

What do I need to start programming an Atari 2600?

Below I've collected a list of links and references that you will probably find helpful in trying to write new games for the old Atari 2600 video game. It isn't exhaustive, but it is meant to help you find the information that you need quickly and with as little frustration as possible.

A Cross Development System

If you've got any Windows, Linux or MacOS system, you can probably use it to cross develop games for the Atari 2600. Pick whichever one you feel comfortable with. You'll need a basic text editor to write your programs: I like vim <http://vim.org>.

A Good Emulator

I recommend Stella, available from <http://stella.sourceforge.net>. It's a terrific emulator, and even includes a visual debugger that allows you single step through your code and examine the various hardware registers to see what's going on. Without a good emulator, programming this machine would be very difficult. Stella runs on Windows, MacOS and Linux, so whatever your chosen platform would be fine.

An Assembler

All Atari 2600 programs are written in assembly code: there simply isn't enough resources in the machine use higher level languages like BASIC, C or Pascal. You can pick whatever assembler you like, (lots of people recommend DASM) but I'll recommend P65 from <http://hkn.eecs.berkeley.edu/~mcmartin/P65/> as one that's simple, convenient and written in perl, so its very portable.

Documentation

You'll definitely want to get a copy of Steve Wright's Stella Programmer's Guide, available from <http://www.atarihq.com/danb/files/stella.pdf>. It's got a lot of detail on how the Atari hardware actually works, and you'll be lost without it. But even with this fine document, you'll probably need more help and example code. Try checking out the forums on the Atari Age website at <http://atariage.com/forums> for some help and example code. I couldn't have completed my project without the information they gave me.

Real Atari Hardware

With just the bits you find above, you can actually develop games for the Atari 2600 and run them on your computer in emulation. If you want to make a real cartridge to run on a real Atari 2600, you'll need a few more bits such as an EPROM programmer (I got mine from <http://mcumall.com> for about \$50), some actual EEPROMS (from <http://digikey.com>), some cartridge PC boards (from <http://atariage.com>) and a donor cartridge to provide the case.

The Enigma 2600 Project by Mark VandeWettering

<http://brainwagon.org/the-enigma-2600>



This project is probably one of the quirkiest and most useless that I've ever done: I implemented a simulator the German three rotor cipher machine commonly called Enigma as a cartridge for the Atari 2600 video game. This project scratched several odd intellectual itches of mine, including the desire to write a significant program for an old retro machine, to actually purchase and use an EPROM programmer, and to learn some more about the German Enigma machine.

I first became interested in the Enigma machine years ago, but my interest crested back in 2000 when author Simon Singh released *The Code Book*, a book on the history of codes and cryptography. As part of the book, it included a set of 10 challenge ciphers, and promised a \$10,000 reward to the first person to crack all 10. I managed to crack 7 out of the ten, working mostly by myself using a variety of C programs that I wrote. Perhaps the most satisfying code that I broke was Stage 8, which was encrypted by a German Enigma machine.

It turns out that hardest part of actually cracking the code was in trying to make a 100% accurate simulator for the Enigma machine. The inner workings have a number of odd curiosities that are poorly described in most of the references that I found. It took the better part of three weeks (and some posts on the Internet) to reveal the problem with my simulator, but once they were ironed out, I implemented a ciphertext only analyzer as proposed by Jim Gillogly, and managed to crack the code in a single evening of work, and a single three minute run on my 120Mhz P5. When the 100% correct German language decrypt popped out of my program, I must admit it was one of the most satisfying intellectual puzzles I've ever broken.

Until, of course, I started programming an equally puzzling machine, the Atari 2600.

This video game would hardly even qualify as a computer these days. It had a 1Mhz 6502 processor, 128 bytes of RAM and 4K of ROM in its standard configuration. It has no framebuffer: to draw graphics you have to generate the output video signal on the fly, by telling it directly how to change the color as each scanline is traced out in real time. Programming the Atari 2600 is kind of like filling in a crossword puzzle, you figure out what work you have to do, and when it needs to be completed, and then carefully count instruction cycles to see if everything can get done at the right time.

That being said, with the powerful machines to run emulators and assemblers, the cost of making mistakes is pretty low. I spent about 15 hours from beginning to end, designing and porting my Enigma simulator to 6502 assembly code, and getting it to run. I then went the extra mile and burned my very own cartridge (with carefully designed graphics to match the original Atari's look). It was a fun project, and reminded me of the programming I did in my youth. I hope you enjoyed this writeup, and see the backside for where to find more information about Atari programming.

Mark VandeWettering <brainwagon@gmail.com>