Amadeus Board R 1.3
User Manual

Roberto

Rev. 1

Contents

1 Wiring

An Amadeus board consists of two YM2149 Integrated Circuits guided by an ATmega328-P micro controller. 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.

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:

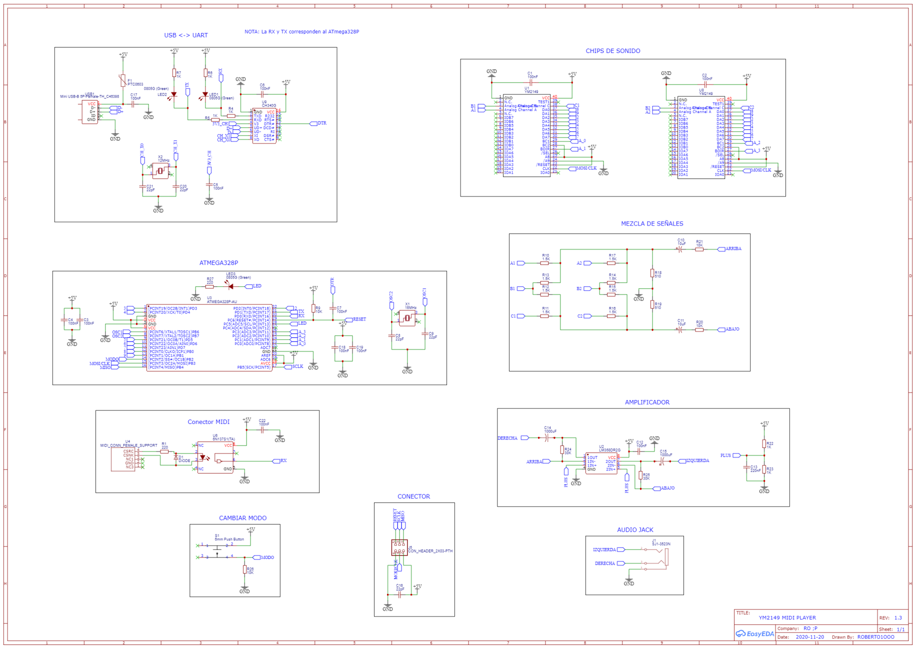


Figure 1: Schematic of the board.

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

1.1 USB ↔ UART communication.

This is the main method uploading code without the need of 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.

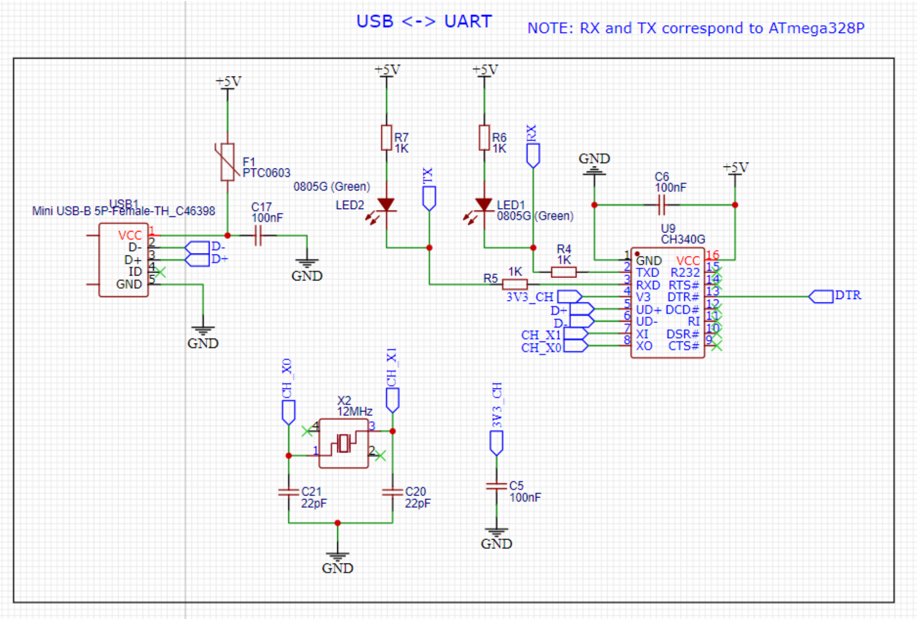


Figure 2: Schematic of the USB to UART connection.

