



Submitted by:
RONALD INGA
JEYNER CESPEDES

CET 4711

Final Project
Midi Controller

Due Date: December 19, 2012

Submitted to:

Dr. Mendoza

What is MIDI?

Musica Instrument Digital Interface it is a computer language that allows communication with other devices. This language carries instructions of music describing what notes are being played, how long each note is being play, the velocity, and notation pitch. Midi does not emitt any sound, it sends messages from a controller to another controler or software which interprets and converts them into sound as instructed. Today many music instruments are equipped with Midi capabilities. There are many Midi devices that are used for music but also for lightning systems and audio mixers.

In this project we are building a Midi controller. Our controller will have 12 buttons, 4 linear potentiometer and 4 sliding potentiometers. Each will be assign to control a software, the software we are using in this project is Traktor Pro. Traktor Pro is a DJ software that allows to be controlled by external devices. Our controller will communicate with this software using arduino Uno as the brain of our Midi controller, also Serial-2-Midi converter software to convert bits send by arduino into Midi messages.

Lets take a look how to build this controller.

Components Needed

- 1 x Arduino Uno



- 12 x Sanwa 24mm arcade buttons



- 4 x 10k linear potentiometer



- 4 x Pot knobs



- 2 x 10k linear sliding potentiometer



- 2 x Slider knobs



- 1 x 4.7 kΩ resistor



- 1 x Plywood

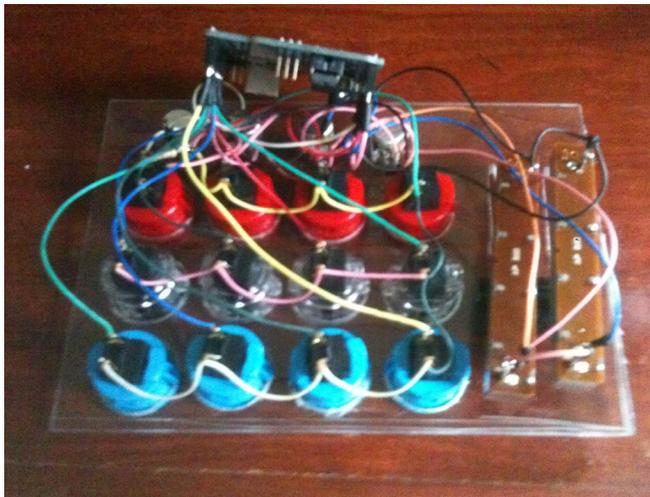
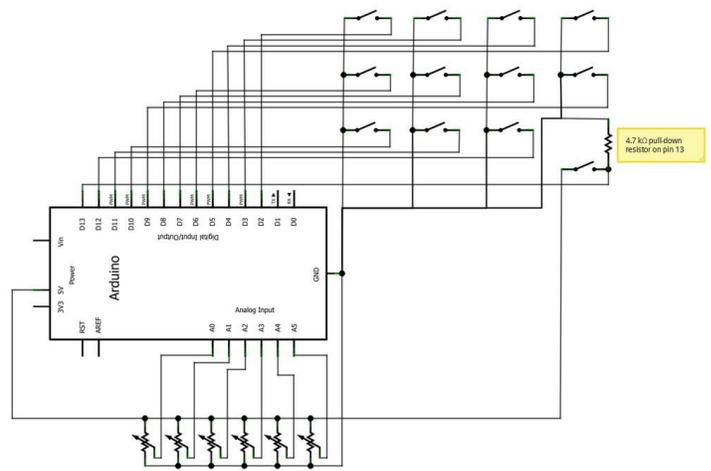
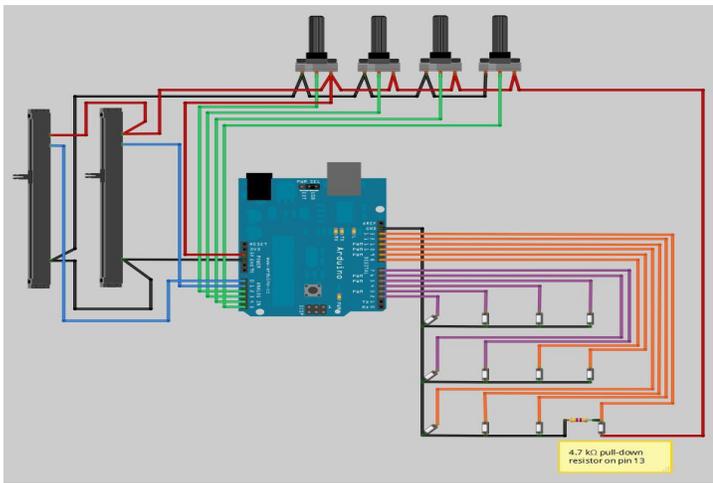


- 1 x Plastic sheet



The Price of this project is between \$90 to \$100.

