Amadeus Board Rev 1.5
User Manual

Roberto

March 1, 2023

Contents

1 Wiring

An Amadeus board consists of two YM2149 Integrated Circuits guided by an ATmega328-P microcontroller. To communicate with the board and upload custom software, two options are proposed:

- Serial communication over USB provided by the CH340G.
- ICSP communication over the ICSP header (should only be used without the YM2149's installed).
- MIDI communication.

Once generated, the sound of the YM2149 is amplified by the LM358 IC. This also makes some channels sound more over one side than the other.

Finally, a button **MODO**, and a LED are proposed to the user to communicate in a more user-friendly manner. Putting all the modules together we have the following:

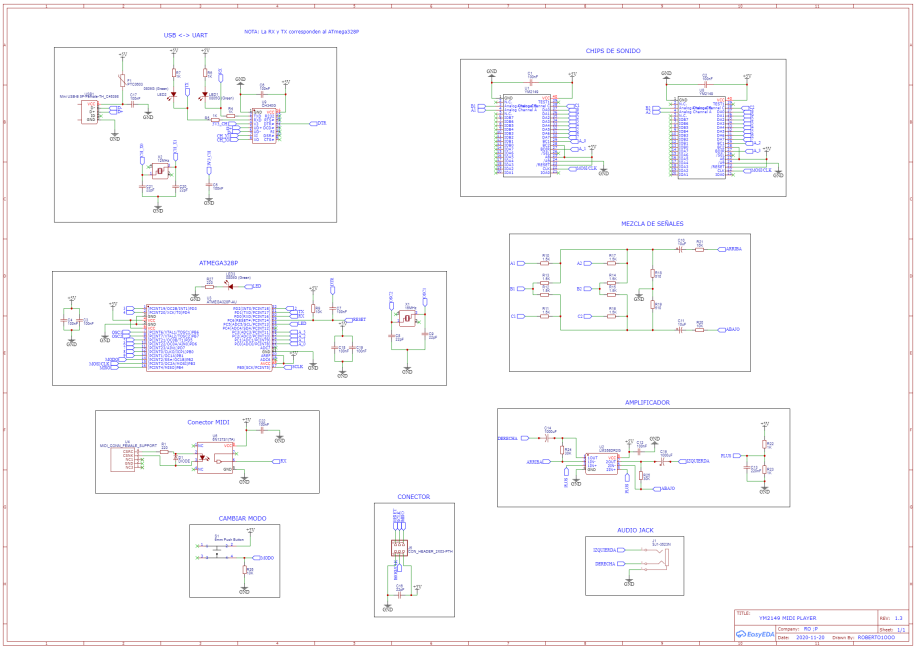


Figure 1: Schematic of the board.

We will take a closer look now into each individual part.

1.1 USB ↔ UART communication.

This is the main method of uploading code without the need for a programmer. It is done thanks to the CH340G[1]. This integrated circuit received the USB signal from the host and translates it into UART directly into the RX and TX pins of the ATmega328-P. Two LEDs have been installed in an Arduino-like manner to make is visible when the board is receiving or sending data.

The CH340G also takes care of auto-resetting the board when a program is uploaded, this can be seen as the **DTR** connection.