Another function of the USB connector is powering the board. In order to protect the board from high currents, a fast blown fuse of 1.5A has been in series between the **VCC** pin of the USB and the **+5V** of the board.

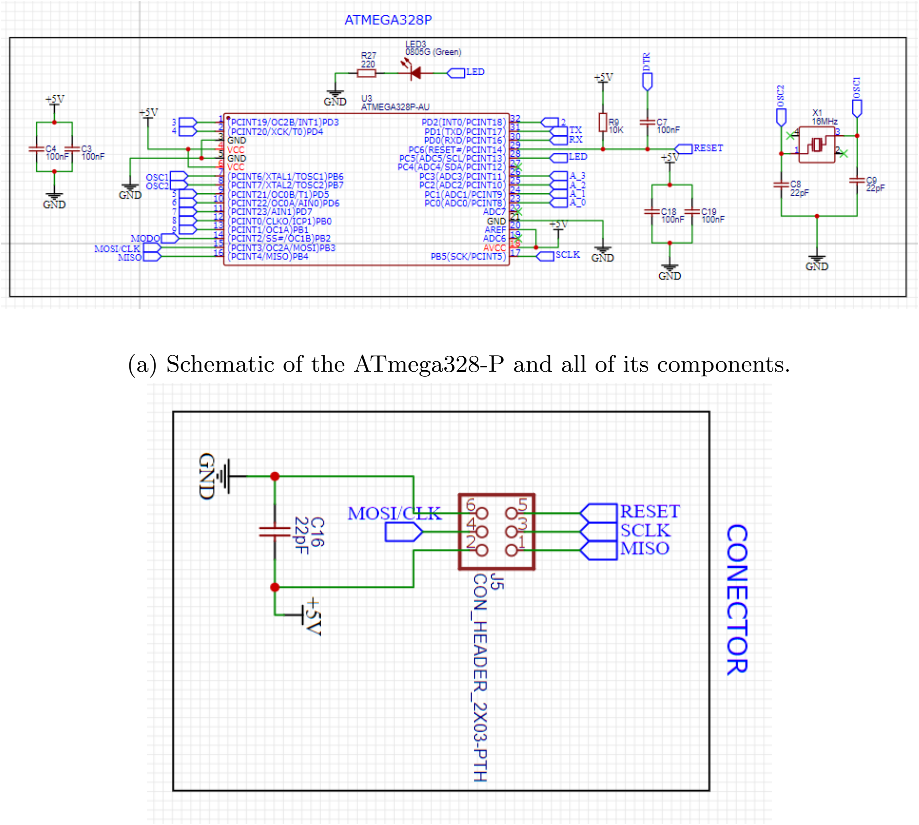
1.2 MIDI IN

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

[1] In order for your computer to be able 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`.

specifications are the same of a regular Arduino (even the bootloader is that of an Arduino). In Fig. 3a 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 3: .

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**.

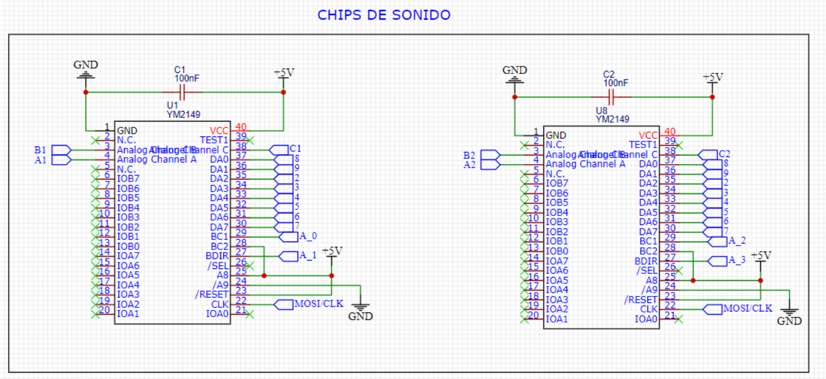


Figure 4: Schematic of the two YM2149.