Procedure



- Now we have to program our Arduino to receive signals from the buttons and send it to the software. The code can be downloaded from <http://www.instructables.com/id/Arcade-Button-MIDI-Controller/step7/Programming/>. We use the Arduino IDE to program our Arduino.

```

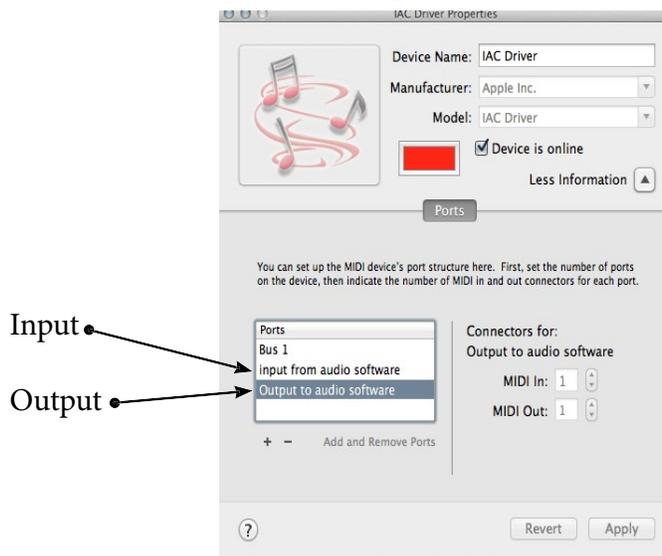
MIDI_Controller
#define DEBUG
Serial.println("Button released.");
Serial.println(pitch);
#if defined(TEENSY_PLUS_PLUS) || defined(TEENSY_2) || defined(TEENSY_PLUS_PLUS_2)
usbMIDI.sendNoteOff(pitch, 80, channel);
#else
Serial.write(channel);
Serial.write(pitch);
Serial.write(byte)80);
#endif
}
}

// Send a MIDI control change message
void controlChange(byte channel, byte control, byte value)
{
  // 0xB0 is the first of 16 control change channels. Subtract one to go from MIDI's 1-16 channel
  channel -= 0xB0 - 1;

  // Ensure we're between channels 1 and 16 for a CC message
  if (channel >= 0xB0 && channel <= 0xBF)
  {
    #define DEBUG
    Serial.println(control - MIDI_CC);
    Serial.println(" ");
    Serial.println(value);
    #if defined(TEENSY_PLUS_PLUS) || defined(TEENSY_2) || defined(TEENSY_PLUS_PLUS_2)
    usbMIDI.sendControlChange(control, value, channel);
    #else
    Serial.write(channel);
    Serial.write(control);
    Serial.write(value);
    #endif
  }
}

```

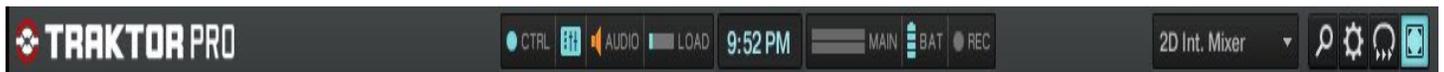
- At this point our box is done, but it is not ready yet to send messages to Traktor. To make our controller work we need to set up our audio driver. In this project we are using Mac, our audio driver is located in Utilities/AudioMidiSetup/IAC Driver. In this driver we need to add an input an output. In the port section we click the + sign to create one input port and one output port. What this port do is to input the signal from our controller and send it to the software through the output port. As we can see in the picture below the ports that we create are: input from audio software and output to audio software.



