APPLE II TWITTER DISPLAY

Document version 0.1, 13-October-2010, Chris Yerga (yergacheffe@atomsandelectrons.com)

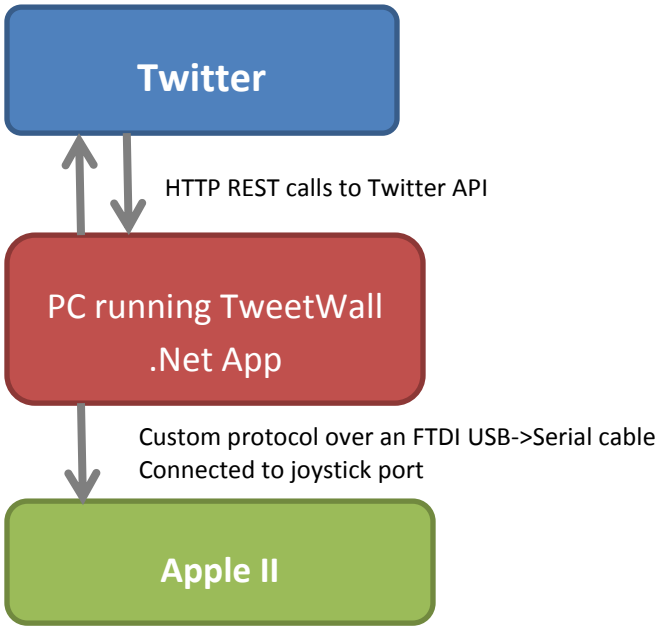
INTRODUCTION

This document describes how to duplicate the Apple 2 Twitter display I showed at Maker Faire Bay Area 2010 and documented on my blog at http://www.atomsandelectrons.com/blog/post/Apple-t.aspx. The project was hacked together quickly to get it functional, but there has been interest from others who want to duplicate it. The intent of this release is to provide the software and documentation necessary for others to replicate this and carry it forward.

Note that the project doesn't currently have pre-built binaries available. Getting things going requires some testing and code modification, so you will need a copy of Microsoft Visual Studio to build the code provided. You can get an evaluation copy for free if you don't have one already. The code is written in C# and requires the .Net 3.5 runtime.

DESIGN OVERVIEW

The high-level design is shown in the following diagram:



The design intentionally has very minimal code running on the Apple II, to keep things simple. The project is called the Apple II Twitter Display because that's all the Apple II is doing – being a display. The actual Twitter client is running on a PC that queries the Twitter API via HTTP. This code retrieves the text of the tweet and the avatar

bitmap for the user, it then converts the bitmap to Apple II graphics format (both LORES and HIRES) and pushes the converted image to the Apple II.

There are 3 major components to the project that you will need: The interface cable, the Apple II software and the PC software.