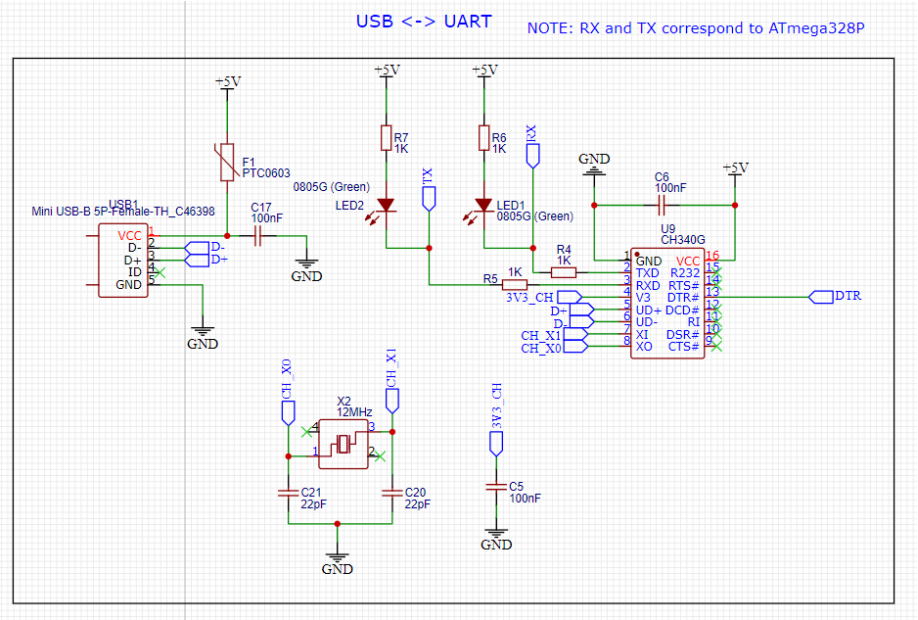


Figure 2: Schematic of the USB to UART connection.

1.2 MIDI IN

To make the board be able to receive MIDI commands, an optocoupler (6N137) inverts the signal received and makes it sit either in 5 or 0 Volts, independent of the device's voltage. The signal received is then fed to the ATmega's **RX** pin that will receive the signal as if it were UART.

IMPORTANT: Do not send data via USB to the board if you are using the MIDI connection. The USB and MIDI connection share the **RX** pin, meaning

[1]In order for your computer to be able to talk to the board, make sure to have installed the correct drivers at `https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all`.

that if the CH340 and the optocoupler send data simultaneously, the board will only receive useless bytes.

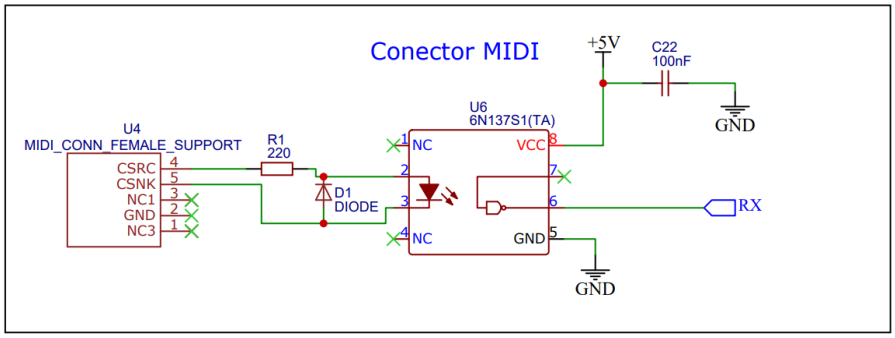
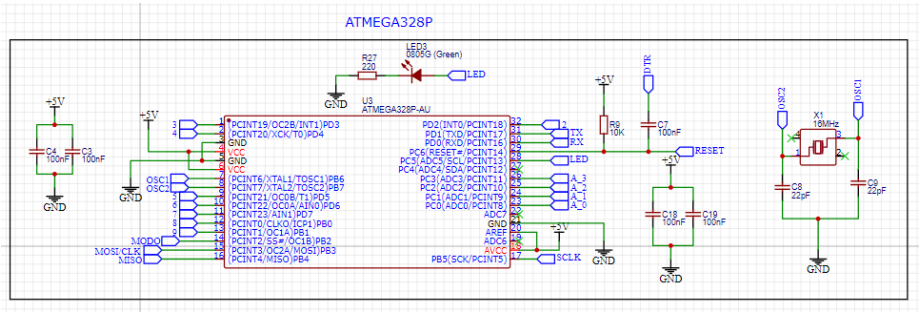


Figure 3: Schematic of the MIDI connection.

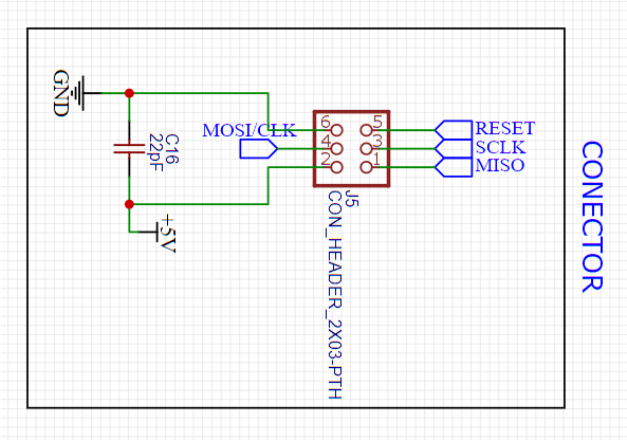
For more information on how to use it, please refer to the *How to use: MIDI connection* section.

