Volume 20, Number 8 August, 2011

The Greensboro Amateur Radio Association

Feed Line

Providing Amateur Radio news for the Triad

Brian Wilson, NX4U, Editor www.w4gso.org info@w4gso.org

Arduinos, Ahoy!

By David Macchiarolo, AJ4TF

Some of you may know that I have an interest in embedded microcontrollers, both professionally and personally. I have worked with, over the years, various flavors of Intel microprocessors (8080, 8085, 8086) and Motorola 6800 & 68HC11 types. About 15 years ago I built from scratch a 68HC11 single board computer, complete with 32K of RAM, digital and analog input/output, and a serial port to control and download programs to it, just to see if I could make it work, which I did. Along with a freeware C compiler and assembler, and a boot loader program that I wrote, it was a complete embedded computer system, built on perfboard with point to point wiring. It's still sitting here, but that's another story for another time.

Recently I have been thinking about doing something more sophisticated for a home project. Microprocessors have come a long way in the last 15 years, evolving from just a processor, to more of a "system on a chip", containing lots of built-in peripherals from analog to digital converters, to built in networking interfaces, and more! There are a lot of different implementations out there, so I spent some time doing some research. I started seeing a lot of writing about one particular product that is aimed not at the engineering/professional market, but at a more casual user that may or may not know much about engineering or microprocessors, but has a desire to

see **Arduino**, page 7

NEXT MEETING

Monday, August 22

The next meeting of the Greensboro Amateur Radio Association will be Monday, August 22nd, at the Golden Corral Steak House, 4404 Landview Dr, Greensboro, NC 27407, off Wendover Ave. near Sam's Club.

Breaking News!

As of Friday, August 19, GARA has obtained a permit to hold a sprcial event at the Guilford Courthouse Nationl Military Park in March of 2012. N4G will be active on the weekend of March 16-18, 2012, from the Hoskins House. A big "Thank You" to David, AJ4TF, and others who have worked to make this a reality! Now it is time for everyone else to get involved!

Hammin' it up, Two-Wheel Mobile Or Tour de Triad

By Robert Ferguson, KD4TVE

Hi all,

Tour de Triad was Saturday, August 6, 2011. Serenity and I were there, needing another cup of coffee, at 7am. I was the first Goldwinger, but not the first to arrive. Many cyclists and Hams were there getting registered and setting up. Many of us were excited about Serenity, (my Goldwing) having a dual band on her. We all had high expectations on how this was going to work out.

The morning started out cool and damp. The rain was hard at the house. I woke early and listened to the rain. I tuned the house radio to the 145.25 to listen out for any announcements. It wasn't long till I heard several come over the air talking about the rain here and there. Soon, "The race will go on, rain or shine!" I didn't catch who said it but I'm glad it was put out there.

see **Tour**, page 3

GARA Meeting Minutes

Regular Meeting Minutes July 25, 2011

The Greensboro Amateur Radio Association held their Monthly meeting on July 25,2011, at the Golden Corral restaurant located near the intersection of Wendover Avenue and I-40. Chris Thompson, K4HC, called the meeting to order at 7:15pm and asked everyone to identify themselves.

Todd Smith, AK4TS, said that he is your at-large representative to the board so please contact him if you have anything that the board needs to be aware of. Todd also said that he wants to do some more antenna work at the Natural Science Center when the weather cools off some.

Bob Mays, KE4MOW, had nothing to report at this time.

Ernie Wall, NC4EW, gave the treasury report and stated that all major bills for the year have been paid. We currently have 106 members on the roster.

Al Allred, K4ZKQ, said that the income is ahead of forecast.

Greg Spencer, KG4UQV, had nothing to report.

Roy Smith, N4BYU, had nothing to report at this time.

David Macchiarolo, AJ4TF, gave the engineering report and said that the UHF cabinet is running cooler than before he added the vent to the rear of the cabinet. All equipment seems to be working properly.

Chris said that there is a soccer tournament coming up on Labor Day weekend. Please contact Allen Bradley, KD4IUN, if you want to volunteer to help.

Robert Ferguson, KD4TVE, spoke about working the Red Sleigh bike ride and how much he enjoyed it.

Bob Mays handed out flyers for the Open House coming up on October 1, 2011.

Chris said that Gerald Donnelly, N4PAX, has stepped down from the Vice-President position due to personal reasons.

Chris then introduced David Macchiarolo, AJ4TF, who gave a presentation about a proposed special event station at the Guilford Courthouse National Military Park to commemorate the 231st Anniversary of the Battle of Guilford Courthouse. David spoke about what would be required. Roy Smith, N4BYU, made a motion to have David make a formal proposal to the park, Terrell Brown, K4TLB, seconded, and the motion carried.

The meeting ended at 8:22pm.

Respectfully submitted by Greg Spencer, KG4UQV, GARA Secretary.

The Greensboro Amateur Radio Association

President Chris Thompson, K4HC
Vice-President vacant
Treasurer Ernie Wall, NC4EW
Secretary Greg Spencer, KG4UQV
Financial Al Allred, K4ZKQ
Engineering Chair David Macchiarolo, AJ4TF
Operations Bob Mays, KE4MOW
Member-at-Large Todd Smith, AK4TS
Member-at-Large Betty Williams, KC2BFN
Appointed Positions:
Webmaster Tom Forrest, N4GVK
Newsletter Editor Brian Wilson, NX4U

The Feed Line is ©2011 by the Greensboro Amateur Radio Association and published monthly. Our address is P.O. Box 7054, Greensboro, NC 27417. The purpose of the newsletter is to provide the club and prospective members information about the club and amateur radio in general. Material and information should reach the editor by the 2nd Friday of the month for the next edition of the newsletter. Opinions expressed in "The Feed Line" do not necessarily represent the views of the officers, directors, editor or members of the Greensboro Amateur Radio Association. Material may be reproduced, provided proper credit is given to GARA.

Todd Smith (AK4TS) will be riding for Team Champion in this year's ***Tour to Tanglewood***. This charity event benefits the Central NC Chapter of the **Multiple Sclerosis Society**. Please support Todd's efforts with a tax deductible donation.

