

W1XBS

We do what we must, because we can!

- [HOME](#)
- [About](#)
- [LICENSING](#)
- [Temp Control](#)

- [rss feed](#)

2014

05.01

[What A Case For Effective A/C and Dehumidification](#)

Category: [Uncategorized](#) / Tag: [air conditioning](#), [broadcast](#), [humidity](#) / Comments Off

WOW! What a way to get me out of my blogging slump! I burst forth with this spectacular failure.

One of my radio clients on Cape Cod has been beset with with a spate of severe failures of his 5KW class transmitter. First, a RF module died, reducing power by about 600 watts. I should have let enough alone! But in the interest of good engineering practice and doing the right thing for my client, we sent the transmitter back to the manufacturer (it's really not field serviceable).

The newly repaired transmitter lasted 1 day when he lost 2 more modules, resulting in a reduction of power by about half. It's a real gamble, depending on which multiple modules you lose, output power can vary from the value equal to the two modules to 1/2 of the power rated. The manufacturer was honestly embarrassed by this failure and did right by us, but the long and short of the story, the transmitter is starting to rot from the inside out due to the humid marine environment.

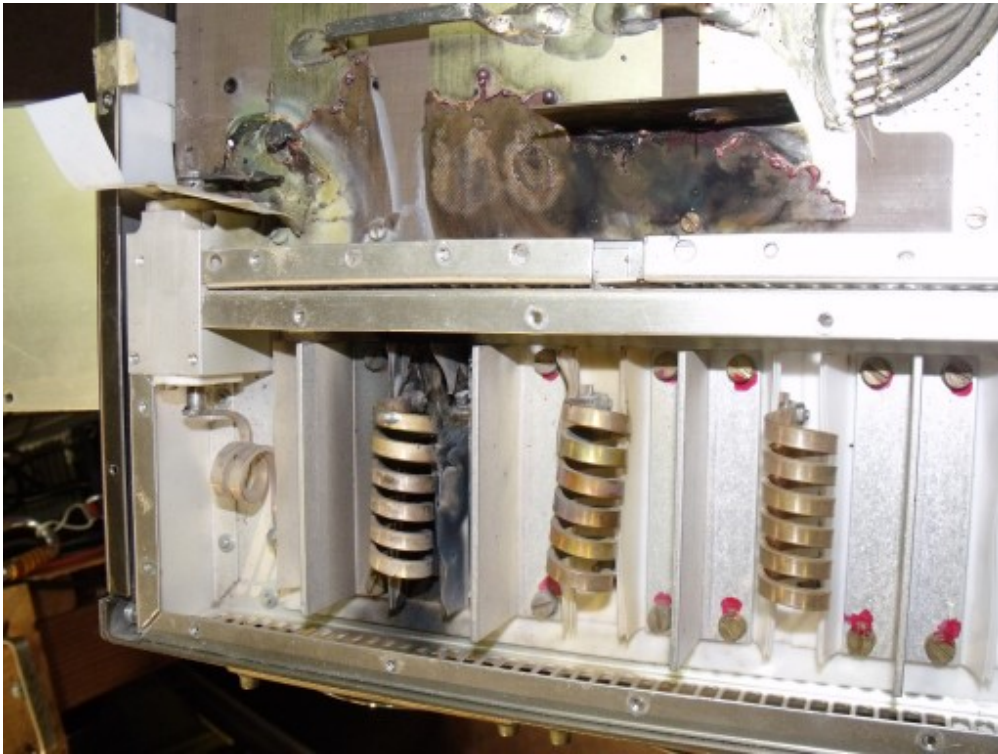
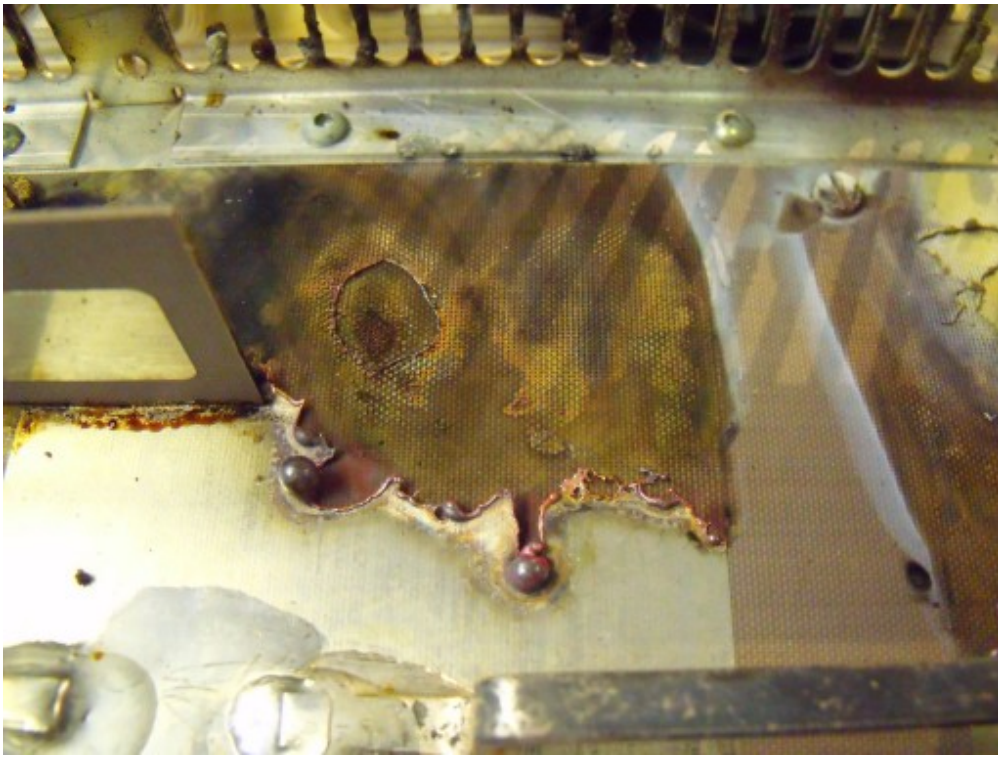
In all cases I managed to arrange a loan of a transmitter of about 1/2 the power. The two stations are good friends and help each other with donations, trades and promotions, so the engineering part of equipment loans was no big leap.

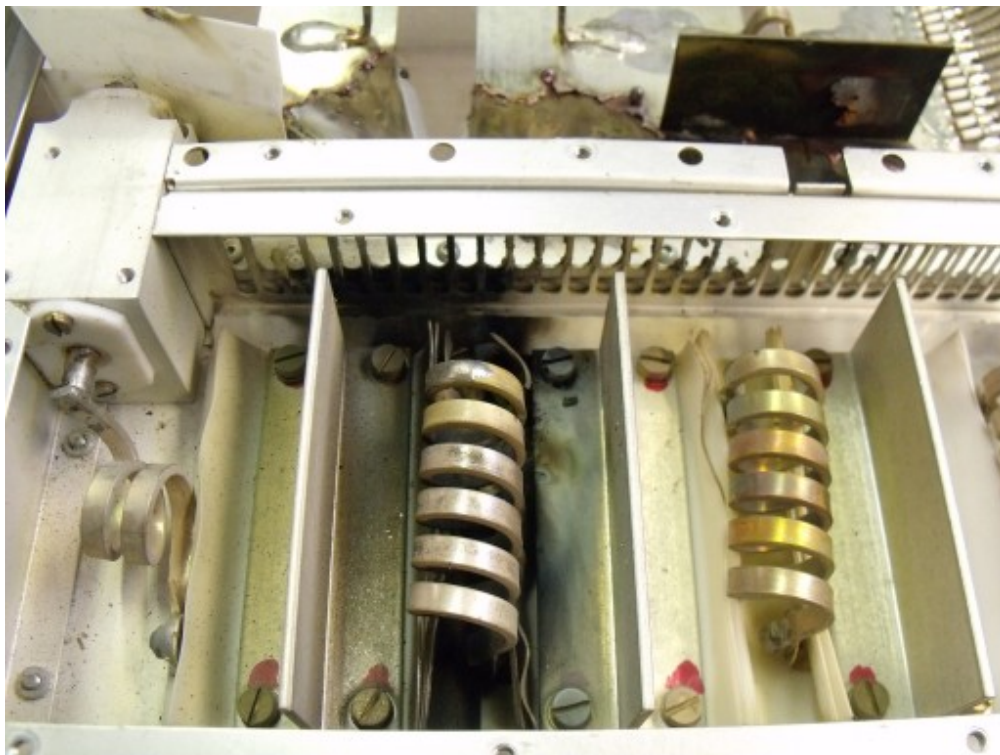
Now we are up to speed... the transmitter lasted about a month and suffered a total failure on a dark and stormy night, with no error logs, no alarms, it simply doesn't make any power, though you can see current consumption as it tries, but nothing comes out of the output. The manufacturer was totally stumped though he suspected the directional coupler, the very last device that measures forward and reflected power, was defective... oh boy was he optimistic!



