Repairing Auxiliary Jack Log

Tyler Berezowsky

December 26, 2014

Day 1

The 3.5mm auxiliary jack on my father's Jeep Compass has recently failed. I removed the radio hoping that the jack would be a standard package, and that I could easily locate a replacement from Digi-Key. Unfortunately, despite the millions of jacks Digi-Key offers, none of the footprints matched. I ended up attempting to repair the jack with hotglue and hastily re-assembling the radio to discover the jack was as bad, if not worst then it was before. So I sat on it.

While Internet browsing I stumbled upon the XS3868 Bluetooth module which is based upon the OVC3850 chip. The module offers full integration as a transmitter, receiver, audio amplifier, battery charger, and a seemingly comprehensive set of AT-command support. I decided I would repair the broken auxiliary port with a panel mount jack, and integrate this module.

The schematic for the module is basically non-existent. The only source of semi-viable information was available on the EEVBLOG form[1]. xpervis appears to have developed a pin-out for the module via the OVC3850 datasheet which he also links to a post.

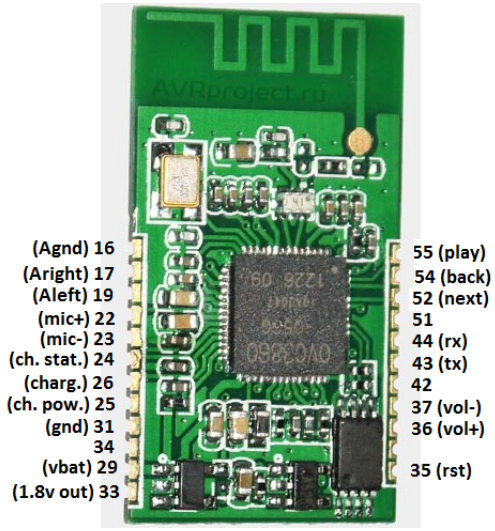

Figure 1: Pin-out for XS3868 module developed by xpertvis.

Using that pin-out and a suggested wiring diagram, I applied 3.3V to the module, but with the RESET pin shorted to the supposed 1.8V regulated output. I then attempted to make Bluetooth connection with my cellphone, but the device could not be discovered. I then applied 5V which could have potentially blown the module, but I expected some margin on the input.

[1] http://www.eevblog.com/forum/beginners/bluetooth-audio-someone/

This left the module still unresponsive. I then measured the voltages at the input and V_{reg}. The input voltage was certainly 3.3V but V_{reg} was 0V. I disconnected the RESET pin and the measurement was the same.

Baffled and potentially worried I had blown the module I started measuring current draw and resistance between pins looking for shorts. The current draw from my analog meter spiked out the 250mA range which would indicate a short or blown component. The resistance readings between pins near the power inputs were also different than an untouched module. One pair was even fluctuating in resistance. I am unsure of what would cause such a meter movement.

Tomorrow using the OVC3850 datasheet I will develop my own pin-out for the module and attempt to power it up. Today I concluded by stumbling through KiCad to develop the schematic symbols for the module and a MSP430G2253.

Day 2

Instead of reworking the previously soldered component, I attempted to follow the schematic vigilantly provided by *xpervis*. I used a thinner enamel coated wire to make the connects to the XS3868 module. All of the connections were made listed in figure 2 expect for the push buttons. In additions, the UART was wired to a Bus Pirate to attempt AT commands. An adjustable regulater (LM317) was used to provided the required input voltage. On second thought, I could have just used two diodes to drop the voltage to 3.6V which according to *xpervis* is within spec for operation via the internal buck converter.