Go to http://tinyurl.com/HelpFightMS

Board Meeting Minutes August 11, 2011

The GARA board of directors held their monthly meeting at the Benjamin Branch Library on August 11, 2011. The meeting was called to order at 7:11pm by Chris Thompson, K4HC. Those in attendance were Chris Thompson, K4HC; Roy Smith, N4BYU; Ernie Wall, NC4EW; Al Allred, K4ZKQ; David Macchiarolo, AJ4TF; and Greg Spencer, KG4UQV.

Chris, K4HC, read an e-mail from Bob Mays, KE4MOW, (who was unable to attend) concerning the upcoming Open House. The board then discussed the Open House and what will be featured this year.

Chris said that John Strandberg, W4DX, will head the nominating committee.

Roy said that Thomas Fleming, KJ4YDP; Terrell Brown, K4TLB; and Chan White, KJ4ADX, will form a soccer committee for GARA.

Ernie said that we are doing ok, and we still have money in the bank.

David said that there have been no complaints and that everything is working well. The UHF cabinet still seems to be running cooler than before the extra vent was installed.

Al said everything is agreeing with forecast.

David passed out a final draft of the proposal for the special event station for the boards review. The board discussed the proposal and Chris made a motion to approve the $50.00 application fee. Al seconded the motion, and the motion carried. David will send off the proposal and a money order this week. David said that he has reserved the call sign N4G for the event.

Roy made a motion to adjourn and David seconded.

Respectfully submitted by Greg Spencer, KG4UQV, GARA Secretary.

Tour

continued from page 1

At 8 am, the first group of riders left on the 85 mile route led by Jack Snead on his Blue 2005 GW. I was to lead the next group on the 63/50 mile runs. As luck would have it, I left my gloves in the trunk and needed them. So, I got off my bike got my gloves, and the cyclists left. So I followed and kept several different packs in sight. I ran the route keeping an eye on some of cyclists that seemed to be struggling.

As the morning wore on, the cyclists became more and more separated. There was at least one report of a cyclist getting lost. Although there were no official reports, at least one biker got off track too. I know how it is, you get to riding through beautiful country, you can forget yourself……..

As I ran the course, fewer and fewer riders were around. I came across Ralph DeGenova on his Big Bird yellow Goldwing after he swept the route. After confirming that the last two riders were in front us now, he decided to return to the start / finish line. I decided to ride with the last two riders. It was getting very hot. I was joined by another Ham and another Goldwinger. We tag teamed the intersections and made sure that they had all the water they needed. With the Yaesu FTM-10R mounted on Serenity, I was able to communicate with Net Control with location information.

On final approach to the finish line, we were greeted with cheers. One hot day, with friends

see **Goldwing**, page 9

From the President's Shack

I've had quite a bit going on in my personal life recently, and this underscores one on the principles of "The Amateur's Code" - balance. You are familiar with this, aren't you? If this doesn't sound familiar (or even if you are) take a few minutes to review it and apply it to your own actions. We all need to be reminded occasionally of the important things….

The Radio Amateur is......

Considerate: -- Never knowingly operates in such a way as to lessen the pleasure of others.

Loyal: -- Offers loyalty, encouragement and support to other amateurs, local clubs, through which amateur radio in the United States is represented nationally and internationally.

Progressive: -- With knowledge abreast of science, a well built and efficient station and operation above reproach.

Friendly: -- Slow and patient operating when requested; friendly advice and counsel to the beginner; kindly assistance, cooperation and consideration for the interests of others. These are hallmarks of the amateur spirit.

Balanced: -- Radio is an avocation, never interfering with the duties owed to family, job, school or community.

Patriotic: -- Station and skill always ready for service to country or community.

Until next month, keep havin' fun!

73,
Chris,
K4HC

THE GREENSBORO AMATEUR RADIO ASSOCIATION

invites you to our 2011 Open House ....

FREE !

Saturday
October 1, 2011
10 a.m. - 4 p.m.
at
Hinshaw United Methodist Church
4501 High Point Rd.
Greensboro, NC 27407

GARA
Greensboro Amateur Radio Association

W4GSO D-STAR DIGITAL 442.8625 Mhz.

- Displays of Amateur (Ham) Radio
- Amateur Radio Emergency Service
- Skywarn - Storm Spotting for the National Weather Service
- On-air demonstrations
- New and modern radio technology
- Information about obtaining/renewing your license
- Representatives from various amateur radio organizations

Whether you're a formerly or presently licensed amateur, or a complete newcomer to the hobby, this will be a chance for you to catch up on technology and what's going on in the Amateur Radio hobby.

If you have any questions please e-mail us at:
info@w4gso.org

Our web site is:
www.w4gso.org

Talk-in:
145.150 Mhz. 100 Hz. tone
D-Star - 442.8625 Mhz.

Ham Radio in the 21st Century

SKYWARN

Congratulations! You Passed!

Every month on the second Saturday, except in March when it is the third Saturday, W4VEC holds a testing session for those who live in the area. This provides an opportunity for individuals to be able to test for their amateur radio license or upgrade. Thanks to Glenda Nicholson, AG4NC, and a group of volunteers, each month there are new or upgraded hams in our area.

In August, we had 3 individuals who passed their test(s) and earn either their Technician, General or Extra class license.

Next month's testing session will be on September 10th at #3 Centerview Dr, Hickory Building, Greensboro, NC 27407 at 9:00am. While walk-ins are welcome contacting the lead VE is preferred. Please contact Glenda Nicholson at (336) 674-3810 or by email at ag4nc@bellsouth.net

Walter E Gaither - KC4KZR, Upgrade to Extra

Jonathan L Kuster - KK4DPQ, Passed Tech

Colin J Aubrey - KK4DPR, Passed Tech

All Photos Courtesy of Todd Smith, AK4TS

During the testing session the Volunteer Examiners watch while the participants work to answer the questions correctly. After the examinees are finished the test are graded and checked by at least 3 VEs.