We theorize that the air became humid during an intense rainstorm. The low pass filter developed condensation, and after a brief power outage, the condensation dripped, the voltages swung around finding their place on startup and an arc developed. A huge arc as shown by the pictures(s).

This is the downside of going with an “economizer” method of air conditioning. A high end A/C unit will pull in outside air during the winter months, but it takes into consideration humidity and dew point. In the early spring, it might be cool enough to still take in outside air, but it will be humid and wet. Add salt air to this mix, and things start to oxidise and rot.





So the long and short of this is, don't try to second guess your A/C scheme or your contract engineer. Though it may increase your electric bill a little, call this insurance! In the Northeast US I can see pulling in outside air from early November to early March, with the odd wet day off.

I hope to build up a humid, temp, air pressure, etc, sensor to control the A/C vs. the intake fan very soon. It will also log those parameters for later review and fine tuning. More on that later, as 90% of the work has been done, just not blogged.

2012
11.14

[Hacked Ikea Entertainment Center](#)

Category: [audio](#), [Building Stuff](#), [DTV](#), [Ikea](#), [Television](#) /

Tag: [center](#), [entertainment](#), [expedit](#), [hack](#), [hacker](#), [Ikea](#), [ikeahack](#), [ikeahacker](#) / Comments Off

With the acquisition of a large HDTV, its associated 5.1 surround sound system, cable box, home theater PC and a couple of DVD players, my existing TV table was woefully inadequate. In fact it began to collapse and finally disintegrated when I dismantled it to make room for the new IKEA hacked entertainment center.

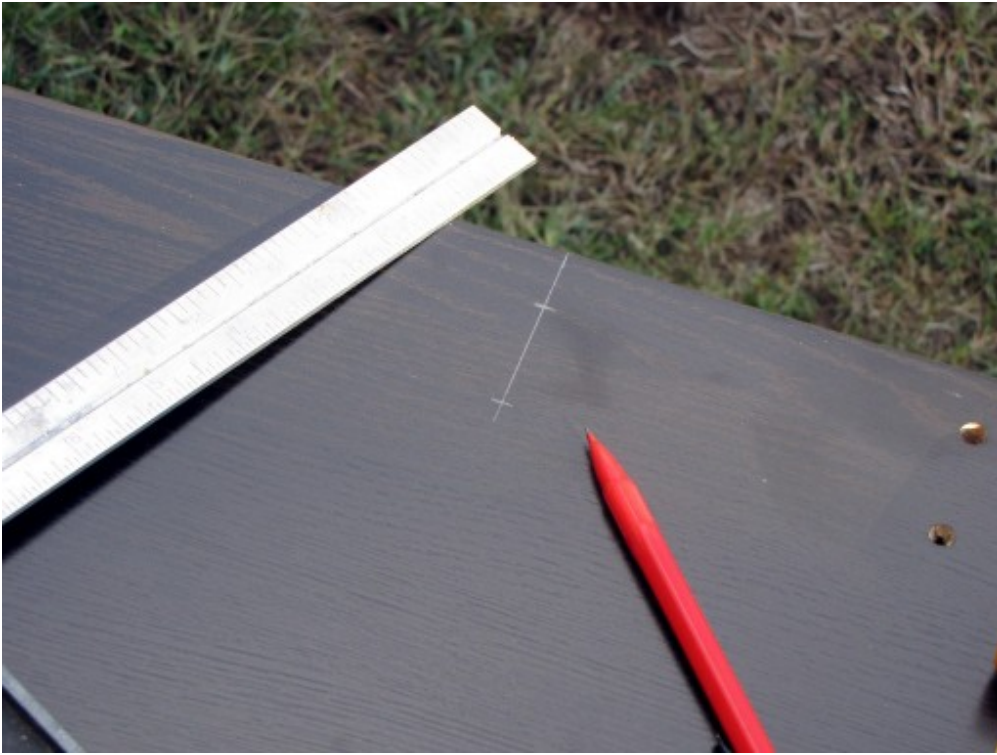


Many people re-purpose Ikea furniture and components to other uses, my desk is one such project, using a couple of Ikea cabinets and a few other odds and ends to comprise the drawers and supporting structure. This project is no different but is a bit more involved as it was necessary to move a few of the “shelves” to make compartments wide enough to accommodate the electronics.

It all started with [two 5×1 Xpedit bookshelves](#) hauled back from the local Ikea store.



Everything was unboxed and I carefully measured out where I wanted the “shelves” on the top unit to end up so it made three compartments rather than the five narrower compartments.



With the new dowel holes drilled, the bookshelf was assembled as normal.





And finally the two shelves (the bottom was left as is, simply did not install shelf #1 and #4) were screwed together from underneath to hide the screws.



A piece of 3/8" MDF was cut to size and screwed on the back, this gives the units a bit of added stability. The side facing inward was painted a shade of brown that matches the Expedit shelves. 1.5" holes were drilled at the bottom center of each compartment. Legs were fashioned out of three 2x4s cut to length to match the depth of the bookshelf and painted the same color as the MDF. Finally, the felt bumpers Ikea supplied were stuck to the 2x4s so they would not scratch the floor when the unit is moved.

