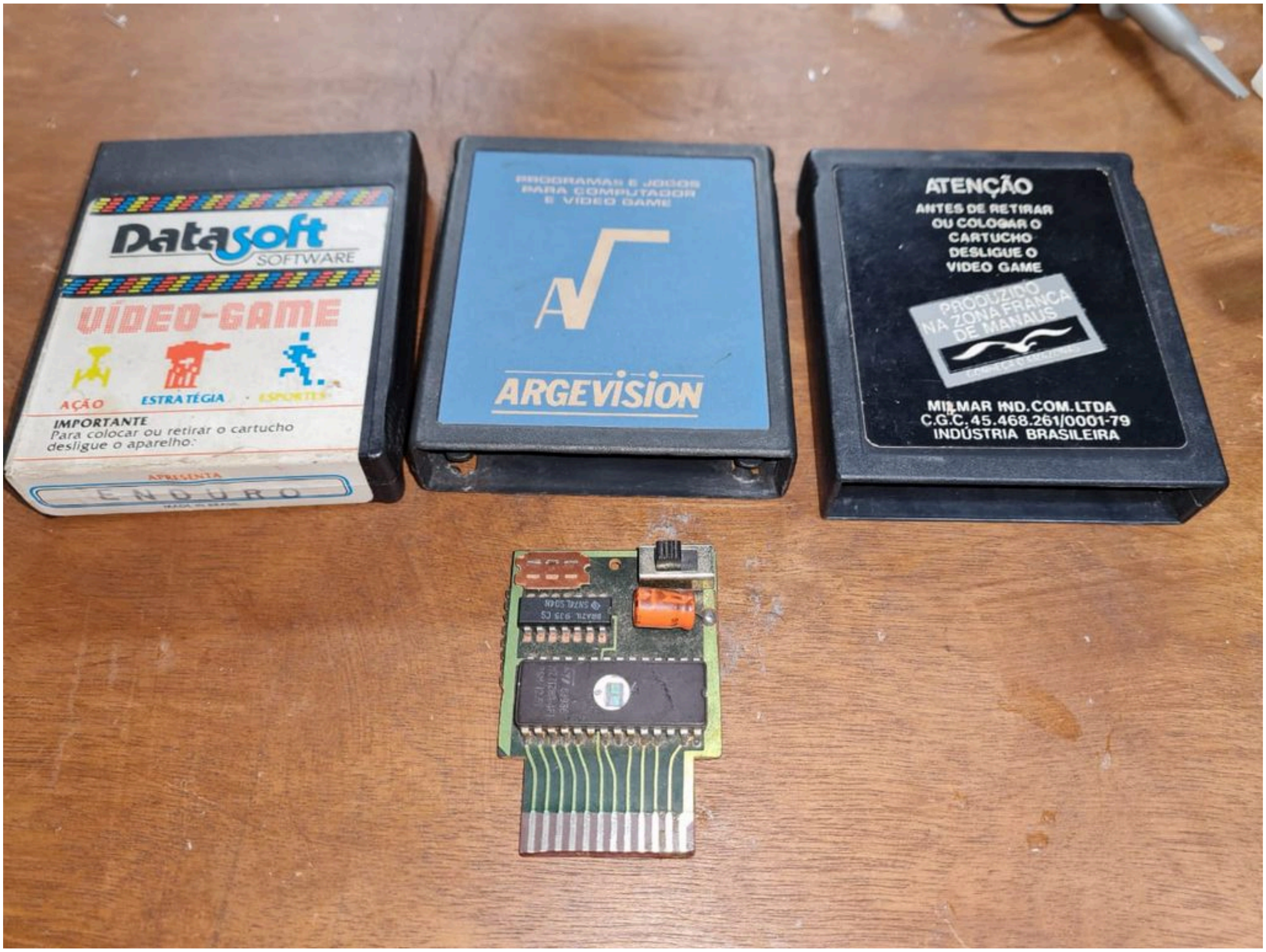
Atari 2600 ROM dumper using Arduino

During a house cleaning I found an old box containing several ancient Atari 2600 cartridges. I really wanted to play them again, so I started this weekend project: The simplest Atari 2600 ROM dumper I could assemble to play those games as fast as I could.