Engineering Update for August, 2011

From David Macchiarolo, AJ4TF

I really don't have a lot to report on this month. Your repeater equipment is running well, and staying nice and cool. I have not heard, nor have I received any reports of the interference that sometimes desensitizes the VHF repeater input. I did hear however, on a couple of days in the last month, a band opening on 2 meters where we were lucky enough to catch the tails of another repeater on the same frequency pair as our machine... most likely it is either K4USC in Union, SC, or N3ZE in Radford,VA. Unfortunately, there is not a lot we can do about that, but it wasn't too annoying, since it could only be heard during the short interval between unkeying your transmitter, and the repeater transmitter dropping, and besides, it doesn't happen very often anyway.

I have not gotten around yet to setting up the macro on the D-Star controller to automatically link reflector 30C. As you recall last month, I described this reflector that is based in Atlanta, and has quite a bit of drive-time traffic on it, including some employees of the Ham Radio Outlet store there. I'll get that done soon enough. For those of you with D-Star equipment, give a listen, and join in the conversations! For those of you without D-Star equipment, go get some... there are many options to get you on the air with D-Star!

As always, if you have any questions or concerns about the club repeaters, either VHF or D-Star, please contact me at aj4tf@arrl.net

73,
AJ4TF

Silent Key:

Charles Lawrence "Jack" Cockman Jr., K4DRE

GREENSBORO — Mr. Charles "Jack" Lawrence Cockman, Jr., age 86, passed away Thursday August 11, 2011 at Moses Cone Hospital.

A graveside service will be 2 p.m. Monday at Lakeview Memorial Park.

Jack was a local radio disc jockey and musician for many years in the Triad. He was also preceded in death by his son James Cockman.

He is survived by his caregiver Marlene Harrold of Liberty; sisters Betty Harrington and Hilda Reid both of Greensboro.

Memorials may be directed to the donor's choice.

George Brothers Funeral Service is assisting the Cockman family.

Published in News Record *on August 14, 2011*

For Sale

Contact Tom Forrest, N4GVK, n4gvk@bellsouth.net or call 336-202-8650

HEATHKIT 1 KW LINEAR AMP. SB-200 w/ Manual

Was purchased a while back. Never used it as I don't have the antenna system to do it justice. It was suppose to be in good working order when I purchased it. This is a ! KW on CW and 1200 on SSB using 2 - 572-B tubes - wired for 110, but can be wired for 220 Volts. Asking $300 (they have been advertised on QRZ for $400) - sold as-is, no warranty. No shipping, pick-up only. Over all it's very clean. More information: www.rigreference.com/rig/5254-Heathkit_SB_200

Arduino

continued from page 1

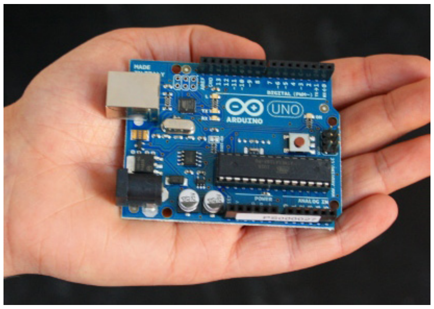
make some object or machine "smart". This platform is known as the "Arduino" (ar-dween-o). Even though it might be too "simple" for someone that has experience with microcontrollers, sometimes, simple is good if one values their time.

What is an Arduino?

[From http://www.arduino.cc/] Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

In other words, a simple microcontroller platform that is (theoretically) easy to use for a variety of tasks. You can probably immediately think of several ham radio-related uses for something like this.

As you can see, it is quite small.



Arduino Uno

The processor is based on the "AVR" architecture, an 8-bit RISC (reduced instruction set) microcontroller which was developed by two students in Norway and later sold to Atmel for further development in 1996. It has an efficient instruction set and architecture optimized for executing code generated from higher level languages such as C.

Hardware

The basic platform is the Arduino Uno, a microcontroller board based on the Atmel AVR Atmega328 microcontroller. It has 14 digital input/output pins , 6 10-bit A/D analog inputs, a 16 MHz crystal oscillator, and a USB connection for communicating with a PC during program development, and costs around $30. The board can be powered directly from the USB port 5V power, but for stand-alone applications, the board has an on-board voltage regulator to generate the 5V supply from externally-supplied 7 – 12 VDC, or if you have a regulated 5V supply available, you can connect it directly.

The Arduino platform is designed to be expanded by the addition of plug-in boards that are known in Arduino-speak (for some reason) as "shields". There are many different types of shields that are available. I selected the Uno, along with an Ethernet shield (which provides Ethernet connectivity) and a prototyping shield, which is basically empty and has room for adding additional interface circuitry. There are several vendors who sell the Arduino line, but the most popular seems to be Sparkfun Electronics (www.sparkfun.com). I ordered my hardware items on June 20th 2011.

Software

Being an open-source platform, the Arduino software package is, of course, free for the downloading. As most of you know, I don't like to use Windows at home. My shack computer is running Fedora 9 32-bit Linux, so that's what I wanted to set the Arduino software up for. There are some prerequisites that are needed, mainly, extensions to the standard Linux C and C++ compiler known as "gcc" that handle the Atmel AVR processor suite. These extensions turn the gcc compiler into what is known as a "cross-compiler": simply put, it's a compiler that creates executable code for a "target" processor that is something other than what the complier is running on. All of the required the AVR libraries can be downloaded from the Fedora 9 software repository. Also required is the Java Runtime Engine (JRE) which can be quickly downloaded from Sun's website. All of these instructions are fairly clear on the Arduino web site. Once all of the support software is installed, then the Arduino software can be downloaded and installed. I installed version 22 of the Arduino package, started it up, loaded up one of the example programs, and tried to compile it... uh oh, something is wrong, I am getting a strange error message! Back to the Ardunio web site, and a few minutes of searching found the answer. The AVR library that comes with Fedora 9 has a missing file

see More **Arduino**, page 8

Doggett Bar-B-Que

This years Doggett Bar-B-Que has been scheduled for October 1 at 6pm. If you haven't attended in the past, this time of food and fun is open to all radio amateurs.