INTERFACE CABLE

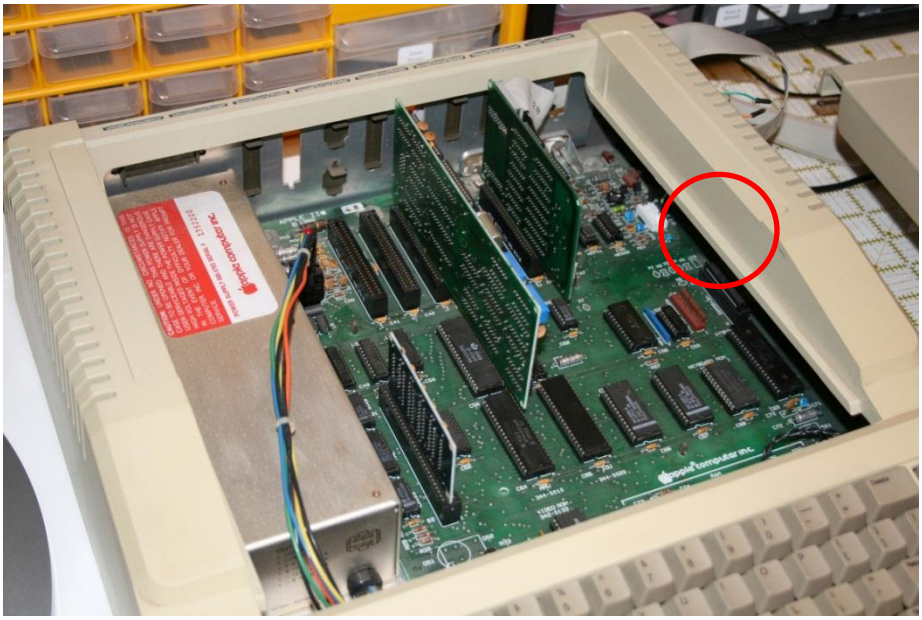
The one piece of specialized equipment you'll need, aside from a working Apple II, is a USB FTDI TTL-232 cable. This is a USB cable that plugs into a PC and allows the PC to control signals sent to the Apple II's button inputs. These cables are available from a variety of electronics hobbyist sources for around US$20

- Adafruit: http://www.adafruit.com/index.php?main_page=product_info&cPath=18&products_id=70
- Sparkfun: http://www.sparkfun.com/commerce/product_info.php?products_id=9718
- RobotShop: http://www.robotshop.com/ftdi-usb-to-ttl-serial-cable-5v.html

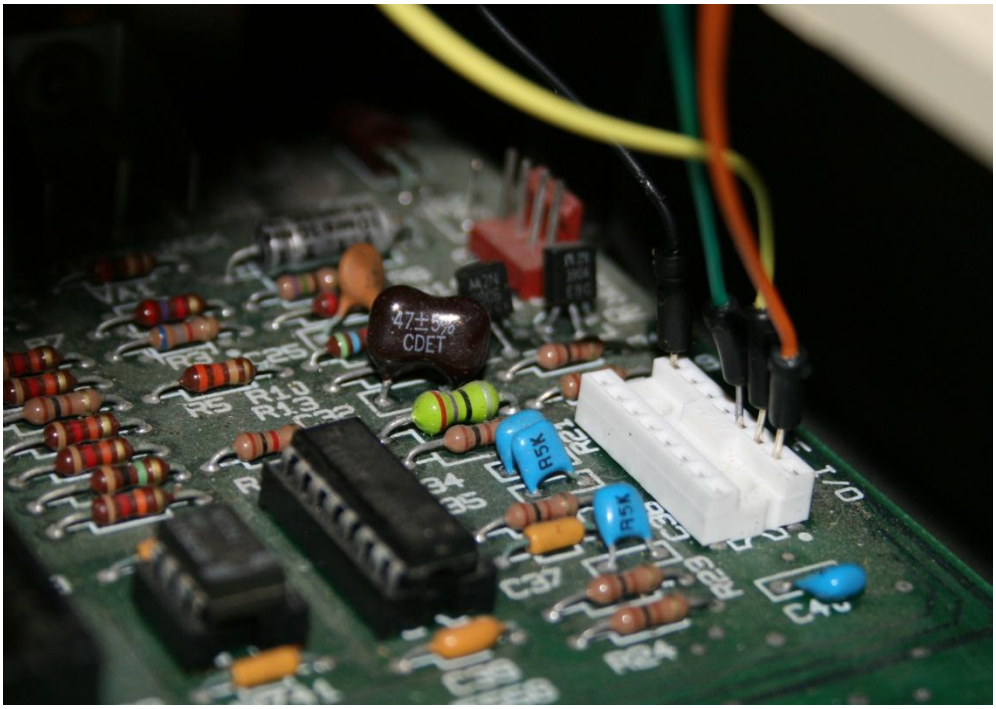
You will need to wire this cable up to the joystick input port. The Apple II has a 16-pin DIP socket on the motherboard that exposes these signals. I used wire jumpers of the sort used for breadboards to connect between the FTDI cable and the joystick socket in the Apple II. The connections we need to make are as follows:

Apple II Game Port	FTDI Cable	Purpose
GND (Pin 8)	Black Wire (GND)	Ground
SW0 (Pin 2)	Orange Wire (TX)	Data from PC->Apple II
SW1 (Pin 3)	Yellow Wire (RX)	SPI Clock
SW2 (Pin 4)	Green Wire (RTS)	Attn/Framing

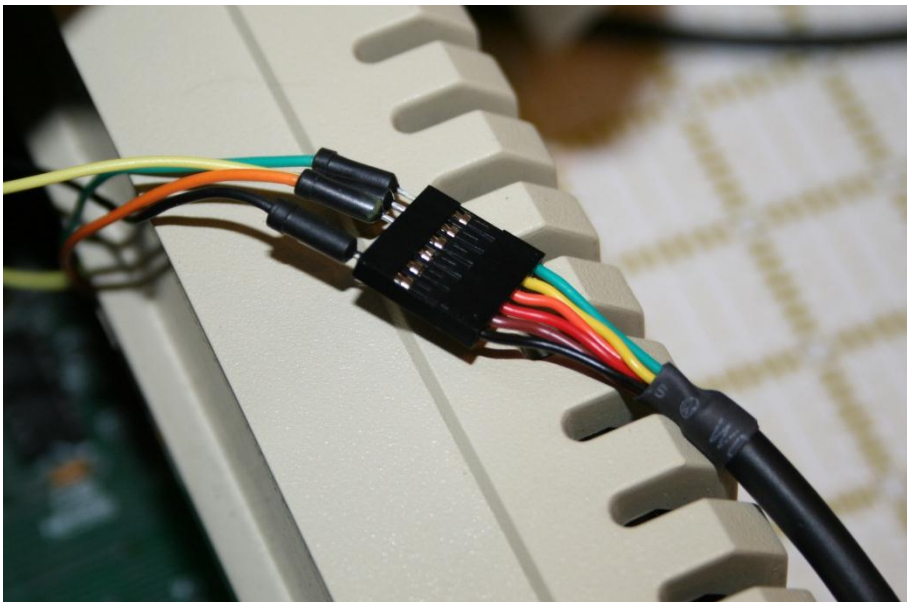
The location of the game port connector on the Apple II motherboard is shown here:



A close-up detail showing the wires connected:



And the other end of the colored jumper wires plugged into the FTDI cable:



Once you have an interface cable constructed the next piece is to test it with the PC software and Apple II software.

APPLE II SOFTWARE

First, you'll need an Apple II of course! Once you have this, you need to get the Apple II portion of the software transferred to the Apple II and running. For me this was the hardest part of bootstrapping the project. I will describe here what I did, but hopefully someone can make this simpler by providing disk images or something of that sort.

The complete code that runs on the Apple II is about 600 bytes in length. So while it's not large, it's too big to enter by hand the first time. To address this issue I wrote a small 55 byte bootloader that you can enter by hand fairly easily and then use that to receive the full program as transferred from the PC software. Note that I didn't have a functional Serial Card in my Apple II. If you have one of these, the task *should* be much simpler, but since I didn't do this I can't give concrete instructions on how to do it.