Datasoft SOFTWARE
VIDEO-GAME
AÇÃO
ESTRATÉGIA
ESPORTES
IMPORTANTE
Para colocar ou retirar o cartucho desligue o aparelho.
APRESENTA
ENDURO
PROGRAMAS E JOGOS PARA COMPUTADOR E VIDEO GAME
ARGEVISION
ATENÇÃO
ANTES DE RETIRAR OU COLOCAR O CARTUCHO DESLIGUE O VIDEO GAME
PRODUZIDO NA ZONA FRANCA DE MANAUS
MILMAR IND.COM.LTDA
C.G.C. 45.468.261/0001-79
INDÚSTRIA BRASILEIRA

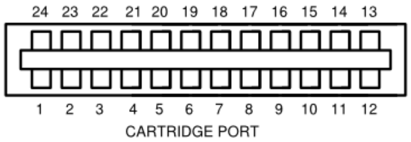
I was particularly interested in this little guy at the bottom. There's no cartridge packing, cap, no box, no information, nothing. Just the exposed circuit board. What fun memories from my childhood this little thing could be holding? However, that transparent circle on the center of the memory could be a problem: The "cartridge" could be totally empty, having lost all its stored data due to the exposed EPROM memory. That window was used to clean the memory using light, so developers could use it to test several versions of their code using the same cartridge. After programming it, the window should be covered to protect the data. But more than 30 years with the memory fully exposed to eventual light sources could have make this game lost forever.

So, I wanted to play those games, and I wasn't interested in investing too much to dump those ROMs, since they could be useless after so many years. After looking out for the Atari pinout, I figured out that it could be easy to extract the ROMs using only an Arduino Mega. No other ICs, no other components, no PCB design required.

Here are the project details

Connecting the Arduino to the Atari 2600 cartridge

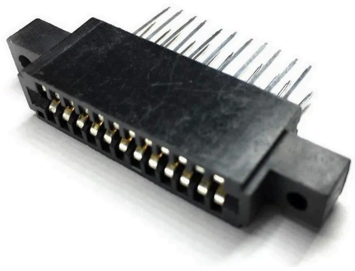
Here's the pinout for an Atari 2600 cartridge. The image represents what you see if you look the Atari console from above.