It is a GREAT time and GREAT food. The Doggett family loves the amateur community and provides this as a time to enjoy some good food and get to know other amateurs.

Please RSVP to John Doggett at dogg7460@bellsouth.net and let him know if you are coming. He can also provide directions.

He has said it will be at the same location as last year.

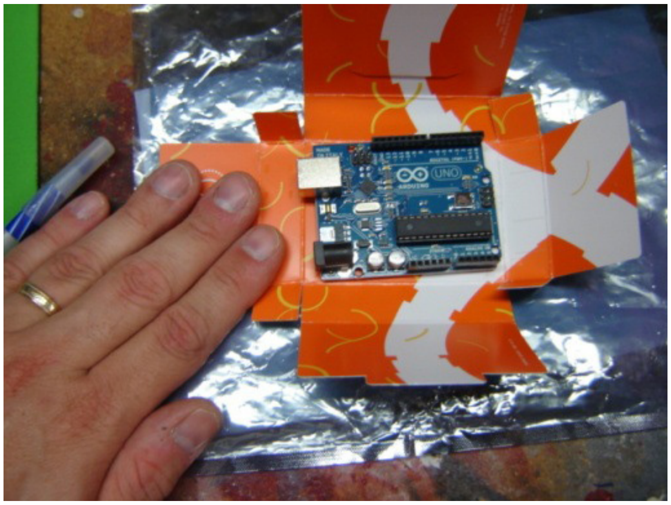
More Arduino

continued from page 7

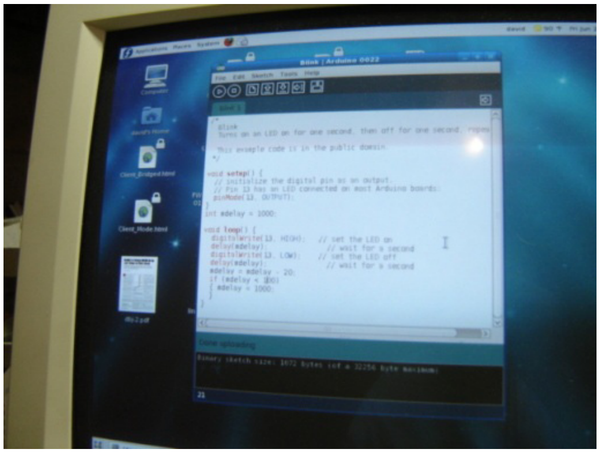
in it, so a newer version of the library needs to be installed. I downloaded avr-lib 1.6.6 as raw source code, and used the cross-compiler to generate a new library. Once it was installed into the proper location (/usr/avr) the software was able to compile Arduino programs. Ardunio-speak refers to programs for the target hardware as "sketches" for some reason, but I don't like that terminology so I'll stick to "programs". Arduino programs are written in the C language. There are many references to this programming language out there, so I don't plan to cover that in any detail here. At this point the software was ready for the hardware to arrive.

Initial Checkout

The package from Sparkfun Electronics arrived in 3 days. I really didn't appreciate how small the Arduino platform is until I opened up the box; it is the size of a credit card. Unfortunately, I thought I would have the right interface cable in my junk box, but I didn't...a USB A-to-B cable is needed (like one would get with a printer). A quick trip to a local computer shop yielded the right cable. I asked if they had any used ones, so I got the cable for a buck.



Modern Linux distributions have a hot-plug facility called "udev" which constantly monitors the hardware and detects when something new shows up (the analogous function to Windows plug and play). I watched the device table as I plugged the Arduino in, and sure enough, a new device was added "/dev/ttyACM0" which is a generic serial device. To make sure things were working right, I plugged and unplugged the Arduino several times, and each time the ACM0 entry appeared and disappeared as expected.



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repea
  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}
int mdelay = 1000;

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(mdelay);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(mdelay);              // wait for a second
  mdelay = mdelay - 20;
  if (mdelay < 100)
  { mdelay = 1000;
  }
}
```

```
Binary sketch size: 1072 bytes (of a 32256 byte maximum)
```

Arduino Software running on Fedora Linux

I then started the Ardunio software and selected one of the example programs which does nothing but blink a built-in LED on the Arduino board. Within 15 seconds, I had it complied into a binary, uploaded into the Arduino board, and, lo and behold, the LED started blinking. That was pretty easy!

Sensors and Analog Inputs

Since the Arduino has 6 analog inputs and 14 digital I/O pins, it should be easy to interface some sensors to it. I started out with a simple temperature sensor, the LM335. This device comes in a 3-pin TO-92 package, and is basically a calibrated Zener diode. The reverse Zener voltage is calibrated to the absolute temperature in degrees Kelvin at 0.01 V / degree. So, in other words, at room temperature of 25°C (equal to 273 + 25 = 298°K) the reverse Zener voltage is 2.980 volts. All we need is to provide a small reverse bias current (~ 1mA) and read the voltage. Using a 5V supply, that will require about 2 K ohms in series with the sensor, I used a 2.2K resistor that I had available.

By default, the A/D reference voltage that is used is the power supply voltage on the board, which, for the Arduino Uno, is 5V. The A/D system allows an external reference to be used, and it can be connected via software control, or an internal 1.1 V reference on the ATMega 328 chip can be selected. For now, I'll just use the default 5V reference.

When commanded to perform an A/D conversion, the A/D provides a 10-bit number (2^10 = 1024, output is from 0-1023) which represents the input as compared to the reference voltage. So, to get the actual voltage, simply multiply the reading by 5 / 1024 to get the actual voltage. Since this sensor is outputting a voltage that represents the temperature, to get a temperature reading we would convert the A/D value as follows:

Voltage = A/D reading / 1024 * 5.0
Temperature (°C) = (Voltage * 100) – 273.15

see **Sensors**, page 10

Goldwing

continued from page 3

new and old.

In an email from Ralph, (one of the Goldwing organizers), he says, "We made a big difference by our presence among the Bicyclist. Also thanks for putting up with the weather. I feel, this was a better situation than riding in 100+ degrees with high humidity. I'm sure you all experienced something memorable that happened with a bicyclist today. One of mine was finding a rider who was suppose to be on the 63 mile run, at Old Greensboro & Yokley roads and did not have a clue how he got there. He was about out of steam so I had him sagged back to RS1 @ the Union Cross Fire Station to get him back on track. All and all there were few incidents that I know of and all returned safe and sound. Speaking of great communications, The HAM radio volunteers did a great job as always. Robert Ferguson with his mobile HAM radio set up on his 1800 worked great. He could stay in contact with the HAM base 25+ miles away. I gotta get me one of them!!"

It seems that the word is getting out.

It was really great stopping at different rest stops and talking with all the different people that came together to support this worthy cause. There were the Goldwingers, the Hams, the cyclists, EMS, Sherriff deputies, North Carolina State Highway Patrol, the city Police Departments, and those that just wanted to volunteer because it felt good.

I believe for me, that there were several reasons. Some of these you will agree with, some you won't. We all volunteer for our own reasons.

1. I really enjoy riding my bike. (DUH!!)
2. I love talking on the radio. (Again DUH!!)
3. I like helping others. (Well, DUH!!)
4. The camaraderie with others.
5. Support of a good cause.

I really enjoy riding my bike. I have taken the time to learn to ride it aggressively, precisely, but most importantly, safely. I am sure that I am preaching to the choir but, safety is always first and foremost. Keeping your first aid kits up and ensuring you actually have them with you and in a place you can get to them. Keep your first aid / CPR training current. I have taken advanced rider training. Also, I am planning on taking the North Carolina Motorcycle Patrol training in September.

These types of exercises allow us to train for the emergencies that will arise.

Ride Safe – Look Twice – Save a life.

Sensors

continued from page 8

The C code that performs this looks like this. Any text past the double slash (//) is a comment and is not part of the program. The #define statement simply defines a constant that can be reused over and over, and only has to be defined once in the program. Other than that, I don't plan to go into a tutorial on the C language, but it should be fairly obvious what is happening.