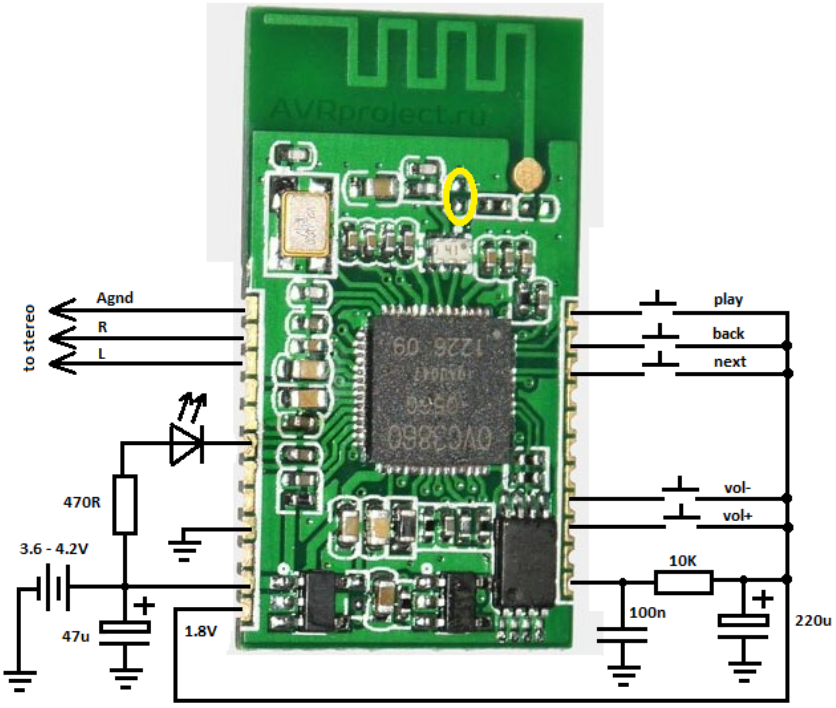

Figure 2: Suggested basic wiring for module via xpertvis.

The module wired to the breadboard is shown below. The circuit did not function on the first power cycle. After various tweaking, none of which I believe would fix anything unless there was a loose connection through the breadboard contacts, everything magically worked! I was able to connect to the module via A2DP and output to an external speaker.