1.3 ATmega328-P

This chip does all the heavy work of receiving the signals sent either by the USB connection or the MIDI and then sending them to the YM2149s. The specifications are the same of a regular Arduino (even the bootloader is that of an Arduino). In Fig. 4a we see the incoming **DTR** signal from the CH340G.



(a) Schematic of the ATmega328-P and all of its components.



(b) Schematic of the ICSP header.

Figure 4: .

Also in this image there are the pins **RESET**, **SCLK**, **MISO** and **MOSI** present in the ICSP connection used for the initial flashing of the chip with the bootloader. This will make it possible for the USB connection to be used. Something very important is that the **MOSI** pin is also called **CLK**, this is because it is used as the clock that drives the YM2149s: this means that when the ICSP is programming, the YM2149 is also receiving a pulse. This is why ICSP shouldn't be used when the chips are on the board.

Finally, there are the pins noted from **2** to **9** which are directly connected to the YM2149 pins, the LED, and the button.

1.4 Dual YM2149

We arrive at the chips that make the magic happen, the two YM2149s. They receive:

- The orders through the pins **A_0** and **A_1** for the first IC, and **A_2** and **A_3** for the second one.
- The values through the pins from the pins **2** to **9**.
- The clock through the pin **MOSI/CLK**.

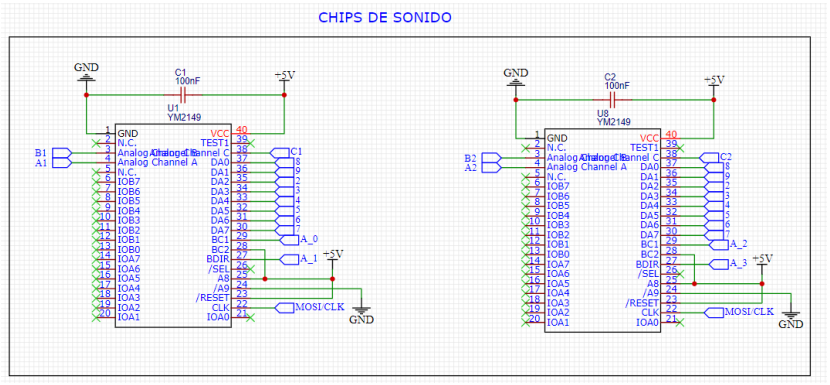


Figure 5: Schematic of the two YM2149.

Then, the sounds come through the pins **A1**, **B1** and **C1** for the first chip, and **A2**, **B2** and **C2** for the second one. These signals then have to be combined, which is done in the following way.

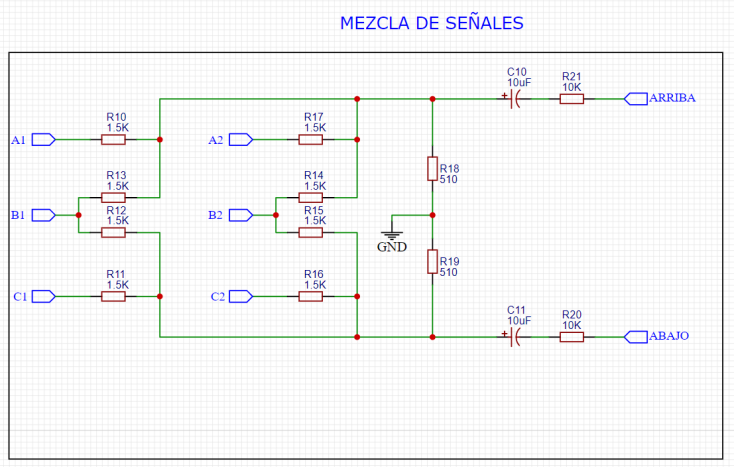


Figure 6: Schematic of the two YM2149.

1.5 Amplifier and 3.5mm jack[2]

Once the channels' outputs have been put together, the final step is amplifying the mixed signal. This is done by the LM358 Dual OpAmp, which then puts the signals to the audio jack.