```
#define REFVAL 5.00    // on-board analog reference voltage
#define MAXCOUNT 1024    // number of counts for a 10-bit A/D converter
rd = (REFVAL * ((float) analogRead(chan) / (float) MAXCOUNT));
tmp = (rd * 100.0) - 273.15;
```

Additionally, I can take multiple readings and average them to filter out low frequency noise (5 in this case):

```
 for (j=0; j<5; j++)
{
rd = rd + (REFVAL * ((float) analogRead(chan) / (float) MAXCOUNT));
delay (2);
}
data = rd / 5.0;
```

And, I can make another loop that reads all six channels, and put it all together into a function (subroutine) that my main program can call any time it needs to make A/D measurements:

```
int readAnalog(float d[])  // read all the channels, convert to float voltage
  {
   int chan;
   float rd;
   int j;
      for (chan=0; chan<NUMCHANNELS; chan++)    // loop through the channels
 { // read it 5 times and and average
    rd = 0.0;          // start over at 0.0 value
    for (j=0; j<5; j++)      // perform 5 times
    {
     // convert a/d counts to voltage
       rd = rd + (REFVAL * ((float) analogRead(chan) / (float) MAXCOUNT));
     delay (2);  // wait 2 milliseconds for the next channel
    }
  d[chan] = rd / 5.0;      // get the average value
  }
 }              // return to calling program
```

In the main program, I define an array of 6 floating point numbers called "readings":

```
float readings[6];
```

The index value starts at 0, and goes up to the N-1 where N is the number of values that I defined (0 through 5 gives you 6 numbers... count it on your fingers if you don't believe it). Don't get confused in the C language about arrays, just remember that the first index value is always 0. When I need to measure the A/D channels, all I have to do is "call" my function like this:

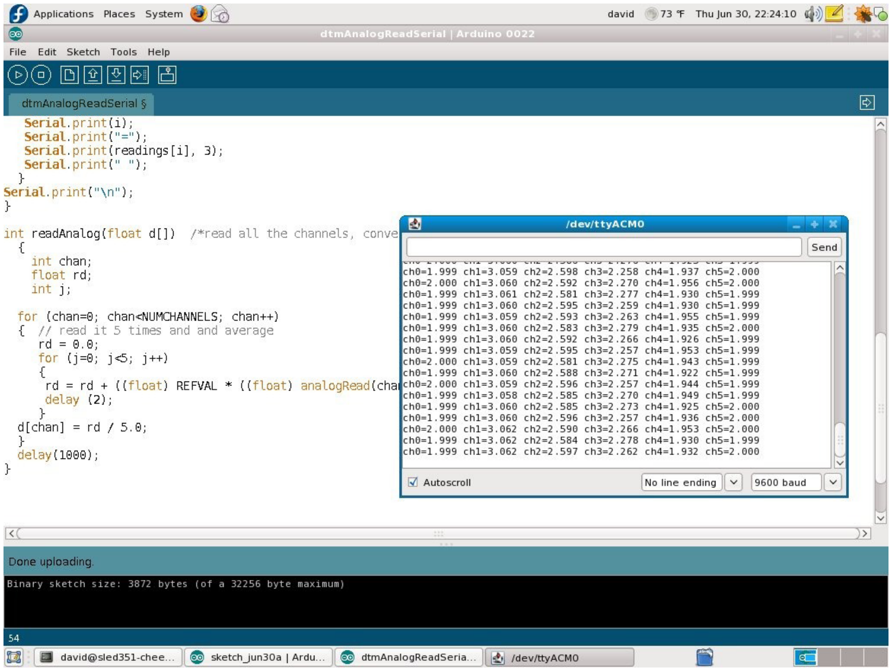
```
readAnalog(readings);
```

When the function returns to the calling program, the array readings will contain the A/D values, in volts, for each of the 6 A/D channels from the Arduino.

Here is a screen shot of a program that is performing this function, and continuously outputting the values through the Arduino's serial port and displaying them. I connected the LM335 temperature sensor (as previously discussed) to the second A/D channel (channel 1). If a channel is left floating, the value that the A/D reads is pretty much random.

As you can see, channel 1 is reading about 3.059 volts. If that

see **Program**, page 11