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.

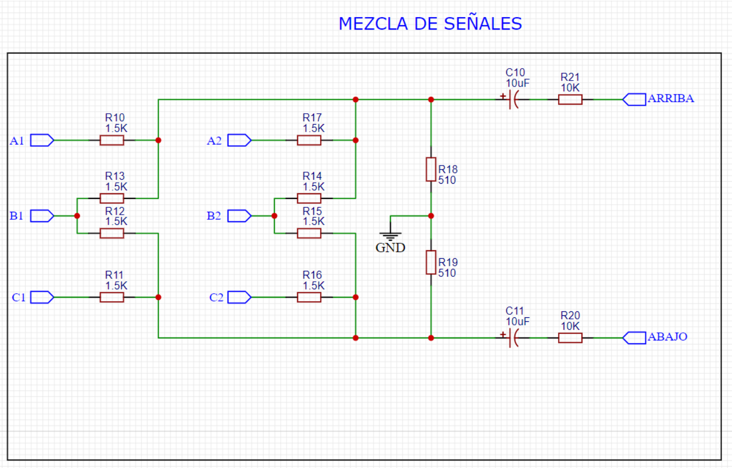


Figure 5: Schematic of the two YM2149.

1.5 Amplifier and 3.5mm jack

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 them puts the signals to the audio jack.

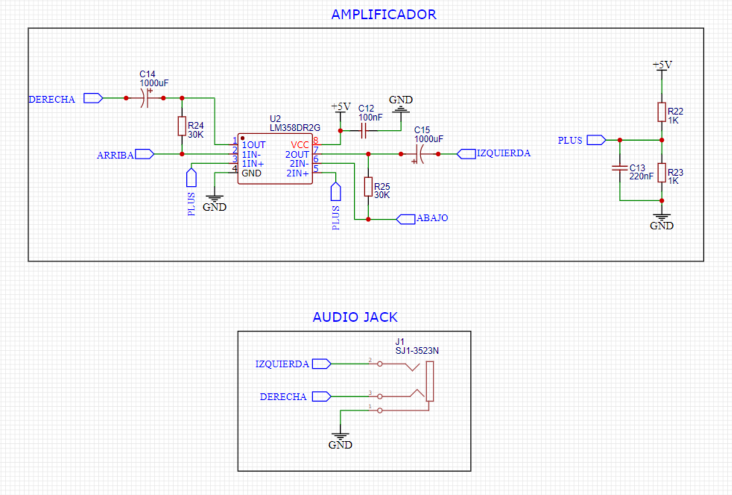


Figure 6: Schematic of the amplifier.

It can be seen that during the mixing phase at Fig. 5, 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 only connections necessary are the USB input to power and send data to the board, and the 3.5 mm jack to listen.

With the default board program [2], the board expects packets of 3 bytes containing, in this order:

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).

2.2 MIDI connection

To use the MIDI connection, we need:

2.3 Uploading code

First of all, make sure to have the drivers for the CH340G installed. Once that's done, connect only the USB to your computer. Then, in the Arduino IDE, open your code. Then go to**Tools ¿ Ports** and select the port that the Amadeus board is connected to. Now, in **Tools ¿ Board** select **Arduino Nano**, and in **Tools ¿ Processor** select **ATmega328P (Old Bootloader)**. You can now press the upload button.

Note: the code that's installed uses the Amadeus library. This library puts together the clock setup and all the functions that enable the communication with the two YM2149s. This library can be found at `https://github.com/DerSpanischGamer/ay-3-8910-midi/tree/master/Amadeus`.

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