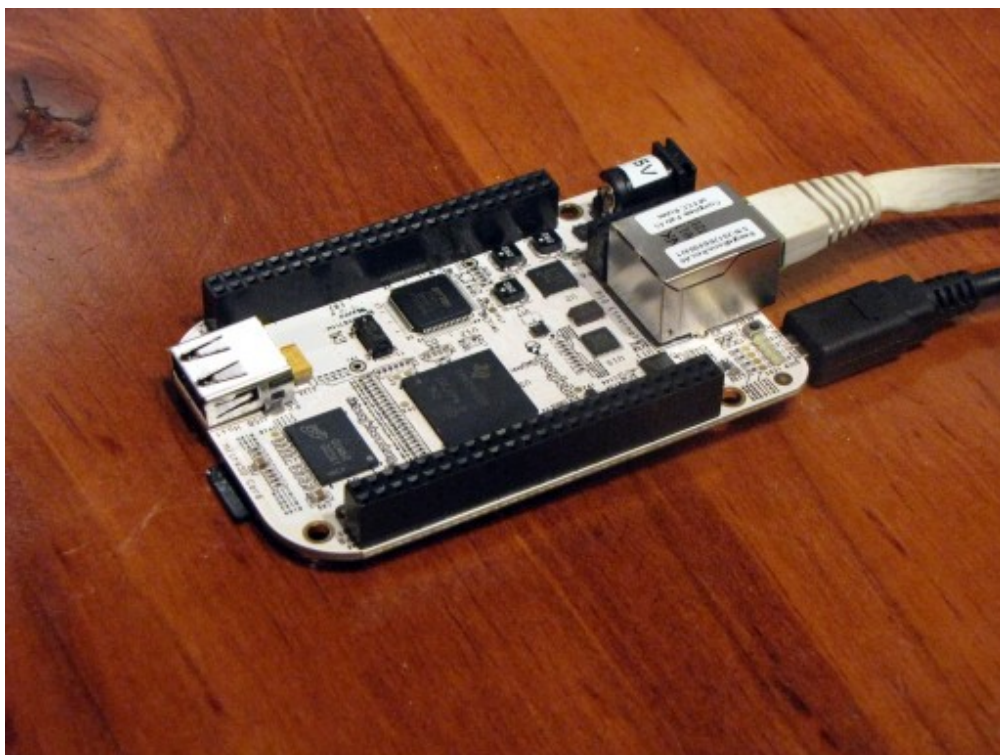
In closing, the unit is rock solid, heavy, and will hold almost anything available now or in the future.

2012
11.12

[New Development Platform – Beaglebone](#)

Category: [Beaglebone](#) / Tag: [Arduino](#), [beagleboard](#), [beaglebone](#), [development](#), [linux](#), [microcontroller](#), [webserver](#) /
Comments Off

I've used the [Arduino](#) platform and to a lesser extent, the PIC processors, as my primary development environment for a number of years. I have no intention of stopping, but some of my ideas have grown beyond the Arduino and it's time for a more powerful platform. Enter the [Beaglebone](#).



I'm happy to see manufacturers getting behind the community to implement their products. Not long ago, they would release something and go "*oh no, our reference applications are copyrighted*".

A few years ago TI and a consortium of other companies and distributors came out with the [Beagleboard](#), a full Linux computer in about the same size and shape of a small tablet, completely open source, hardware and software and it quickly became a huge success. About a year ago, they came out with a smaller version, the Beaglebone, with less features (i.e. no sound, no HDMI (no video at all for that matter), only one USB, etc), in about the same form factor as a smartphone and/or slightly larger but very similar to the Arduino. It even uses the same sort of shield scheme, called a "cape" in Beagle parlance.

Plugging it in and running it was no problem, the OS and many many utilities are already installed. It appears as a linux machine on your network running SSH and a WWW server. Also if you power it via USB, it appears as a drive on your Linux or Mac-OS machine automatically, [some driver installation](#) is needed on Windows. I used the drive feature the first day to read the [HTML5 presentation](#) and get up to speed with it and never considered it again.

I've already got a few projects in mind for this. My thermostat code is beginning to burst the Arduino at the seams, especially since I added a LCD interface. I want to add a nice graphic touch screen interface and this will definitely overwhelm the Arduino. Also my bike computer was already slated to go on an Arduino Mega, but for about the same price, the Beaglebone is much more powerful. And finally, I want to play with robotics a bit more, I've already ordered [a couple of USB webcams](#) to act as machine vision (and nanny cam when I travel).

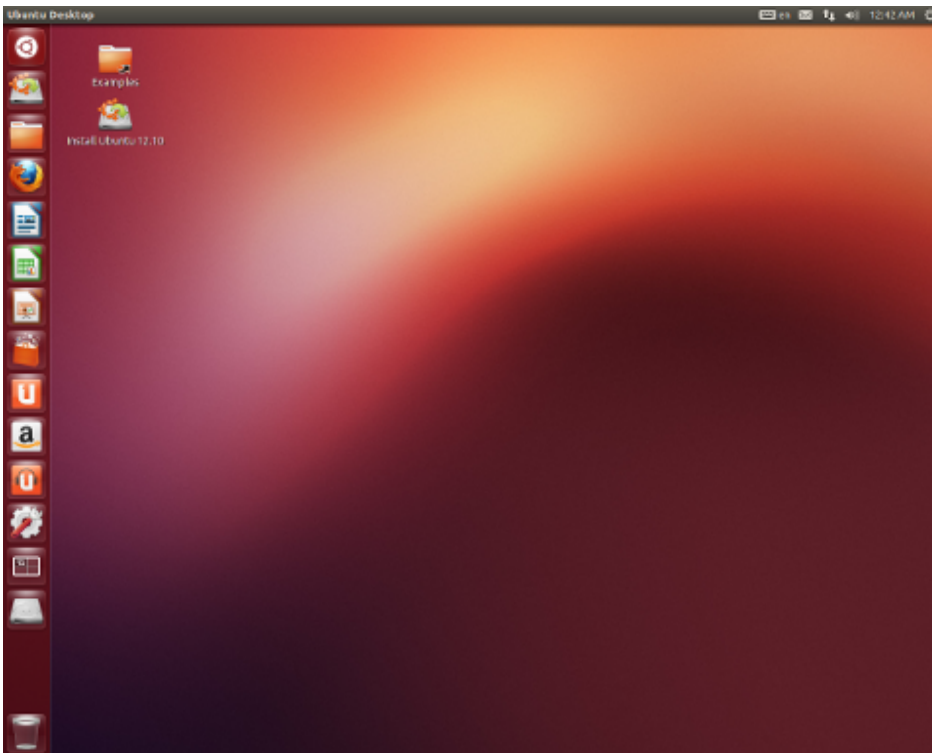
In only a few short days I have managed to accomplish alot. The next installation of this blog will be installing Ubuntu 12.10 on the Beaglebone and getting it ready for a Linux development environment.

2012
11.12

[Migrating To Linux](#)

Category: [Linux](#) / Tag: [12.04.1](#), [7](#), [desktop](#), [linux](#), [PC](#), [ubuntu](#), [Windows](#), [XP](#) / Comments Off

I've used Windows for a very long time, since 3.1! One day in July 2012, my main computer, running Windows XP stopped booting. One of the OS files had become corrupted and the repair tools offered on the XP disk only made things worse. After a few hours of playing I decided to scrap this install of Windows, salvage the license key for something else and migrate over to Linux *finally*. I do not regret it one bit.



It took me awhile to get used to things being in different places and different terminology but that was not a huge deal. For example...

- The minimize, maximize and X buttons are on the other corner of a window.
- Each window does not have its own menu, instead, the menu appears in the bar at the top of the desktop when that particular window has focus.
- There is no task bar out of the box, but I found something nice in the form of Dockey. It can make a static taskbar like Win7 or a sort of round zoomy one like Mac.

I do miss some of my windows exclusive programs, like AutoCad LT2000, some games and I have an AOL account for quick networking tests (your browsing goes through AOL proxies, not your IP address).

All in all, it took me about a month to get used to things, now it's second nature. I have no problem alternating between Linux and Windows XP or 7. I still run XP on the other computer right beside this one.

An added side benefit as this has set me up to do some serious development work on an embedded Linux microcontroller board, natively. All the tools are already in the box. I'll be writing more on that very soon.

2012
08.26

[Adding Spotify to MythTV](#)

Category: [audio](#), [Linux](#), [Spotify](#), [Ubuntu](#) / Tag: [11.10](#), [12.04](#), [linux](#), [mythtv](#), [spotify](#), [ubuntu](#) / Comments Off

Now I am mystified, the article detailing the “preview” version of the Spotify client for Linux is over 2 years old, it seems they discontinued development yet it works flawlessly. I’ve been using it on my [12.04 Precise Pangolin](#) desktop for a few months now and, I dont know why, finally hit on installing it on the home DVR. I’m now listing in glorious 5.1 Dolby Surround on my [Mythbuntu 0.24](#) installation. By the way that version of Ubuntu is 11.10 Oneric.