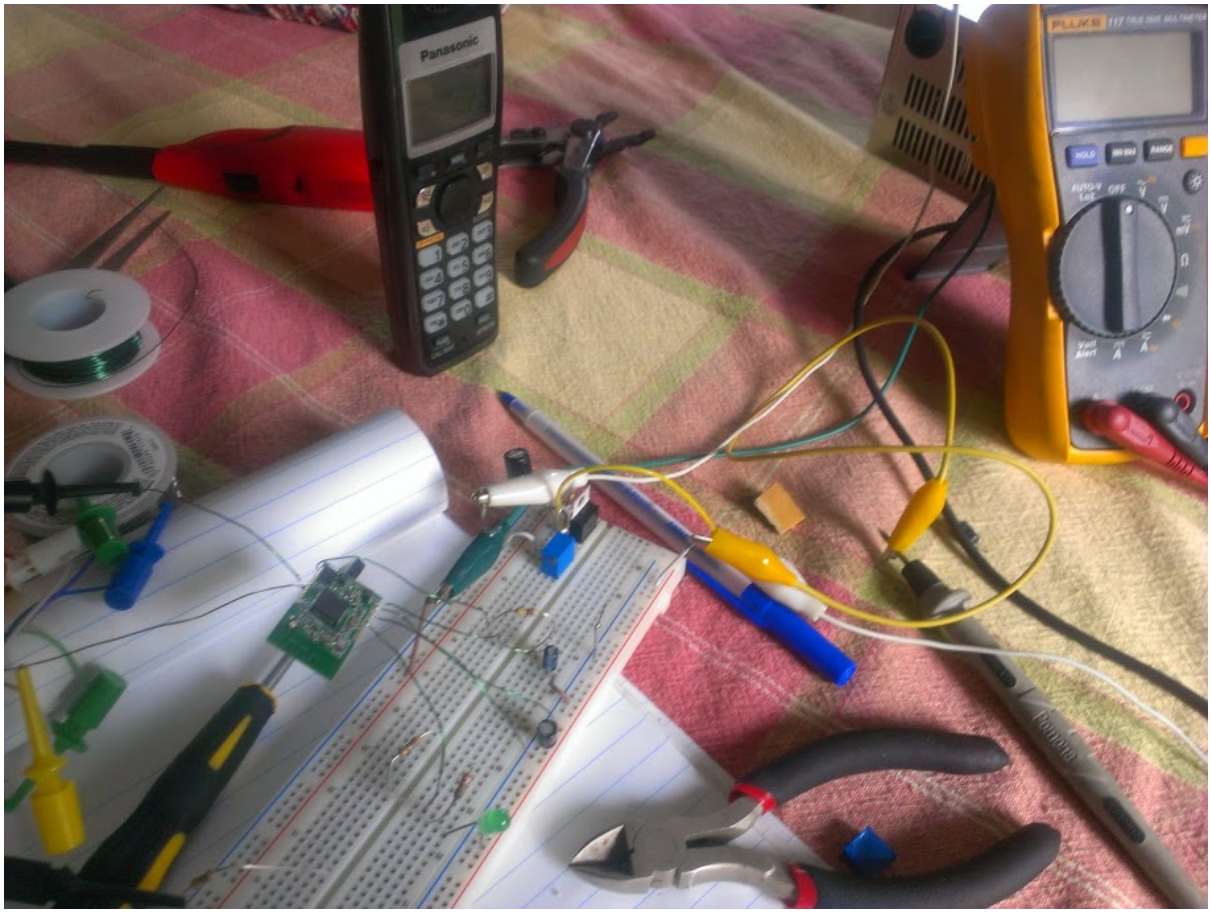

Figure 3: XS3868 module powered via LM317.

The quality of the audio was excellent for a US$4.00 module. When audio was idle, there appeared to be some type of static on the output. While searching for a solution to another problem I found a possible solution to the audio issue. I will discuss in the future.

On occasion, after power is removed, the module refuses to start back up unless power is completely removed from the circuit and the source pins shorted. I'm unsure of the cause, but hopefully a scope will revel an answer or a stable enough supply will eliminate the issue. The LED connect was different that described by *xpertvis* and found to be pin seven from the top right corner.

Next, I utilized a Bus Pirate to communicate to the module via the high-speed UART. The baud rate was 115200, 8bits, 1 stop bit and no parity. The OVC3860 chip supports external logic from 1.8V-3.3V so the Bus Pirate could communicate directly. Using the AT-Commands datasheet, it was possible to pause, play, and stop the audio on my phone from the device.[2] Unfortunately, a number of the commands listed were not supported, mainly the commands associated with HFP and HSP. In addition, a Chinese version of the datasheet lists the commands

[2]http://avrproject.ru/_fr/4/ovc3860-command.pdf

AT#MM and AT#MN to change the name of the bluetooth module and the security pin. Both of these commands were accessible. Below is copy of a Tera Term Pro session with local echo on.

```
 1  AT#MM                \\change module name
 2  MMThe Buddha Car
 3  AT#MN                \\change security pin
 4  MN0000
 5  AT#MN215
 6  MN215
 7  AT#MN31468
 8  MN31468
 9  AT#MY                \\firmware verson?
10  MW20090409
11
12  II
13
14  II
15  AT#MO0000            \\query AVRCP
16  ML3
17  AT#MG                \\enable auto-connect
18  OK
19  AT#MP                \\enable auto-answer
20  OK
```

I imagine that there are additional hidden AT commands. Possibly a command which would enable the required bluetooth profile. The OVC3680 utilizes an external 8K EEPROM to store configuration data. There is an AT command to read memory locations which might allow for a memory dump via script written for the Bus Pirate, otherwise the memory could be physically removed, read, and then examined.

The other possibility for the lack of HFP support could be my cellphone. I believe the command AT#CY queried HFP status and provided a response which would suggest support. Maybe ACVRCP and A2DP needs to be disable to enable HFP or HSP.

