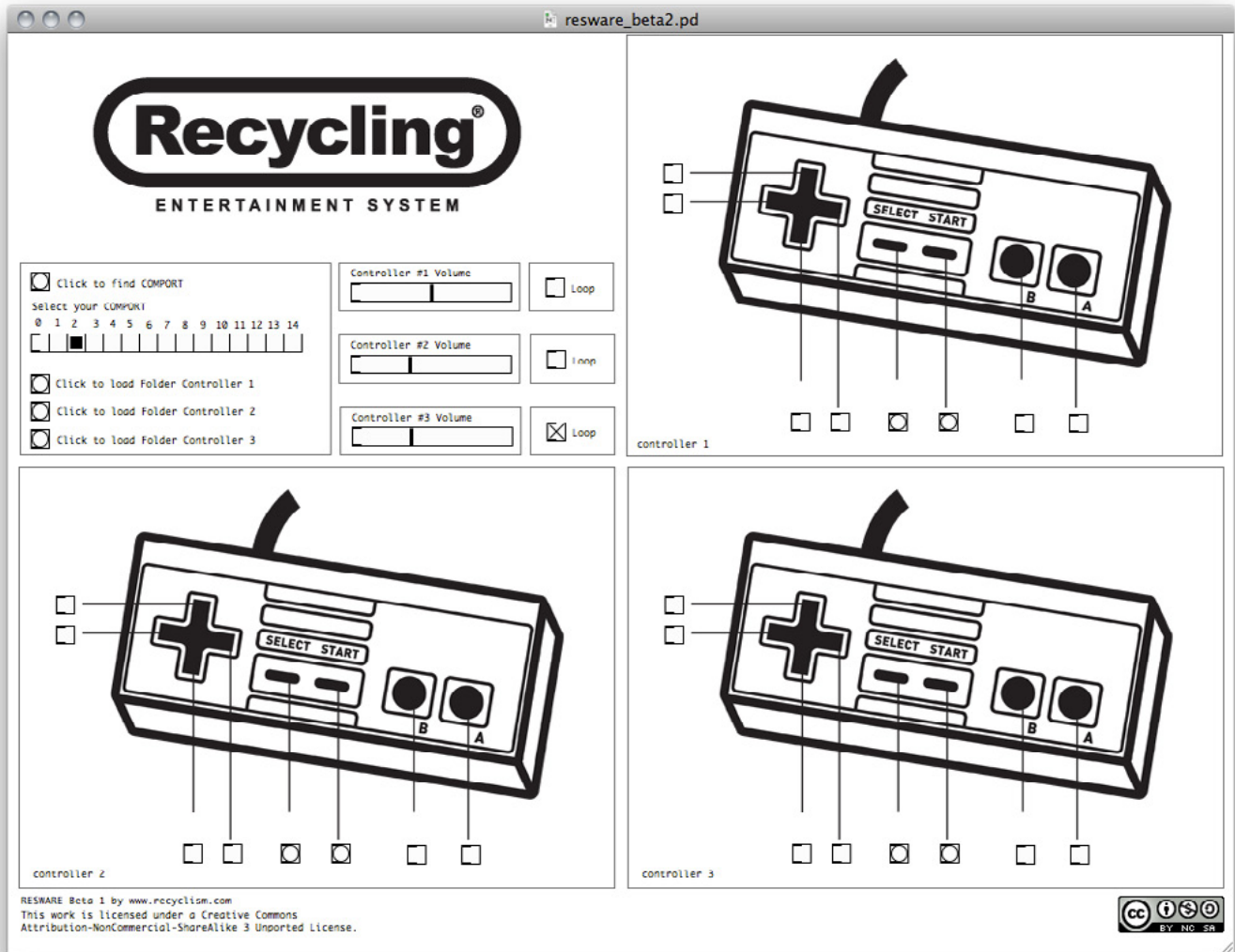
controllerRead3();
for (int i = 0; i <= 7; i++){
  Serial.print(controller_data3[i]);
  delayMicroseconds(200);
}

Serial.println();
delayMicroseconds(200);
}
```



RESWARE Beta 1.0 Settings

The RESWARE Beta 1.0 is a Pure Data based software, here is a screen shot of the Beta Version 1.0:

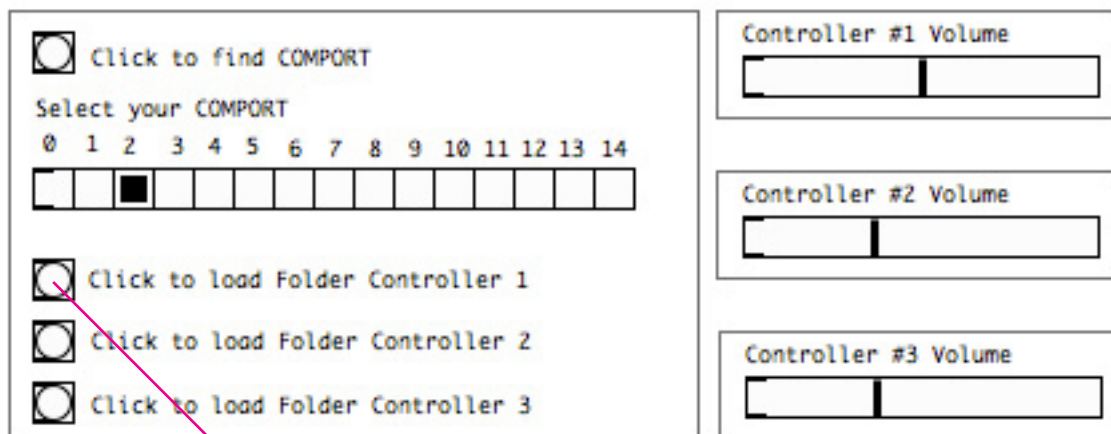


RESWARE Beta 1.0 by www.recyclism.com
This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 3
Unported License.



RESWARE Beta 1.0 Settings