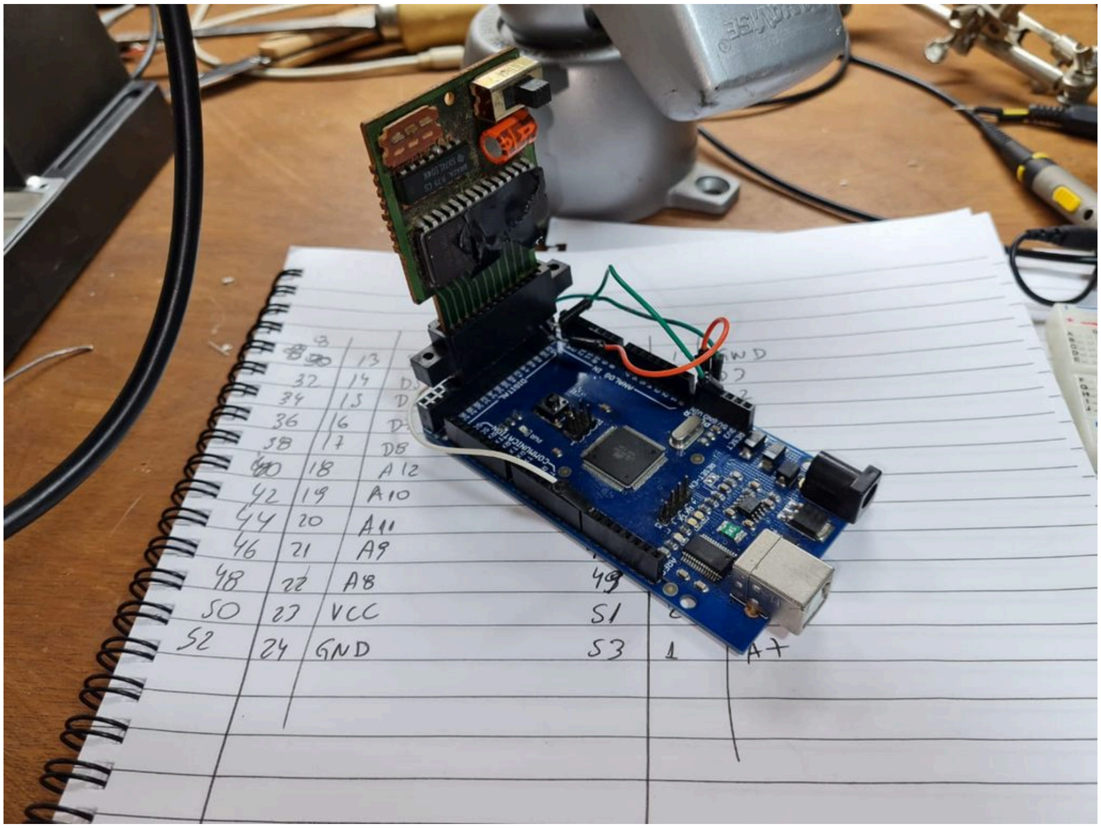
1	A7	7	A1	13	D4	19	A10
2	A6	8	A0	14	D5	20	A11
3	A5	9	D1	15	D6	21	A9
4	A4	10	D2	16	D7	22	A8
5	A3	11	D3	17	D8	23	+5V
6	A2	12	GND	18	A12	24	sGND

Atari 2600 games usually had 4kb of memory. With 13 address pins we could address 8Kb, but the MSB (pin A12) is kept high when accessing the game, so it must be connected to VCC during dump. Then we need 22 pins: 12 pins for address, 8 pins for data, and 2 pins for transmitting the dump to the computer via serial port. That's a bummer for the most common Arduino boards, but the Mega model has plenty of pins available for this.

Of course I still needed to connect the Arduino to the cartridge. I could just solder its terminals to some jumper wires, but that would destroy the board. Luckly, that socket is easy to find on any large component store. You just don't ask for an "Atari socket". Nobody sells that. Go for a "generic 24 pins edge connector" and you can get one for about 1 or 2 Dollars. Here's the one I used:



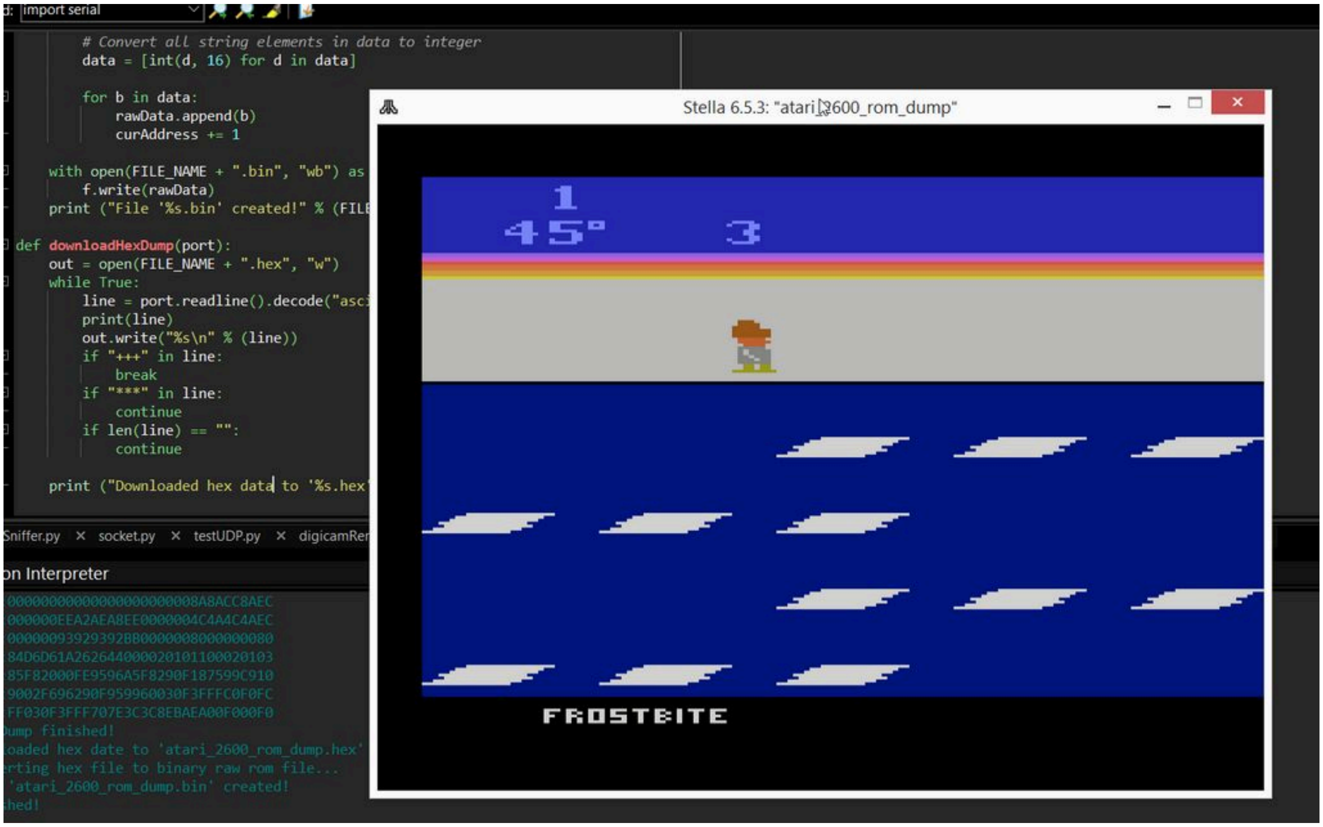
Besides saving the cartridge board, this connector also matches the Arduino Mega PCB footprint. You can almost plug it direct to that double line of pins on the end of Mega board, except for the power pins. Here's how I mounted it:



Dumping the data

After everything is connected, dumping data is a simple repetitive task: You put the desired address(From 0 to 4095) at the address pins and then get the byte stored at that memory position by reading the data pins. I made an Arduino sketch that extract this data and send it to the computer in a text format, so any transmission failure could be detected. A Python script is also provided to download this data and convert it back to raw binary data, which is the format used by most emulators.

So here's what I got from that broken cartrige. Yes, it was still working! :)



```
        # Convert all string elements in data to integer
        data = [int(d, 16) for d in data]

        for b in data:
            rawData.append(b)
            curAddress += 1

    with open(FILE_NAME + ".bin", "wb") as
        f.write(rawData)
    print ("File '%s.bin' created!" % (FILE

def downloadHexDump(port):
    out = open(FILE_NAME + ".hex", "w")
    while True:
        line = port.readline().decode("asci
        print(line)
        out.write("%s\n" % (line))
        if "+++" in line:
            break
        if "***" in line:
            continue
        if len(line) == "":
            continue

    print ("Downloaded hex data to '%s.hex'
```