First things first, you will need to [install Spotify](#). You will find the full instructions here, but these are the basic steps.

```
# 1. Add this line to your list of repositories by
#   editing your /etc/apt/sources.list
deb http://repository.spotify.com stable non-free

# 2. If you want to verify the downloaded packages,
#   you will need to add our public key
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 94558F59

# 3. Run apt-get update
sudo apt-get update

# 4. Install spotify!
sudo apt-get install spotify-client
```

That’s IT! Pretty simple huh?

Now you will want to add a menu button on your MythTV install. [Those instructions](#) are a bit more involved but here is a page describing adding HuluDesktop to MythTV. Just replace references to huludesktop with spotify.

Simple huh?

2012
07.31

[Bike Computer and Hipster 10-speed Revival – Concept](#)

Category: [Arduino](#), [Bike Computer](#), [Building Stuff](#), [development](#), [Uncategorized](#) / Tag: [Arduino](#), [bicycle](#), [bike](#), [bike computer](#), [development](#), [lcd](#), [microcontroller](#) / Comments Off

I purchased a Schwinn Centennial bicycle from a professional picker one weekend for \$25. Its a late 70’s T-E-N speed... none of those extra gears, just ten speeds, 2 gears in front, 5 in back. It has the classic swept under handlebars of the era, the painful seat, but it’s built to last because of [Schwinn’s signature welds](#). It only took a few sprays of chain lube, WD-40 on the cables and some dérailleur adjustment to get everything working again and it had [it’s inaugural ride](#). Now to build a bike computer for it, building upon some of the work already done by others.

[\[Show picture list\]](#)





A bike computer is an evolution, not an all or nothing project. The beauty of microcontrollers is they can be added to, revised, features deleted, and can grow with the project as features are added. I have a million ideas from trip computation, GPS tracking, workout logging, speed and odometer, heck I even thought of implementing a subset of OBD-II! But for now, we will suffice to do only a few things, speed, distance, lap and display the results in realtime, essentially a homebrew speedometer available at any bike shop for about \$15. But with the microcontroller architecture, feature addition will be easy.

So with that in mind, first we will need a display.... I am unsure which one exactly yet, but it must be graphical and LCD. [This line from Adafruit](#) looks nice. Also this [old Nokia monochrome display](#) looks nice, we don't need color for this application.

Next we will need some sort of rotational sensor for the wheel, I have seen simple alarm contacts used, but I think I'll go more elegant with [a hall effect sensor and a magnet](#).

Next we need a platform, I am a fan and most familiar with the Arduino line, so we will use a classic [Arduino](#) (pro version 3.3V, no bells and whistles) and build around that. If we need more memory or power, we can always

upgrade to the Arduino Mega.

And finally, for safety's sake, we should add a flashy light on the back of the bike. I had long wanted to do a Morse Code message in the flashing, to amuse those that can read it, and to provide an erratic flash to make it more attention grabbing. So [a few bright LEDs](#) mounted under the seat will do the trick.

Oh, and we'll need some form of rechargeable battery, [Probably LiPo](#) a [charger](#) and a [battery gauge](#).

And there you have it, the initial design specifications for a bike computer. Now its off to gather parts.

2012
07.11

[Porting Temp Controller Code to Nanode5](#)

Category: [Arduino](#), [Building Stuff](#), [development](#), [Home Monitoring](#), [Internet Applications](#), [Nanode](#), [PHP](#), [Temperature Measurement](#) /

Tag: [Arduino](#), [flot](#), [graphing](#), [microcontroller](#), [mySQL](#), [Nanode](#), [PHP](#), [t](#), [temperature](#), [thermostat](#), [tmp-421](#), [TMP421](#), [webservice](#) / Comments Off

I had put the Arduino stuff on the back burner for awhile, concentrating on some web development work but I needed to track temperature at work and attempt to catch the air conditioning in the act of malfunctioning. I decided to finish my porting the temperature controller I made in 2010 with the Arduino / Ethernet shield to the [London Hackspace](#) – [Nanode-5](#), all with pretty good results.



If you look closely you will see a [TMP-421 temperature sensor board](#) installed on the analog header, as outlined by [Liquidware](#) and [ModernDevice](#), the vendor of this little breakout board. It utilizes the I2C port which occupies two of the analog I/O and the other two are used as power and ground. In this configuration it is possible to power down the chip for purposes of a reset or something, but this functionality is not used here.

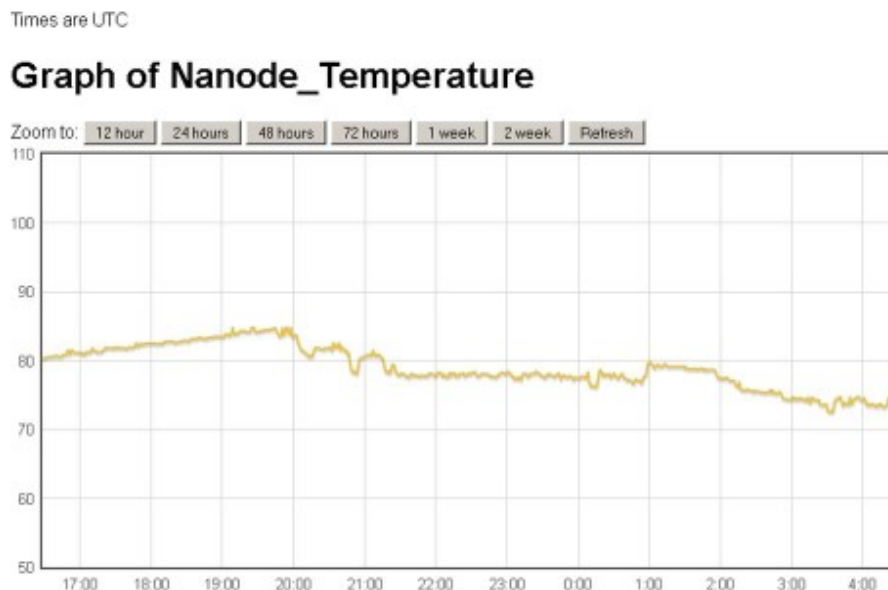
The accompanying sketch and php scripts on [github](#) basically do the following.

- Initialize the ethernet chip
- Initialize the I2C bus and the TMP-421 chip
- Read the temperature every 5 seconds
- Push the temperature to a waiting mySQL database every 60 seconds
- ???
- PROFIT!

I want to add a rudimentary thermostat and relay driver, and a simple webpage to read the temperature on demand and set the thermostat setpoint. So keep an eye on this blog and github for updates to the code. The final feature list will be...

- Temperature reporting and databasing (DONE)
- Temperature graphing (DONE)
- Simple hysteresis thermostat (IN PROGRESS)
- Webserver to display current temp.
- Webserver sets setpoint, persistantly through PROGMEM

Oh, I mentioned graphing.... as always click to display the latest graph...



I will release the graphing code later when I have added some functionality and cleaned it up a bit. But if you want to play with graphing in your own, this is based on the [Flot package](#). It's a little old now, but I am hesitant to change, it works! But there is better out there and I'll eventually get around to evaluating them.