The RESWARE Beta 1.0 is a Pure Data based software allowing the user to load sound libraries (audio samples). Files should be .aif or .wav files. By default the Resware loads its own sound banks if you wish to load your own files simply click on the buttons as below:

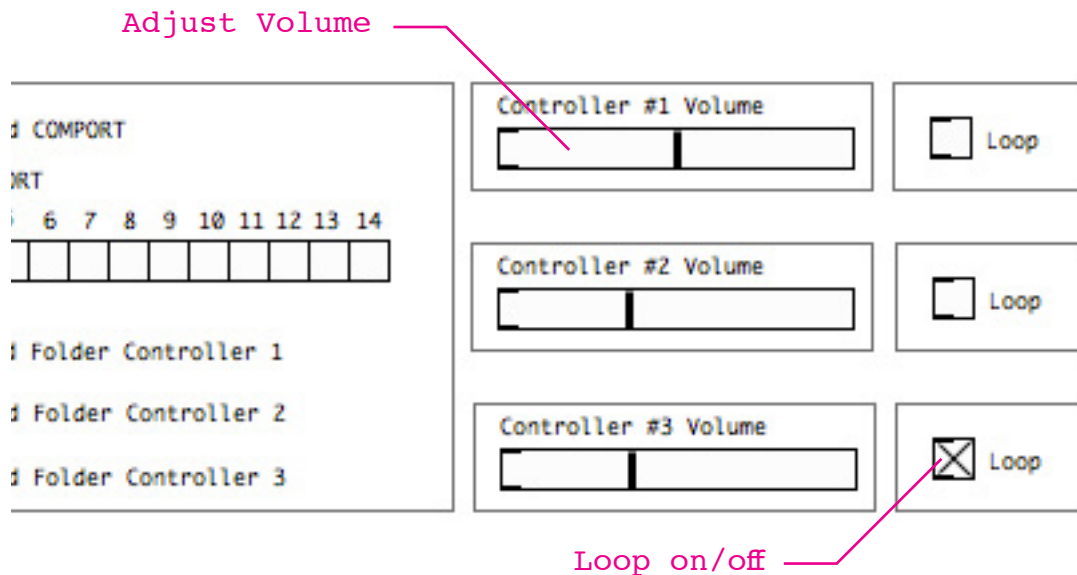


Files will be loaded in alphanumeric order, so make sure to name your files (a, b, c, etc...) or (001, 002, 003, etc...). Since each controller can play 6 samples at the time make sure to organise your samples in series of 6 (i.e. first set of sound = 001.wav, 002.wav, 003.wav, 004.wav, 005.wav, 006.wav, second set = 007.wav, 008.wav, etc...).