Day 3

I scraped my original plans of μM governed bluetooth module when I discover the original eBay page selling the XS3868 module did not list the HFP profiles. I don't know if the profiles are store on the EEPROM or activated from the configuration data stored on the EEPROM. Instead, I decided to pursue a mechanical switching method for the audio, and forget hands-free capabilities. I did work two addition modules which listed the HFP profiles. None of the eBay sellers list SPP profile, despite the chipset showing support. We will see.

The design intent behind this project was a finished product which looked factory installed. That limited the amount of physical manipulation for the radio face. Instead of using a physical button or switch to change the audio feed to the bluetooth module, I utilized a switched stereo jack. A switched stereo jack is a five pin device which contained two addition pin that are normally shorted to the tip and ring of the stereo jack. When the jack is inserted, it severs the connections. The bluetooth module was wired to the switch contacts, allowing audio to be pumped through the auxiliary input until a stereo jack was plugged in upon which the audio input would mechanically transition to the auxiliary cord.

The device was powered through radio's signal wire for an external amplifier. The signal is 12V so an LM317 was use to drop the voltage to 4.0V. The original plan was to use a LM7805 in series with a high power diode to drop the voltage 0.9-0.7 V, but that proved troublesome.

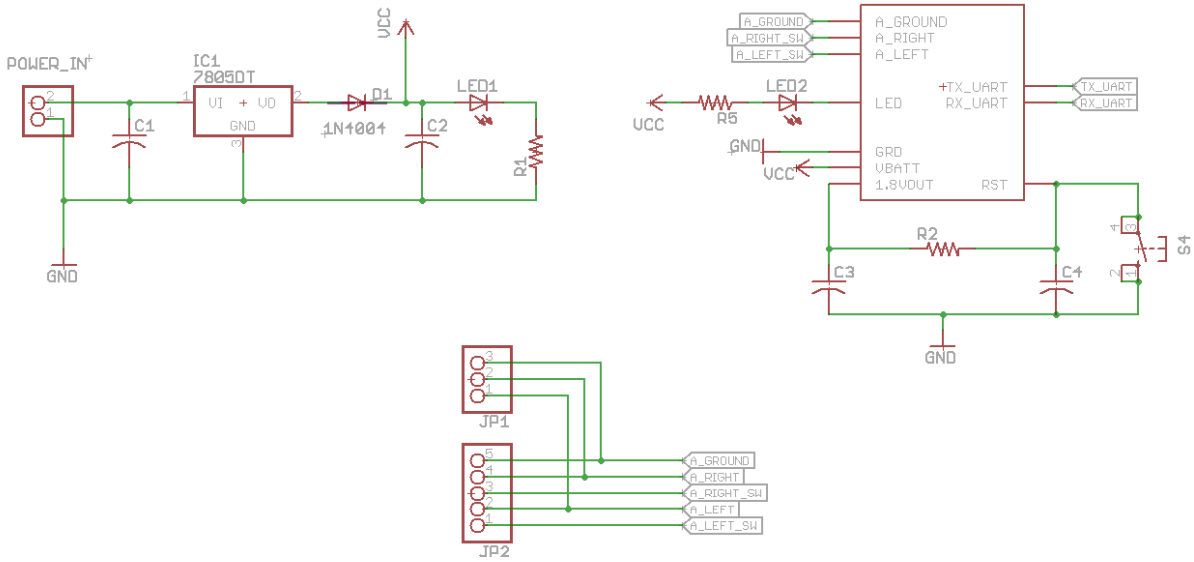

Figure 4: Schematic for a proposed solution.

A schematic describing this approach can be seen in figure 4, along with the connections for a mechanical switch. Not surprisingly, the voltage drop through a diode is proportional to the current it is sinking. I could increase the voltage drop by passing current through a 100Ω resister, but I could only obtain a $V_D < 0.6V$.

The current solution consists of a LM317 feeding 4.0V to the XS3868 module and the module wired to a mechanical switched auxiliary jack. It functions beautifully. There no noise as experienced earlier from the module and no type of ground-loop.[3]

[3]I don't fully understand this phenomenon, but I believe it happens when an audio source shares a ground with an amplifier and creates a feed-back loop. If you have addition information please point me in the right direction.