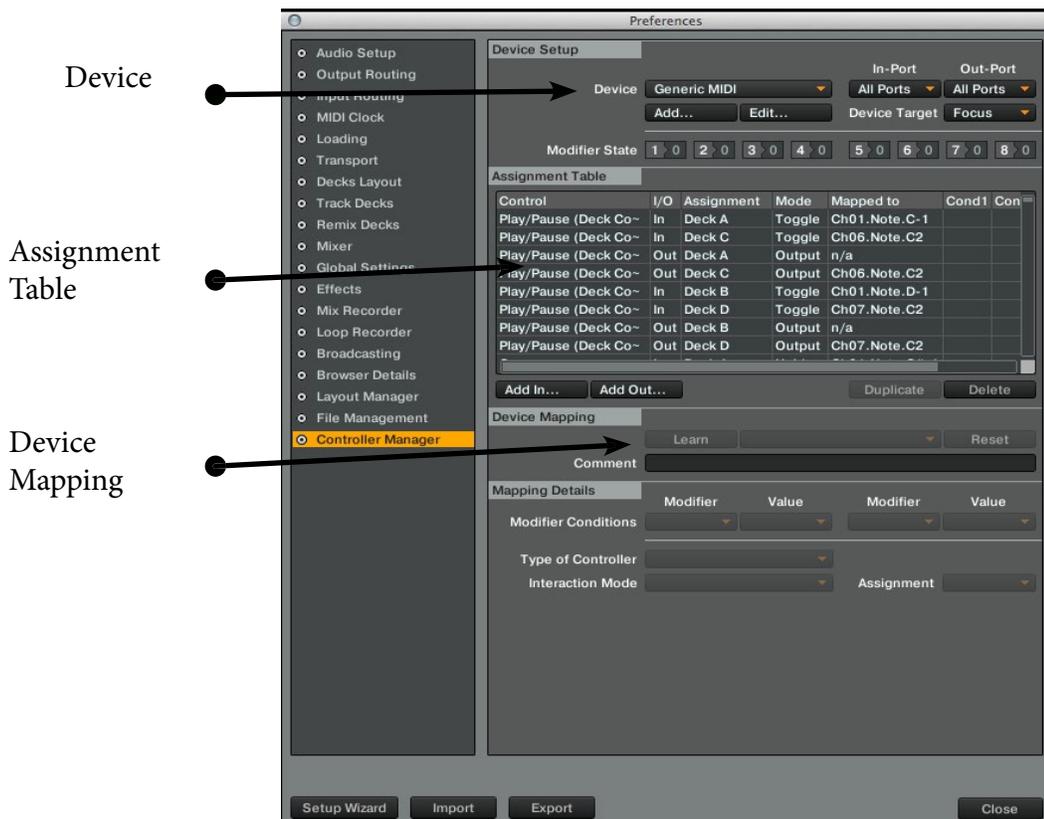
- Our controller is not yet ready to control Traktor. We are using Arduino Uno as the brain of our controller and this controller is using bits. The problem here is that Arduino is not sending midi signals even though we program it and we have our input and output ports. What the code in arduino does is to send bits when we press a button in our controller. To convert this bits in midi signals we need a software called Serial_MIDI_Converter_V2C. This software can be downloaded for free from www.spikenzielabs.com. What this code do is to convert the bits from our controller into Midi messages that are send to our input in our audio driver. We need to run this software before we run Traktor because we need to configure which port is receiving the bits and through which port Midi messenger are send. If we take a look at the picture below , we see that we need to configure the port that we are connecting our device, the sound rate, the input and output. After we are done with the configuration we leave this software running as long as we are using our controller to convert to midi every component we use in the controller