And finally, to get this working on your server, first you will need a webhost with PHP and mySQL. There are many packages out there, look around. Then you will have to change the following lines of code,

In *nanode_temperature.pde*, between lines 45 to 60 change these parameters to match your setup

```

//*****
//**** BEGIN USER DEFINED VARIABLES ****
//*****
#define POST_RATE 60000 // rate to post to the db milliseconds (60 seconds)
#define UPDATE_RATE 5000 // rate to read the RSSI milliseconds (5 seconds)
#define TIME_OUT 30000 // time to wait for a response from internet (30 seconds)
float setpoint = 70.00; // default setpoint upon first use

```

```
float differential = 2.50; // differential
char HOSTNAME[] PROGMEM = "hostname.com"; // hostname of a shared hosting server here
static byte hisip[] = {***,***,***,***}; // ipaddress of the server here
#define HTTPPATH "/somewhere/nanode_temperature.php" // path to the PHP script
#define DATABASE "nanode_temperature" // database name
#define TABLE "node1" // table name
//*****
//*** END USER DEFINED VARIABLES ***
//*****
```

Also between lines 77 to 80, change this to your network scheme...

```
#else
static byte myip[] = {192,168,2,202}; // modify these addresses to match
static byte gwip[] = {192,168,2,1}; // your local network scheme.
#endif
```

And finally in line 260, change the following line to reflect the path to your php script,

```
ether.browseUrl(PSTR("/path/nanode_temperature.php"), paramString, HOSTNAME, my_result_cb);
```

In php.db, you must put your database credentials in there, and the database name you chose above,

```
$host = localhost;
$user = username;
$password = password;
$database = databasename;
```

Thats it! That should get you going with the functionality provided. I'll probably be adding more fields to the table, with on/off times, setpoints and the like, but I will write the php script such that it automatically adds them to your existing database, to make things simple.

Just to reiterate, [the code can be found here](#).

2012
01.16

[Moseley TRC-15 Remote Control Teardown](#)

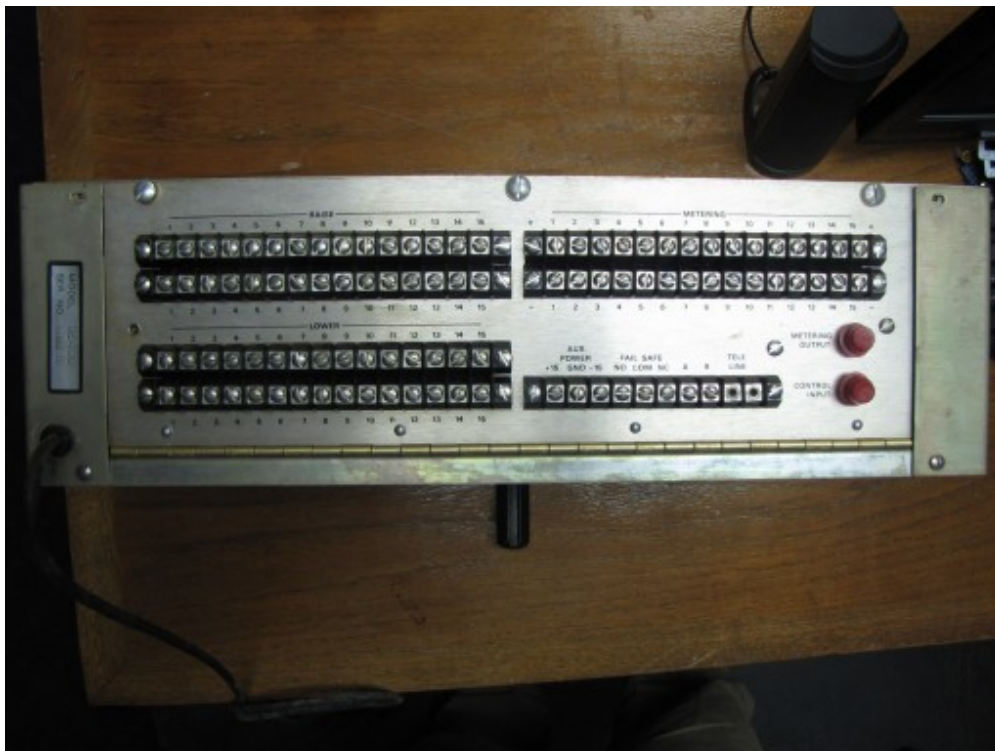
Category: [Building Stuff](#), [microcontrollers](#), [prototyping](#), [Remote Control](#), [Tearing Stuff Apart](#) /

Tag: [broadcast](#), [control](#), [development](#), [hallikainen](#), [microcontroller](#), [moseley](#), [moseley_trc-15](#), [remote](#), [remote control](#), [TEL-171](#), [TEL171](#), [trc-15](#) / Comments Off

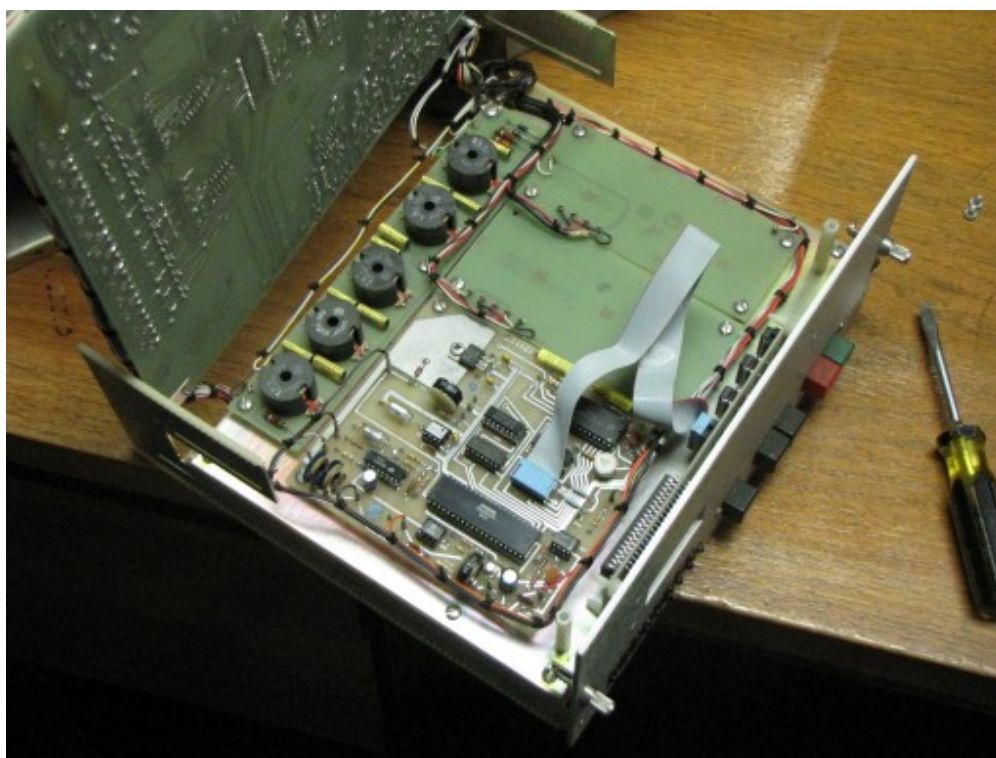
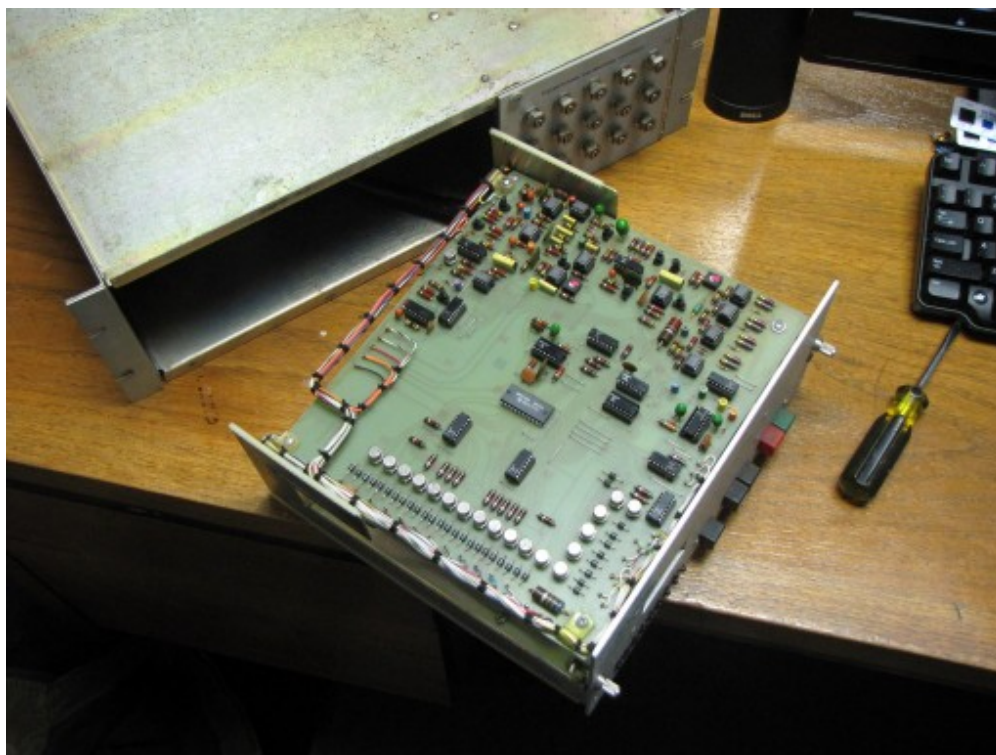
I've long wanted to construct an open source broadcast remote control. I rescued the Moseley TRC-15 out of an AM station some years ago, but never got around to playing with it. During a cleanup recently I looked into it and found that it was pretty obsolete and useless and I don't have the studio end of it. But it is a perfect enclosure to house my effort. So begins the teardown.