```
   Serial.print(i);
   Serial.print("=");
   Serial.print(readings[i], 3);
   Serial.print(" ");
  }
Serial.print("\n");
}

int readAnalog(float d[])  /*read all the channels, conve
  {
   int chan;
   float rd;
   int j;

  for (chan=0; chan<NUMCHANNELS; chan++)
  {  // read it 5 times and and average
     rd = 0.0;
     for (j=0; j<5; j++)
     {
      rd = rd + ((float) REFVAL * ((float) analogRead(cha
      delay (2);
     }
  d[chan] = rd / 5.0;
  }
  delay(1000);
}
```

```
ch0=1.999 ch1=3.059 ch2=2.598 ch3=2.258 ch4=1.937 ch5=2.000
ch0=2.000 ch1=3.060 ch2=2.592 ch3=2.270 ch4=1.956 ch5=2.000
ch0=1.999 ch1=3.061 ch2=2.581 ch3=2.277 ch4=1.930 ch5=1.999
ch0=1.999 ch1=3.060 ch2=2.595 ch3=2.259 ch4=1.930 ch5=1.999
ch0=1.999 ch1=3.059 ch2=2.593 ch3=2.263 ch4=1.955 ch5=1.999
ch0=1.999 ch1=3.060 ch2=2.583 ch3=2.279 ch4=1.935 ch5=2.000
ch0=1.999 ch1=3.060 ch2=2.592 ch3=2.266 ch4=1.926 ch5=1.999
ch0=1.999 ch1=3.059 ch2=2.595 ch3=2.257 ch4=1.953 ch5=1.999
ch0=2.000 ch1=3.059 ch2=2.581 ch3=2.275 ch4=1.943 ch5=1.999
ch0=1.999 ch1=3.060 ch2=2.588 ch3=2.271 ch4=1.922 ch5=1.999
ch0=2.000 ch1=3.059 ch2=2.596 ch3=2.257 ch4=1.944 ch5=1.999
ch0=1.999 ch1=3.058 ch2=2.585 ch3=2.270 ch4=1.949 ch5=1.999
ch0=1.999 ch1=3.060 ch2=2.585 ch3=2.273 ch4=1.925 ch5=2.000
ch0=1.999 ch1=3.060 ch2=2.596 ch3=2.257 ch4=1.936 ch5=2.000
ch0=2.000 ch1=3.062 ch2=2.590 ch3=2.266 ch4=1.953 ch5=2.000
ch0=1.999 ch1=3.062 ch2=2.584 ch3=2.278 ch4=1.930 ch5=1.999
ch0=1.999 ch1=3.062 ch2=2.597 ch3=2.262 ch4=1.932 ch5=2.000
```

☑ Autoscroll

```
Done uploading.
Binary sketch size: 3872 bytes (of a 32256 byte maximum)
54
```

Program

continued from page 10

represents the LM335 voltage, then it is telling me that the temperature is 306 °K, or 33°C. The temperature at my workbench was about 29°C at the time, which is within the range of the uncalibrated LM335 sensor voltage (the LM335 includes a calibration input that you can adjust to get it right on the money).

Hopefully you can see how we can create some small building blocks (a function that reads an A/D converter channel) and put the blocks together to do some useful functions. The nice thing is that once your block is done, you don't have to mess with it again... just use it to build larger blocks.

Digital Inputs/Outputs

All of the Arduino's I/O pins (even the analog inputs) can be configured for digital input or output, using some simple commands. For example, I connected an LED (with a 220 ohm series resistor) between digital I/O pin 9 and ground. I defined pin 8 as a digital input (a momentary button switch).

```
#define LED 9        // LED on pin 9
#define BUTTON 8     // digital input
    on pin 8
```

Every Arduino program has a built-in section called setup, which is executed once whenever the Arduino starts, or is reset. Any program statements that you want to be executed once on startup goes here. Typically, serial port setup, or configuring I/O pins would be done in this section.

```
void setup()
{
 Serial.begin(9600);       // set serial
    port to 9600 bps
 analogReference(DEFAULT);     //
    set analog reference to built in
    5V
 pinMode(LED, OUTPUT);          //
    set LED pin 9 as digital output
 pinMode(BUTTON, INPUT);       //
    set button pin 8 as digital input
}
```

Once the setup is done, whenever we want to change the state of an output, or check the state of an input, two simple functions are used to do that, digitalWrite and digitalRead.

```
// flash the LED
digitalWrite(LED, HIGH);   // turn
    LED on - set output pin to HIGH
    (5 V)
delay (30);                  // wait 30
    milliseconds
digitalWrite(LED, LOW);  //turn LED
    off - set output pin to LOW (0 V)

// check digital input
  if (digitalRead(BUTTON) == LOW)
   { // input is low, do what ever task
    is needed for LOW condition }
     else
       { // input is high, do what
    ever task is needed for HIGH
    condition }
```

Keeping Track of Time

The Arduino provides some built-in functions that allow programs to keep track of elapsed time. The basic board does not have a real-time clock, but it can keep track of how much time has elapsed since it was turned on or reset Real-time clock add-ons are available, popular chips are the Maxim DS1307 or DS3231 real time clock. Since I don't have that capability (yet) I'll just use the Arduino's existing function, millis, which returns the number of milliseconds since the program was started.

```
unsigned long current;
current = millis();    // number of
    milliseconds since board started
```

The data type is "unsigned long" which is a 32-bit integer number, allowing a total count of 2 to the 32nd power, or 4294967296. If each count is 1 millisecond, then working back gives a total count of about 50 days before this counter overflows (and starts back over at zero). I can keep my own counter which can be a sub-multiple of this count, I chose 100 counts per second, which would give 10 times the range (~500 days). There's other software tricks we can do to deal with the counter overflow, but this will be fine for now.

I define another unsigned long counter which I call tick.

```
#define TICKSPERSECOND 100
    // the 'tick' count is used to
    calculate the elapsed time

// time values
 unsigned long tick=0;    // multiple
    ticks per second
 unsigned long days=0;
 unsigned int hours=0;
 unsigned int minutes=0;
 unsigned int seconds=0;
```

By periodically reading the "millis" count, and converting it into the "tick" count (basically dividing it by 10) we can keep track of the elapsed time.