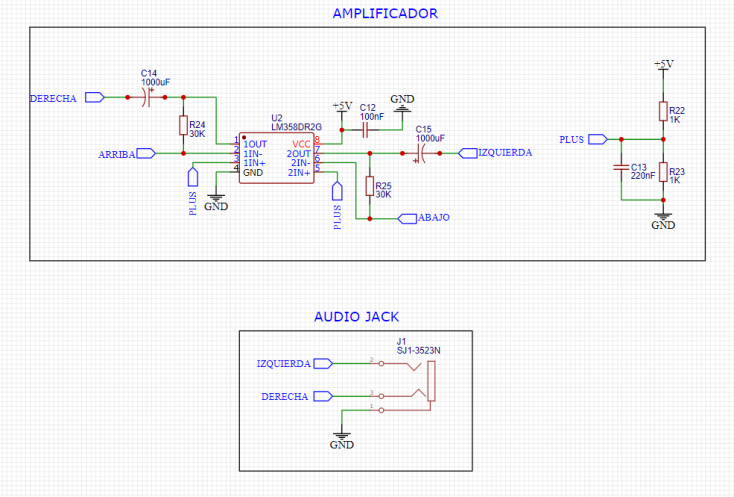


Figure 7: Schematic of the amplifier.

It can be seen that during the mixing phase at Fig. 6, the signals coming from **A1** and **A2** encounter less resistance going to **ARRIBA**, which is then amplified to the right channel (**DERECHA**). This makes channel A of both integrated circuits sound more to the right than to the left. The same applies to **C1** and **C2** which are amplified to the left channel (**ABAJO** → **IZQUIERDA**).

This, and the fact that static noise is amplified are known issues. However, the right/left channel disparities are not very noticeable, and when using a volume level of 10 or more, the noise becomes inaudible.

2 How to use

2.1 USB connection

In this configuration, the board has been programmed with the default board program[3]. The only physical connections necessary are USB to the host com-

[2] In order to reduce noise, revisions 1.5 and upward don't have an amplifier.

[3] AY3910RegWrite-library. Can be found at `https://github.com/DerSpanischGamer/ay-3-8910-midi`.

puter for data and power and headphones/headsets to the 3.5mm jack.

The format of the data expected by the board is the following. Data must be sent in packets of 3 bytes :

1. Number of the chip to write to (0 for the first, 1 for the second).
2. Register to write to (from 0 to 14).
3. Value that has to be written (from 0 to max of the register).

The board will then do the job of writing the value to the selected chip and register.

2.2 MIDI connection

To use the MIDI connection first, we need to flash the board with the MIDI program[4]. Once this has been done, it is time to plug the board.

First, connect your headset/headphones to the 3.5mm audio jack. Then, connect the MIDI cable of your device (preferably turned off). Once this has been done, plug the board into a +5V adapter to an outlet, or through a charge-only cable to your computer. It is really important that the cable doesn't send any type of data, as this could perturb the MIDI signals sent by your device. Once the board has been powered, you can turn on your MIDI device if it was turned off, and start sending notes.

Note: it is a known bug that the first one or two notes sent won't be written to. Also, the volume is set to the constant value of 10.

2.3 Uploading code

First of all, make sure to have the drivers for the CH340G installed[5].

With the drivers, we will now open the Arduino IDE in order to install the libraries necessary. To do this, you will need the two folders that are found in the /Libraries folder in the GitHub repository. You will need to compress each folder into a .zip file independently so that when you open the .zip file, you see the folder. When you have done this for each folder (Amadeus and MIDI) head to the Arduino IDE, click on **Sketch > Include Library > Add .ZIP Library...** and select one .zip file and then repeat with the other. Now you have installed both libraries.

[4] MIDI_receiver. Can be found at `https://github.com/DerSpanischGamer/ay-3-8910-midi`.

[5] More info at https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all.

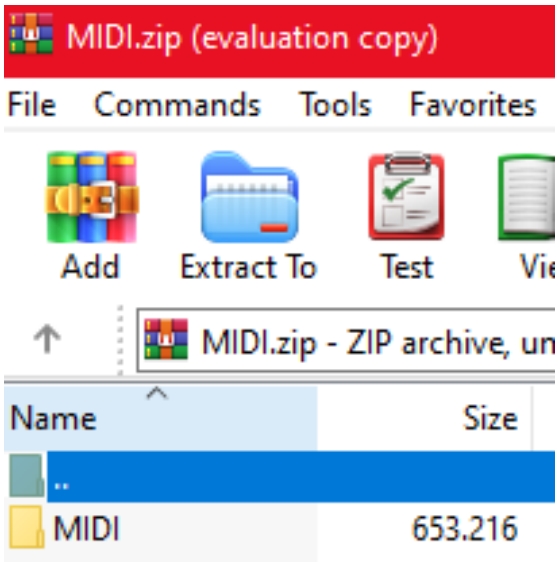


Figure 8: When you open the .zip file it should look like this, a folder inside the .zip.