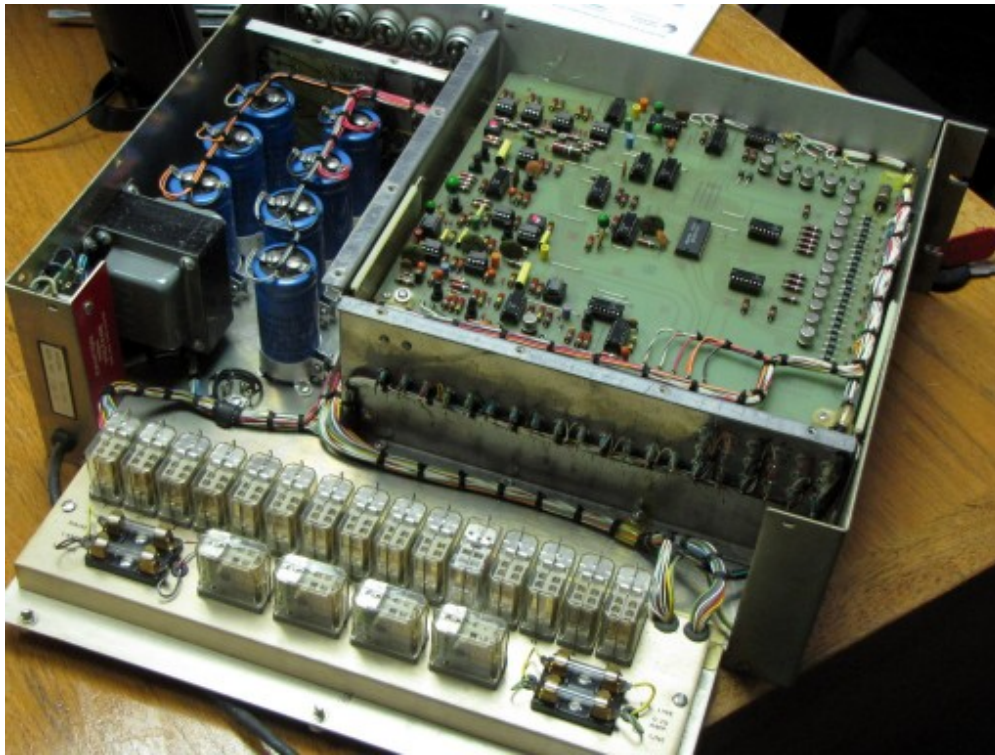
As you can see it has the Hallikainen & Friends TEL 171 retrofit. This converted the TRC-15 from analog to digital. Essentially the TRC-15 would send its data as analog tones, Hallikainen converted this to a digital system and added a LCD digital meter. Not bad for 1978!



This is the back panel of the TRC-15 and already we can see one of its limitations... where are the status (or logic) inputs? It only does command (or relay closure) and metering. Clearly some rearrangement will be in order.



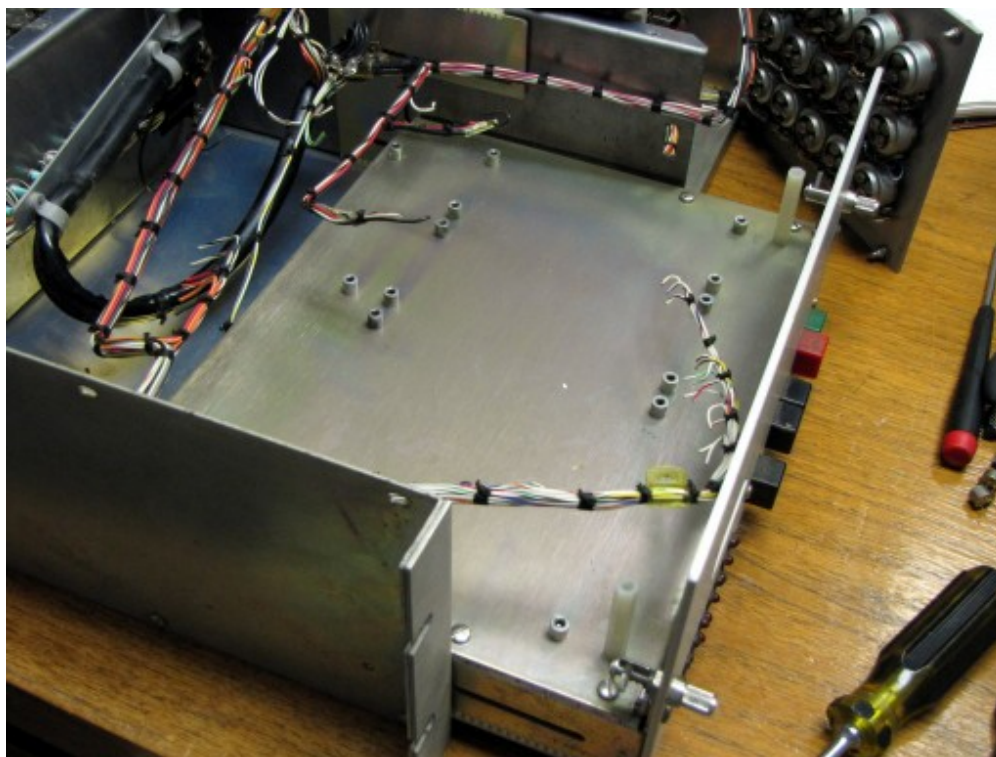
Flipping up the top board reveals more interesting components, but there is no microprocessor! Again, not bad for 1978!



Removal of the top cover and flipping open the back panel reveals the command relays and a very robust power supply.