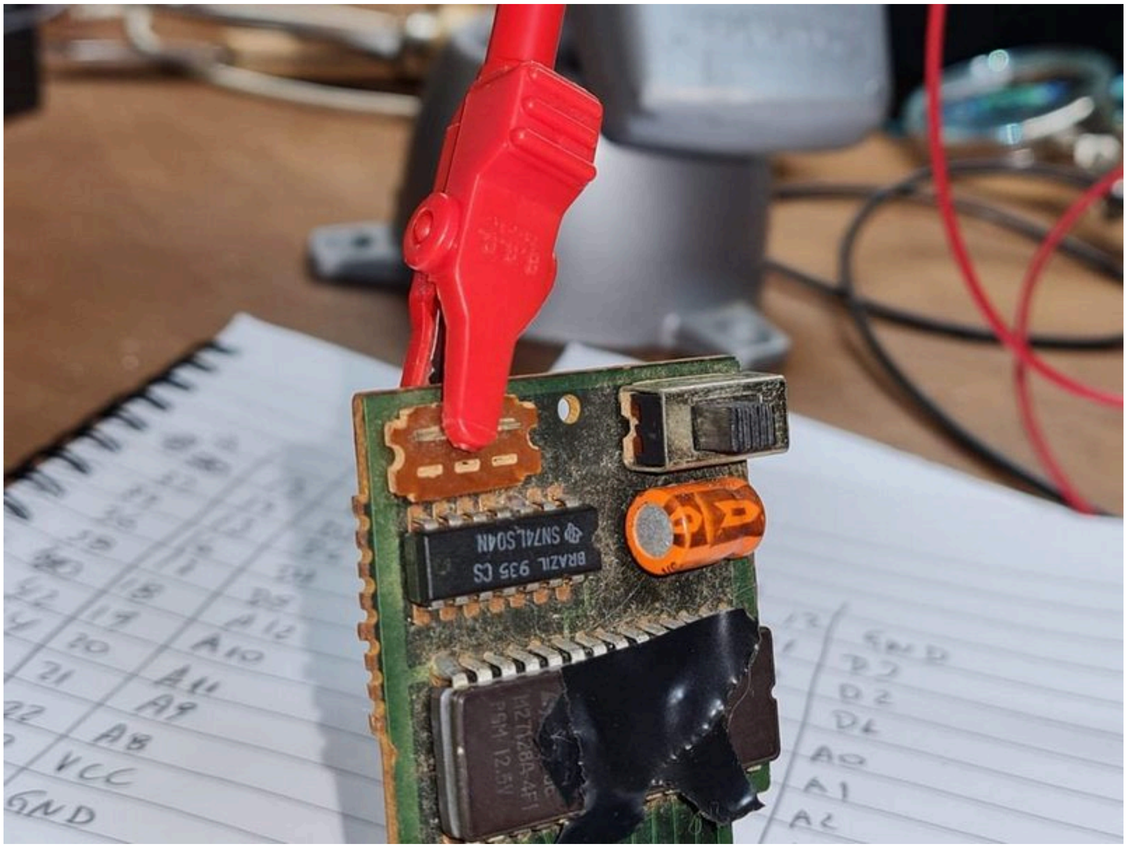
```
0000000000000000000000008A8ACC8AEC
000000EEA2AEA8EE0000004C4A4C4AEC
0000009392939288000000080000000080
84D6D61A26264400002010110002010З
85F82000FE9596A5F8290F187599C910
9002F696290F959960030F3FFFC0F0FC
FF030F3FFF707E3C3C8EBAEA00F000F0
Dump finished!
oaded hex date to 'atari_2600_rom_dump.hex'
rting hex file to binary raw rom file...
'atari_2600_rom_dump.bin' created!
hed!
```

Yeah! Playing Frostbite again!