```
current = millis();    // number of
    milliseconds since board started
Interval = current - last;   //
    milliseconds since last time thru
tick = tick + (Interval / (1000/
    TICKSPERSECOND));   //
    increment ticks
last = current;        // save for next
    time thru
```

Now, if I know how many hundredths of a second have passed, it's fairly easy to convert this to days,

see **Time**, page 12

Time

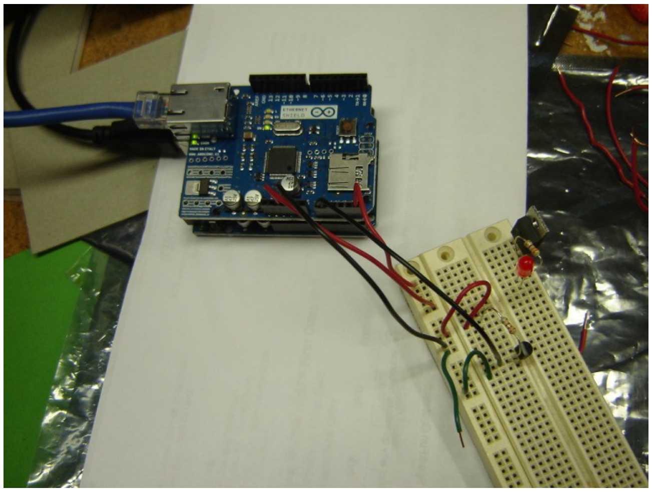
continued from page 11

hours, minutes, and seconds. Here I introduce a C operator called modulo, it is represented by the % sign, and returns the remainder of an integer division.

```
seconds=tick/TICKSPERSECOND;
days = sec /86400; // number of
    days (86400 seconds per day)
r = sec % 86400;   // % is modulo
    (i.e. remainder) operator
                // r now contains the
    number of seconds to the next
    day
hours = r / 3600;  // number of hours
    to the next day (3600 seconds
    per hour)
r = sec % 3600;   // r is now number
    of seconds to the next hour
minutes = r / 60;    // number of
    minutes to the next hour (60
    seconds per minute)
seconds = r % 60;    // number of
    seconds to the next hour
```

Arduino on the Web

The Arduino can be "Internet-enabled" by using the Arduino Ethernet Shield. This plug-in module provides an Ethernet and TCP/IP interface that can be programmed to perform as a simple web server (or any other Ethernet-enabled function). Interestingly, the Ethernet Shield actually costs more ($45) than the Arduino Uno processor.



Arduino with Ethernet Shield

Like all Arduino shields, the Ethernet shield installs on top of the Arduino board. In this photo, you can see the Ethernet shield, with the RJ45 Ethernet cable coming into it... the Uno is underneath it.

The Ethernet Shield implements a TCP/IP stack which interfaces to the user program using some basic read and print commands. Using the example programs, it was fairly easy to get a simple web server interface up and running quickly, although there isn't the type of built in functionality that you would expect with, say, Apache Web Server; the user has to connect to HTTP clients and format the HTTP server responses. However, the program is only a few thousand bytes long (compared to megabytes for a full blown web server).

One thing that I noticed is that the Ethernet Shield draws much more power than the base Arduino, thus causing a drop in the USB +5 V supply. The symptom of this is that the A/D converter was no longer accurate! By changing the reference voltage #define to 4.76V (the measured value) instead of 5.00 V, the A/D accuracy was restored. On a stand-alone application (not powered by the USB cable) this will not be an issue since the power can be supplied directly to the Arduino.

The software comes with a library of functions to allow programs access though the Ethernet port. Basically, you need to tell the board what it's MAC and IP addresses are, and initialize the Server function with what port you want to listen for. In this case, the default HTTP port 80 is used, and I set up the IP address and subnet mask to work on my local network.

```
// assign a MAC address for the
    ethernet controller.
// fill in your address here:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE,
    0xED};
// assign an IP address for the
    controller:
byte ip[] = {
  192,168,1,201 };
byte gateway[] = {
  192,168,1,1};
byte subnet[] = {
  255, 255, 255, 0 };
// Initialize the Ethernet server
    library
// with the IP address and port you
    want to use
// (port 80 is default for HTTP):
Server server(80);
// start the Ethernet connection and
    the server:
  Ethernet.begin(mac, ip);
  server.begin();
```

That's pretty much it for getting the Ethernet connections up and running. Your program then needs to listen for incoming connections, and react accordingly. The code to listen for incoming HTTP requests was lifted directly from one of the Web Server examples.

I still had the temperature sensor connected to analog port 0, so I copied the earlier program to read the A/D channels, but instead of putting that data out on the serial port, I'll write it to an Ethernet client that is connected, in HTTP response format. I also added the elapsed time code that I talked about earlier. The client.print function sends the selected data to the connected TCP/IP client, in this case, the client that made

see **Summary**, page 13

Summary

continued from page 12

the connection using a web browser.

```
  // get the data
       readAnalog(readings);    /*
   read and scale all 6 channels */
      UpdateTimerTick();   // reads
   built-in millisec converts to timer
   ticks
         ConvertTicktoDHMS(tick,
   &days, &hours, &minutes,
   &seconds);