A closeup of the power supply reveals the regulation and the analog calibration pots. I believe I will eliminate these and replace this with a blank panel.



The boards have been removed and work begun on layout and rearranging the I/O. Follow the category “Remote Control” for further updates.

2011
08.02

[Logging And Graphing Signal Strength](#)

Category: [antenna](#), [Arduino](#), [Building Stuff](#), [development](#), [microcontrollers](#), [PHP](#), [Processing Language](#), [prototyping](#), [SparkFun](#), [Tether](#) / Tag: [AM](#), [Arduino](#), [arduino shield](#), [FM](#), [LW](#), [radio](#), [si4735](#), [sparkfun](#), [SW](#) / Comments Off

“This was a triumph, I’m making a note here, huge success! It’s hard to overstate my satisfaction”

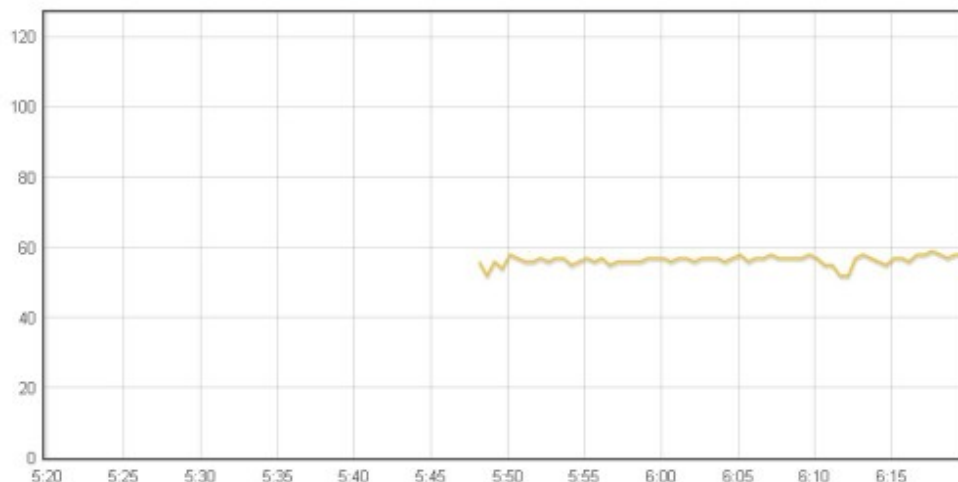
The reason I bought the [Sparkfun SI4735 Arduino Shield](#) last week was to graph and log a certain broadcast station’s signal strength to confirm or deny the “rumor” and “opinion” that they were not reducing power or changing pattern at night, a clear violation of [their license grant](#).

I got this working tonight!! I am now collecting data and waiting for ‘sunrise’ to see if their power jumps up, and ‘sunset’ tomorrow, to see if there is a corresponding drop in signal strength.

So far the graph is filling nicely as shown below.

WXCT Field Strength dbuV

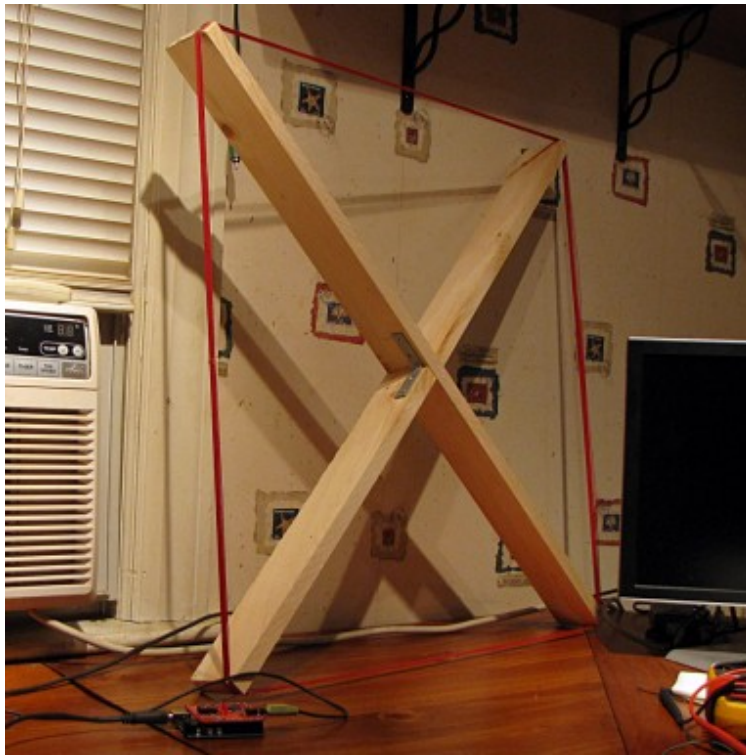
Zoom to:



The code is far too messy to release at this time, and I still have a few bugs to work out, but these are the basic steps that happen.

1. Arduino waits 30 seconds and then checks tuned frequency, received signal strength, signal to noise ratio and AGC value.
2. Arduino spits a comma delimited string out of its serial port.
3. A Processing script receives this string, parses out the data and writes it to a PHP script residing on this server.
4. The PHP script writes this data to the MySQL database.
5. A hapless internet viewer calls up a PHP script which reads the data out of the MySQL database and graphs it nicely using the Flot graphing package.
6. . . .
7. PROFIT!

It should be noted that I built an AM loop antenna for this project. It's only 10 turns, I was getting tired and sore from keeping constant tension on the wire and flopping the cross around, but it works nicely. I want to add more turns, weather proof it for outdoor use and make it tunable.



I will post construction details and a BOM in a future update when I am happy with the design.

For now, behold the graphy goodness at <http://w1xbs.com/flot/fieldstrength1.php>

2011
08.01

[Making The Sparkfun SI4735 AM & FM Receiver Shield Work!](#)

Category: [Arduino](#), [Building Stuff](#), [development](#), [microcontrollers](#), [prototyping](#), [SparkFun](#) /

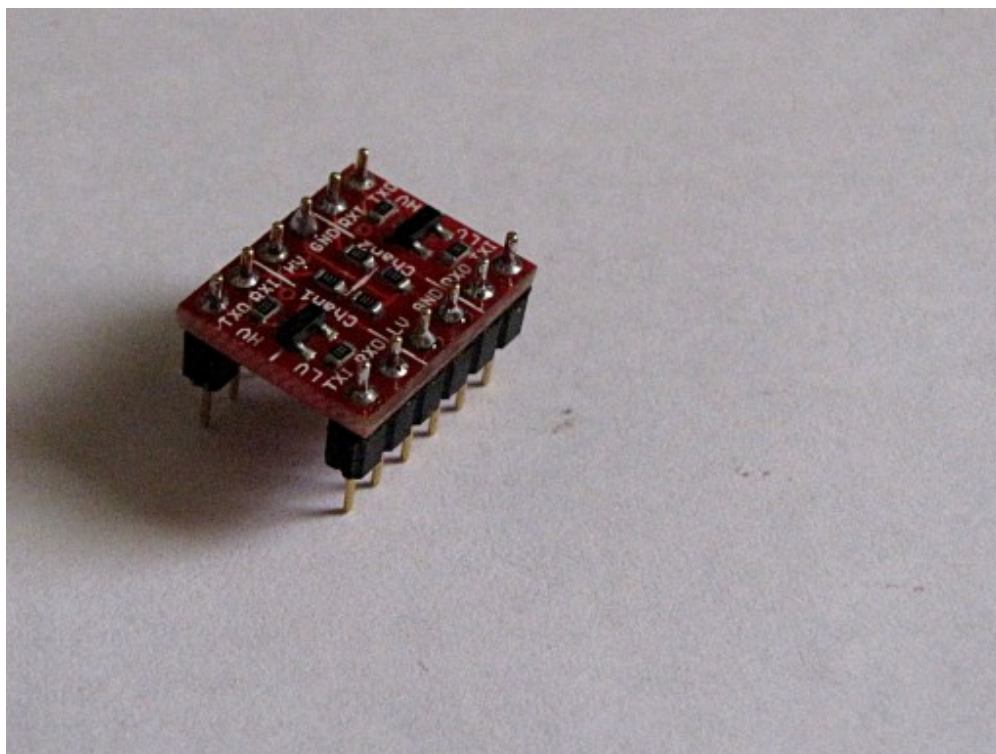
Tag: [AM](#), [Arduino](#), [arduino shield](#), [FM](#), [LW](#), [radio](#), [si4735](#), [sparkfun](#), [SW](#) / [1 comment](#)