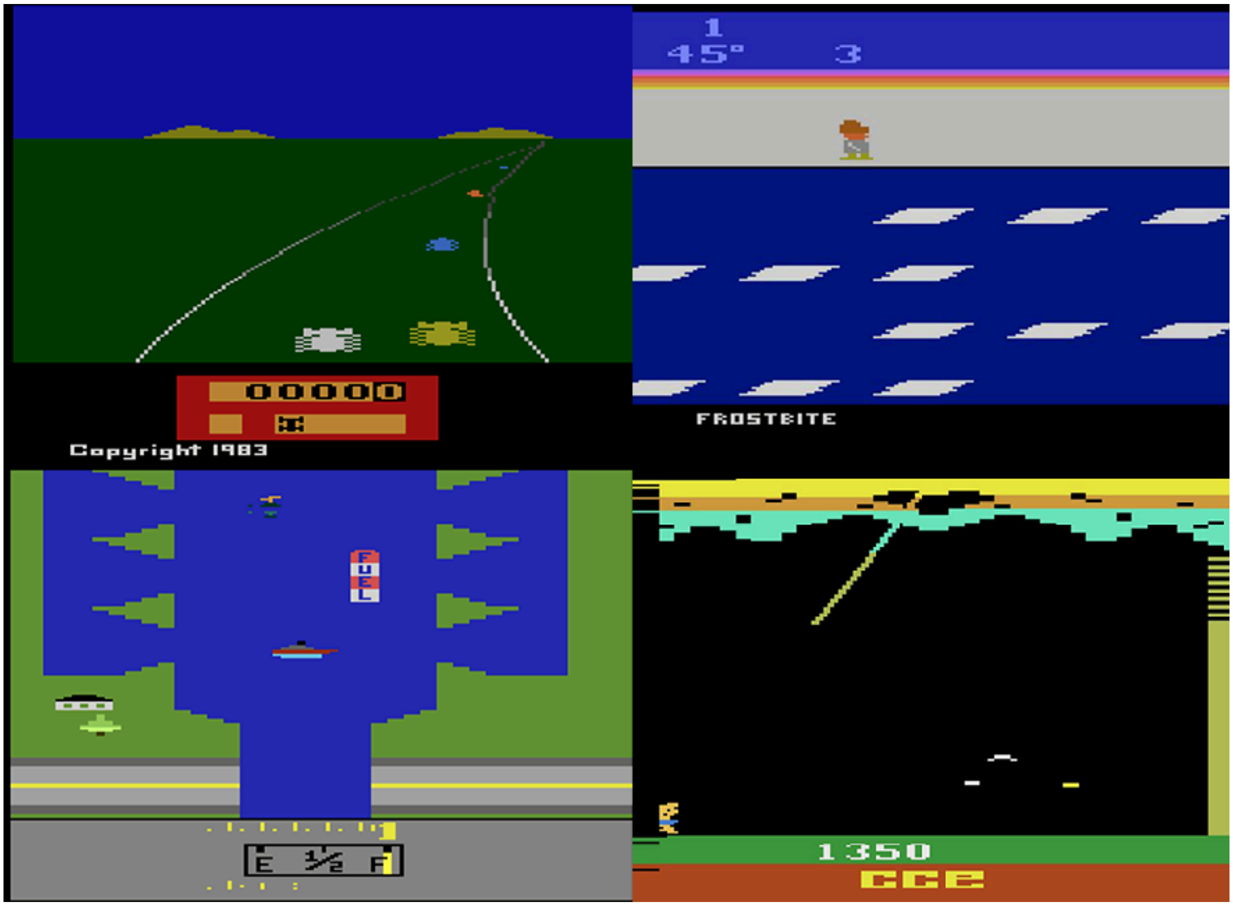
Extra information about the cartridges

None of my games were original Atari products. As you can see in the first image, those are pirate versions of the games. We Brazilians were living in the dark during the 80's, under a dictatorship government that made really hard to import electronics. You could even go to jail if someone found a foreigner computer at your home. And pirating those games were so simple and cheap (As you can see in this article) that several Brazilian companies were selling bootleg games. I think I never saw an original Atari cartridge in person.

One interesting thing is that several pirate cartridges had more than one game stored. "Manufacturers" used larger memories and put on/off switches to control the most significant bits of the address lines, in order to emulate several "memory banks". My broken cartridge has a 16 kb EPROM, so it can store up to 4 games, and you select them by combining the two switches on the top. As one of the switches is also broken, I used an alligator clip to short the pins and access the second half of the memory.



And here are the four games I dumped from that cartridge:Enduro, Frostbite, a botleg River Raid version and an mysterious game that I remember absolutely nothing about :(If you know the name, please contact me!



Project files

You can download all the files from my github page (https://github.com/aporto/atari_2600_rom_dumper). All game screens were captured using this great Atari emulator Stella (https://stella-emu.github.io/)

Search