Once that's done, connect your board only with the USB connection to your computer. Then, in the Arduino IDE, open the code you want to upload. Then go to **Tools** > **Ports** and select the port that the Amadeus board is connected to[6]. The board now has been selected.

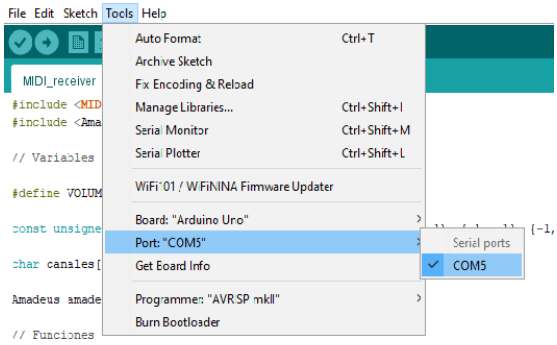


Figure 9: Port selection in the Arduino IDE.

[6]If you don't know to what port it is connected, go to **Device manager** and look for **Ports (COM LPT)**. Expand, and look for **CH340G**, the port should be between parenthesis.

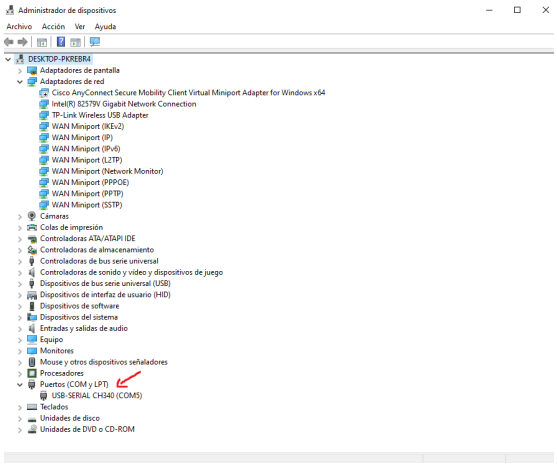


Figure 10: Finding the port in Device Manager.

To be able to correctly send the program, in **Tools** > **Board** select **Arduino Nano**, and in **Tools** > **Processor** select **ATmega328P (Old Bootloader)**. The Arduino IDE now knows how to send the program and you are all set.

Before uploading the code : Verify that there is no MIDI device connected to the board, as this will disrupt the uploading of the program. You may also choose to disconnect the audio output. One you have verified these conditions, click the **Upload** button and upload your code.

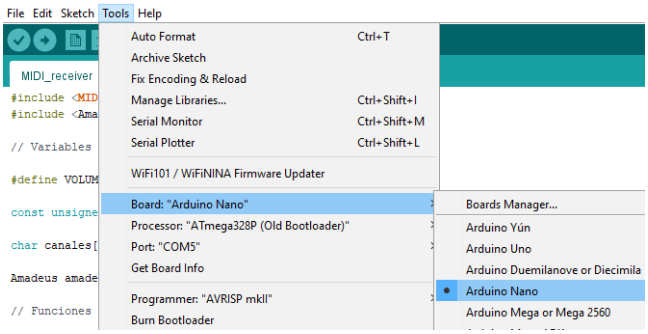


Figure 11: First we select the board type to Nano.

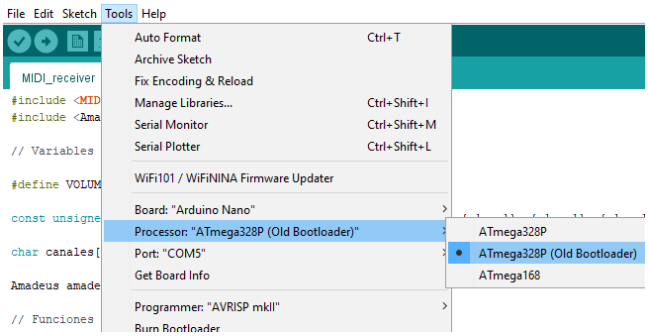


Figure 12: Then we select the processor.

Note: the code that's installed uses the Amadeus library. This library puts together the clock setup and all the functions that enable communication with the two YM2149s. This library can be found at `https://github.com/DerSpanischGamer/ay-3-8910-midi/tree/master/Libraries/Amadeus`. If you wish to upload your own MIDI code, I use a modified version of the MIDI library. This version can be found at `https://github.com/DerSpanischGamer/ay-3-8910-midi/tree/master/Libraries/MIDI`. The modified version makes the board only acts as a receiver.