About a month ago, Sparkfun Electronics released a [SI4735 AM & FM Receiver Shield](#) for the [Arduino](#). Based on the [Silicon Labs SI4735](#) receiver on a chip, it looked to be a very promising board. One small problem, they evidently never tested it on a 5V Arduino, or their Arduino's inputs were tolerant of being driven from a 3V source.



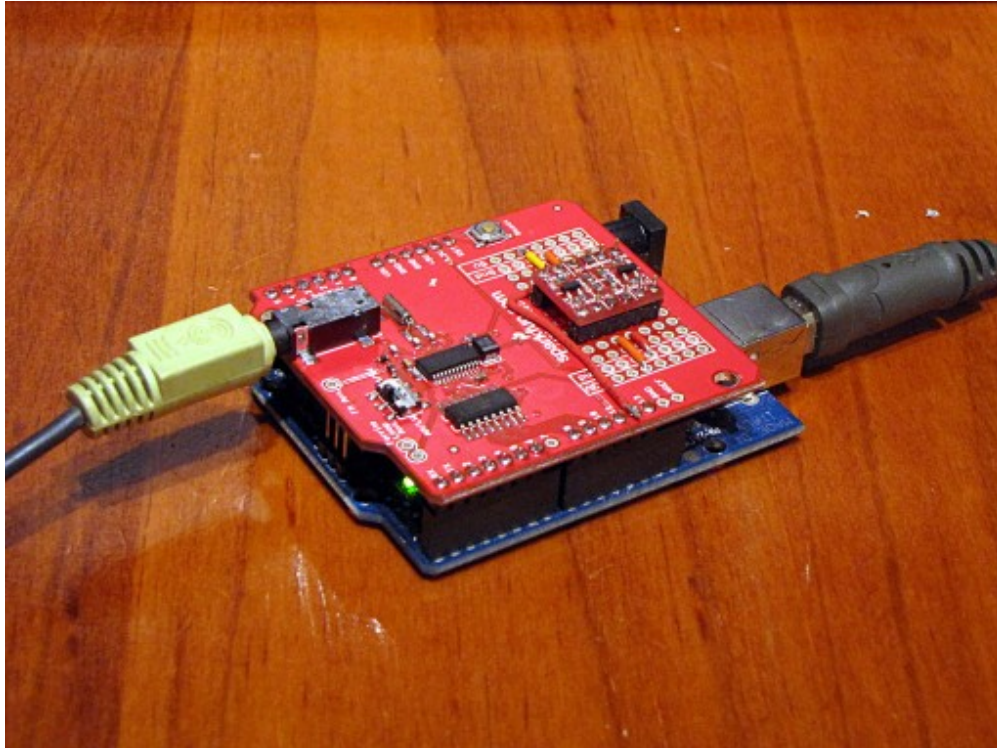
The commenters on [Sparkfun's product page](#) quickly discovered the problem and offered several solutions, however, [over on the Arduino forum](#), another poster found the real culprit and offered an incorrect, albeit working, solution. I decided I wanted to do this right, and reliably, so I decided on a proper level shifter.

I had a couple of [Sparkfun BOB-08745 Logic Level Converter](#) boards kicking around from an iPod interfacing project that never happened so I pressed one of them into service. I'll eventually replace the BOB-08745 with a single FET as soon as I find a proper one in a TO-92 case.



Once again... I have to wonder about Sparkfun's quality control and engineering checks, there were several complaints about the efficacy of the 5V to 3V (RX) section, read the comments on the product page but I don't need that functionality.

I added header pins to the board, so I can later use it elsewhere and on a breadboard and only soldered the pins needed to the SI4735 board. I cut the trace to the D12 line and soldered to a convenient via in this trace.



Some commenters on the Sparkfun product page also noted there is a missing `delay()` in the `getResponse` function, so change the following in `si4735.cpp` from...

```
void Si4735::getResponse(char * response)
{
  digitalWrite(SS, LOW);
  delay(1);
  spiTransfer(0xE0); //Set up to read the long response
  for(int i=0; i<16; i++)*response++ = spiTransfer(0x00); //Assign the response to the string.
  digitalWrite(SS, HIGH);
}
```

To...

```
void Si4735::getResponse(char * response)
{
  digitalWrite(SS, LOW);
  delay(1);
  spiTransfer(0xE0); //Set up to read the long response
  delay(1);
  for(int i=0; i<16; i++)*response++ = spiTransfer(0x00); //Assign the response to the string.
  digitalWrite(SS, HIGH);
}
```

After installation and the change above I was still getting responses I didn't expect and I found *yet another* problem. The source code of the `Si4735_SerialExample` states in the comments;

The command must be terminated with either a newline or carriage* return character. To enable this in the Arduino Serial Terminal, make sure the drop down menu located in the bottom right hand side of the * terminal window (not the baud rate one, but right next to that) reads 'Newline,' 'Carriage Return,' or 'Both NL and CR.'

If you set the line end to CR/LF, this has the effect of sending the command, then a null command, resulting in missing the response. Once I set that to either CR **or** LF, responses began to come in.

Now I am getting responses but they are still nonsensical! After spending **several hours** trying this and that, I found **even yet another problem**, the SPI interface was running too fast, missing bits. I found the following line in the si4735.cpp library

```
SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR1);
```

This runs the SPI interface at its fastest, 4MHz, resulting in missing bits. I set it to its slowest, 250KHz via the following code and now responses make sense!!!

```
SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR1)|(1<<SPR0);
```

I'll play a little further with SPI speed, but it's not really an issue, as long as the data is transferred in a few mS, as long as it happens. With that done, the board is 100% functional. I can tune it and read back important parameters. It's quite late now, I will continue with this tomorrow as something else concerns me, I am getting two different RSSI and SNR readings depending on where I read them.

[« Previous Entries](#)

• Location

 Help

Latitude has
been retired

Google Latitude was
retired on August 9th,
2013. Products retired
include Google
Latitude in Google
.. . . .

• Flickr Content

www.flickr.com

[What is this?](#)

• Archives

- [May 2014](#)
- [November 2012](#)
- [August 2012](#)
- [July 2012](#)
- [January 2012](#)
- [August 2011](#)
- [July 2011](#)
- [June 2011](#)
- [March 2011](#)
- [February 2011](#)
- [January 2011](#)
- [November 2010](#)
- [September 2010](#)
- [August 2010](#)
- [July 2010](#)
- [May 2010](#)
- [March 2010](#)
- [February 2010](#)
- [November 2009](#)
- [October 2009](#)
- [September 2009](#)
- [August 2009](#)
- [July 2009](#)
- [June 2009](#)
- [May 2009](#)
- [April 2009](#)
- [March 2009](#)
- [February 2009](#)

© W1XBS. Powered by [WordPress](#) and [Pyrmont V2](#).