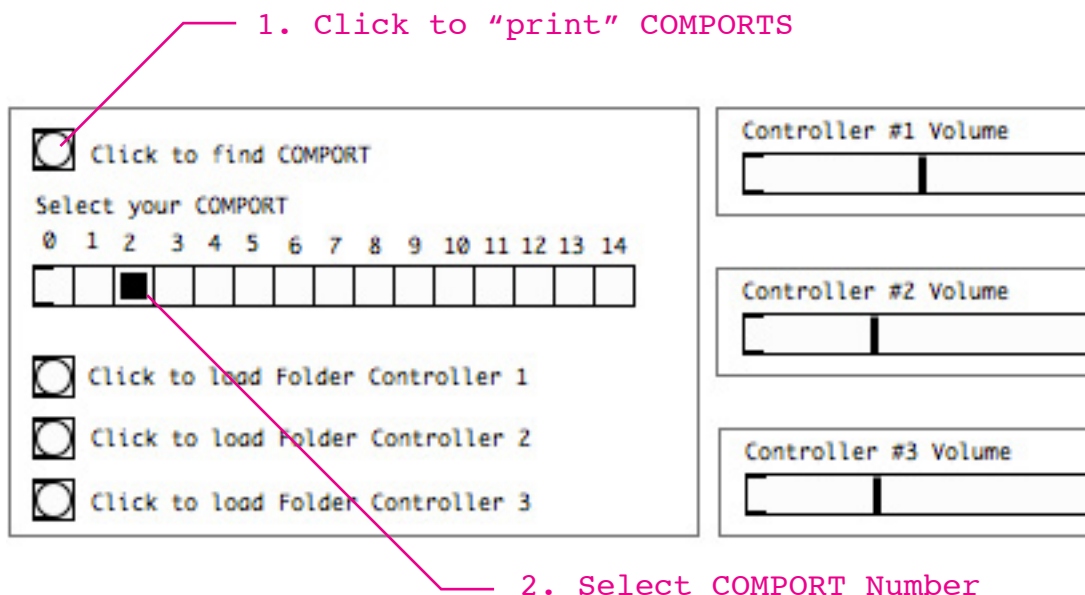


RESWARE Beta 1.0 Settings

You can adjust each controller volume individually and set the sample to play as loop or not, see below:

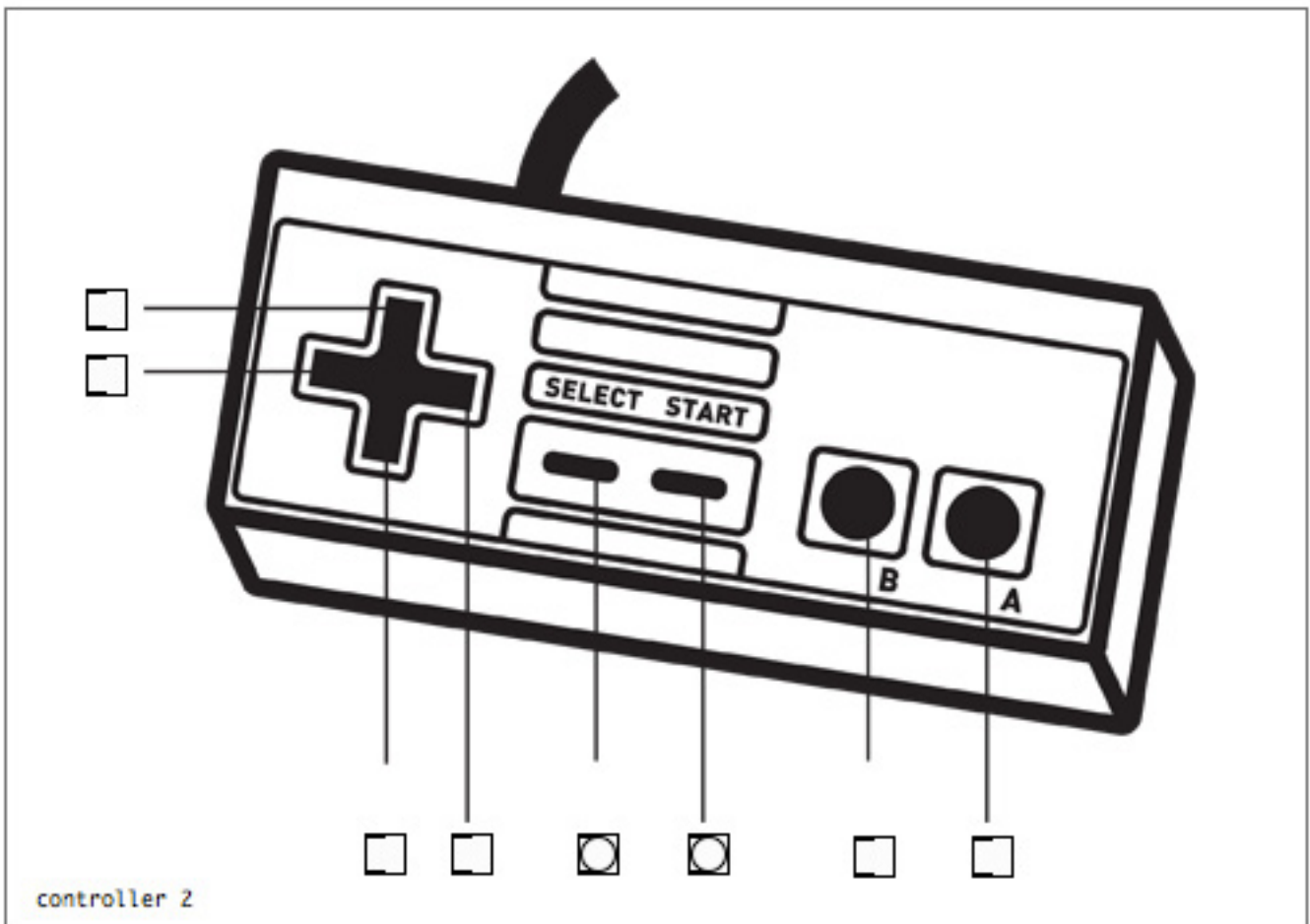


In order to establish a connection between the Arduino Board and the RESWARE you need to find which COMPORT the Arduino is using (this may vary from one computer to another), by default it is set to 2. To find your COMPORT click on the button (as below 1.), the COMPORT list will be displayed in your PD window. Simply click on the number that corresponds to the Arduino Board (as below 2.).



RESWARE Beta 1.0 Settings

Now you should be able to play. Simply press on any key of the NES controller. A preview is built in to the RESWARE Interface so you can see which key you pressed (if nothing is happening make sure you set the right COMPORT - see previous page).



Additional Information

The RES Mini and RESWARE is a project by Benjamin Gaulon based on the RES (Recycling Entertainment System) created in 2004.

Feel free to update, modify and share the Resware and Arduino source code with as many people as possible, however make sure you follow the terms and conditions stated in the license (Attribution-NonCommercial-ShareAlike 3.0 Unported). I cannot be held responsible for any damage to your NES Controllers while hacking them. The RESWARE is a custom made software and you use it at your own risk. I cannot be held responsible for any damage while using it. YOU ARE RESPONSIBLE FOR ANY DAMAGES OR COSTS INCURRED THROUGH THE USE OF THE SOFTWARE OR WHILE HACKING YOUR CONTROLLER(S).

I love to know when people use or improve my projects so feel free to contact me to share your version of the RES Mini (<http://www.recyclism.com/contact.php>) However I DO NOT OFFER ANY ADDITIONAL SUPPORT TO USE THE RESWARE OR HACK NES CONTROLLERS. I've spent some time doing this document so read it carefully and you should be fine. There are many online tutorials on how to use Arduino and Pure Data so do some research if you are stuck ;-)

Benjamin Gaulon, Dublin, 2011

RESWARE Beta 1.0 by www.recyclism.com
This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 3
Unported License.