On the Apple II follow these steps to get the bootloader entered and transfer the full software:

1. If you're at a BASIC prompt (> or]) enter **`CALL -151`** to enter the monitor.
2. You should be looking at the monitor prompt "*".
3. Enter the lines from bootstrap.txt exactly as they are displayed. This data is available in the file apple2/bootloader.txt as well:

```
300:AD 63 C0 10 FB A9 00 8D
:16 03 20 20 03 8D 17 03
:20 20 03 A0 00 99 34 12
:EE 16 03 D0 F3 4C 00 03
:A9 00 A2 08 AC 62 C0 10
:FB 0A AC 61 C0 10 02 09
:01 AC 62 C0 30 FB CA D0
:EB 60
```

4. Once you've entered the code, execute it by typing **`300G`** at the monitor prompt.
5. On the .Net program on your PC uncomment the line near the top of Form1.cs that reads "apple2.LoadCode();" and execute that line of code. This should run for a short while and then return.
6. Validate that you have the code loaded at $6000 by entering 6000L at the monitor prompt on the Apple II. The disassembled code should look like the code in TwitterII.m65. You should see

   JSR $XXXX
   JMP $XXXX
   LDA $XXXX

   etc. and not a bunch of question marks.

7. You'll want to save this to a cassette or floppy. Save from addresses $6000..$63FF which is actually more than needed but will work if the code grows to 1K in the future.
8. Finally run the code by typing 6000G at the monitor prompt. At this point you can run the PC twitter client code and should start seeing tweets appear.

PC SOFTWARE

Prerequisites

To build the Windows software you will need Microsoft Visual Studio. I used Visual Studio 2008, but the more recent Visual Studio 2010 should work fine as well. The free "Express" version of Visual Studio 2010 should work, but I haven't tried it: http://www.microsoft.com/express/windows

One thing that I'd love to see is for someone to port the software to something more portable like Python so it could be run from a wider variety of platforms. I did this in .Net simply because it was the most expedient for me, and I'm not sure if it's the best choice for most other folks.

Building The Project and Testing things out

Open the solution file windows/TweetWall.sln in Visual Studio and you should get the full project loaded and can build it from the Build menu.

To test that things are working, I put a breakpoint at the top of the Idle_Apple2t() function in Fom1.cs and then press F5 to start debugging. Step into the code that loads Twitter items from TwitterProvider and step through the lines that create the "req" request and get the data into "data". Once these have executed, inspect the "data" variable to see what response you got from Twitter. If there is an error in the string, then something went wrong. If there appears to be valid XML with items then step through the next few lines until the variable "items" has been populated with the parsed items.

Inspect items and you should see an array of Twitter statuses. At this point you'll know things worked correctly fetching data from Twitter.

The next hurdle is communicating the tweets to the Apple II. Set a breakpoint on the line in Form1.cs that says "apple2.DisplayTweet(item)". If the code reaches this point then it believes it has a valid tweet ready to display. You can step into this function and step through the code that converts the image to Apple II format, etc. I would just let the program run at this point and after about 5 seconds you should see tweets in LORES on the Apple II.

Troubleshooting

If nothing appears on the Apple II, then chances are one of the following is causing the problem:

1. The cable isn't hooked up correctly. Verify the wires are going to the right spots on the Apple II game port connector and that the cable is plugged into the PC.
2. The FTDI cable is not being recognized by the PC software. Place a breakpoint in the Apple2() constructor in the file Apple2.cs and step through the code that attempts to open up the FTDI cable driver. Check the Err result codes at each step and they should be zero. If not, then try inserting the cable into a different USB port or downloading a driver from http://www.ftdichip.com/FTDrivers.htm for your OS. Karma points will be awarded to someone that adds some code to display an error in these conditions. Right now it fails silently. "It works on my machine"
3. The receiving software on the Apple II has not been loaded and running. The code needs to be loaded at $6000 and run via the 6000G command so it's listening to the serial cable commands.

If everything is good you should see something like the following:


@apple2t Uhh.. kinda running out of time here. Would it be too much trouble for you to JUST WORK FOR @#%^ SAKE!
apple2t twitter
Apple //t
Twitter Display
Twitter data fetched on a PC. Avatars converted to Apple II low-res and hi-res graphics formats and then pushed to Apple II frame buffer over a bit-bang protocol using the joystick button inputs
Mention @apple2t on Twitter to see your tweet and your avatar on the screen.