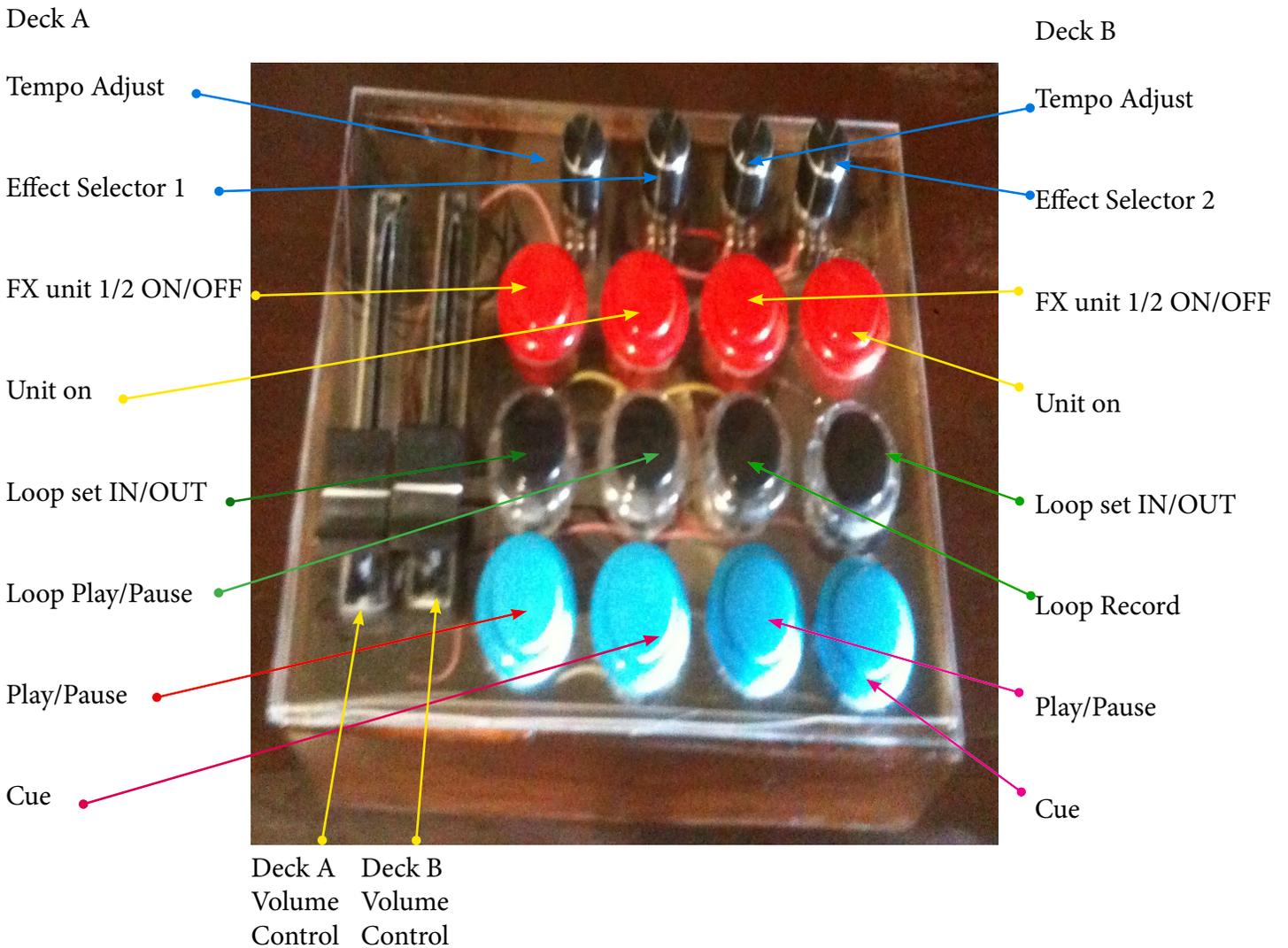
8. Now we are ready to open Traktor but our controller will not work yet. This is the last step before the party begins. At this point every button we press is converted into Midi signals and Traktor recognizes it.



We know Traktor recognizes our controller when we press any button and the CTRL light in the software turns on, as we see in the picture above a blue light in CTRL. From here we just have to map every component in our controller to Traktor. To map we need to go to preferences by clicking the  icon on the top left corner and select control management.



As we see in the picture above, this is where we map our controller to the software. In the Device section Generic Midi should be selected for our device to be recognized. On the assignment table we choose a control (ex play/pause). On the device mapping section we choose a button from our device to be assigned to the control. Our mapping is detailed in the following picture.

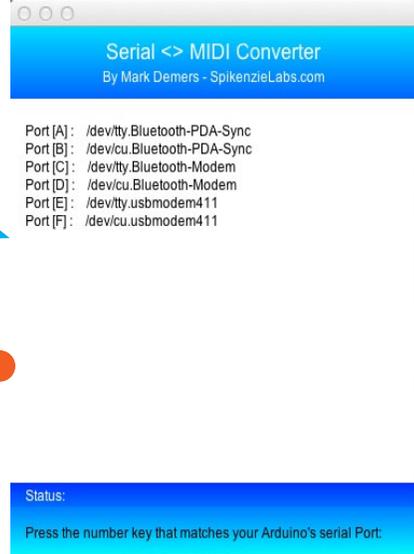


After mapping our controller we are ready to party.

How our Controller works



Controller send signals to Serial_Midi converter



Serial_Midi converts signals into Midi messages and send them to the IAC driver



IAC driver transfer Midi signals to Traktor where we map our controller