      for (i=0; i<NUMCHANNELS;
   i++)
      {
        client.print("a");
        client.print(i);
        client.print("=");
        client.print(readings[i], 3);
        client.print(" ");
    client.print("<br> Elapsed Time
   D:H:M:S  ");
       client.print(days, DEC);
       client.print(":");
    client.print(hours,
   DEC);
       client.print(":");
   client.print(minutes,
   DEC);
       client.print(":");
               client.
   print(seconds,
   DEC);
               client.
   print("<br><br>End
   of Data");
       client.print(" de
   AJ4TF");
     client.println("<br
   />");
```

As you can see, program calls the client.print function repeatedly to creating an HTTP page line by line that the client's browser will understand. The end result is when I point my browser to the IP address of the Arduino (in this case 192.168.1.201), the web page is generated by the Arduino:

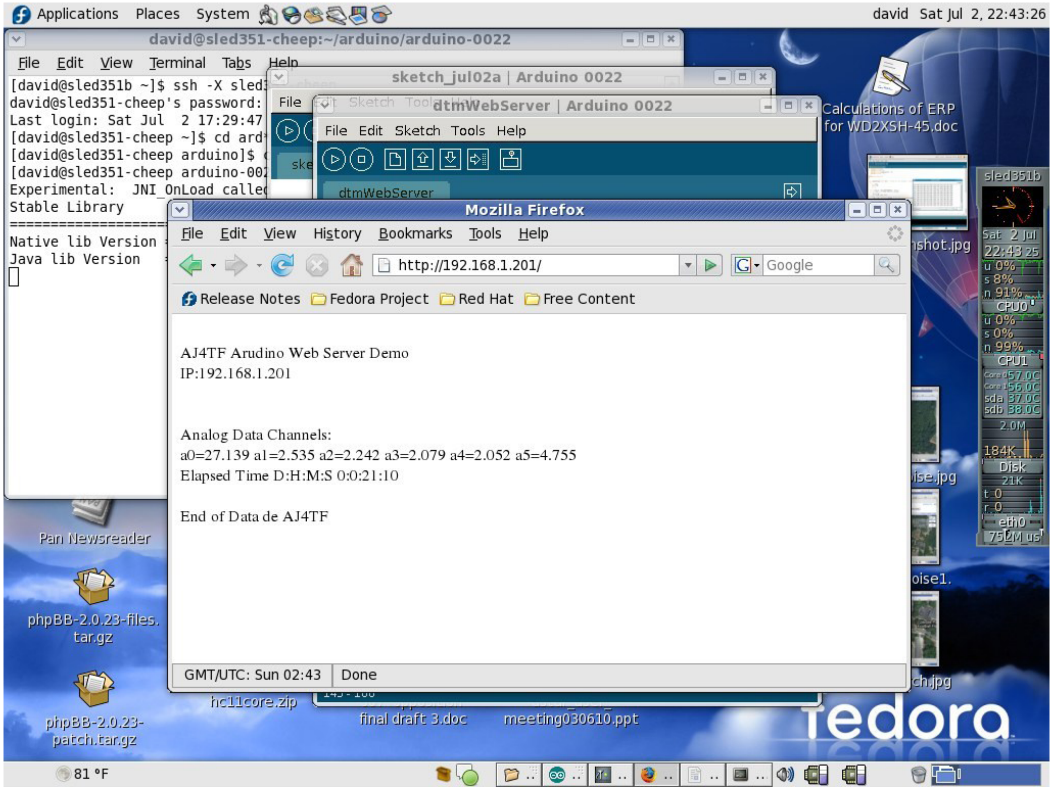
After adding all of these pieces together I now have a web-enabled embedded data measurement platform.

Summary and Further Work

The first thing I though of when I read the specs on the number of inputs on the Arduino was, "I wish there were more". There are multiplexer shields available, but I wanted to design my own input conditioning circuit. Although it's not quite completed yet, I have designed an input conditioning board that will multplex two sets of analog inputs (doubling the number of input channels) using two 74HC4053 analog switches. I have also included input low pass filtering (to reject RF) and the ability to provide resistive dividers for scaling. Also on the board are included the calibration circuitry needed to calibrate a number of LM335 temperature sensors. I have this board mostly laid out using the ExpressPCB software, but I have not yet pulled the trigger to order it or the parts needed to assemble it yet.

My experience so far with the Arduino product line has been extremely positive. The designers have gone out of their way to make it easy to use and simple. OK, so the software libraries take care of a lot of the details for you. This is good and bad... bad if you really want to know how the Atmega 328 works down to the bit/register level, but good if you want a simple autonomous embedded computer and you aren't interested in writing your own TCP/IP networking code. For purists, the Arduino designers have included the Atmel in-system-programming connection, so you could use Atmel software and debuggers if you chose to.

If you're looking for a simple embedded computer to control your next ham project, there are many choices out there, but don't forget to take a look at the Arduino.



Applications Places System
david Sat Jul 2, 22:43:26
david@sled351-cheep:~/arduino/arduino-0022
File Edit View Terminal Tabs Help
sketch_jul02a | Arduino 0022
dtmWebServer | Arduino 0022
File Edit Sketch Tools Help
dtmWebServer
Mozilla Firefox
File Edit View History Bookmarks Tools Help
Release Notes
Fedora Project
Red Hat
Free Content
Google
Calculations of ERP for WD2XSH-45.doc
sled351b
Pan Newsreader
phpBB-2.0.23-files.tar.gz
phpBB-2.0.23-patch.tar.gz
hc11core.zip
final draft 3.doc
meeting030610.ppt
fedora
81 °F

```
[david@sled351b ~]$ ssh -X sled3
david@sled351-cheep's password:
Last login: Sat Jul  2 17:29:47
[david@sled351-cheep ~]$ cd ard
[david@sled351-cheep arduino]$ 
[david@sled351-cheep arduino-002
Experimental:  JNI_OnLoad called
Stable Library
==================
Native lib Version
Java lib Version
```

http://192.168.1.201/

AJ4TF Arudino Web Server Demo
IP:192.168.1.201

Analog Data Channels:
a0=27.139 a1=2.535 a2=2.242 a3=2.079 a4=2.052 a5=4.755
Elapsed Time D:H:M:S 0:0:21:10

End of Data de AJ4TF

GMT/UTC: Sun 02:43 Done

Area Happenings

FOURTH MONDAY – at 6:30pm, the Greensboro Amateur Radio Association have their regular monthly meeting at the Golden Corral, 4404 Landview Dr, Greensboro, NC 27407, off Wendover Ave, near Sam's Club. Please plan to gather at 6:30pm for dinner. The meeting is scheduled to start at 7:15pm

CLUB NETS:

SUNDAYS – at 7:30pm, the **Technical Forum** on the 145.150;

- at 9pm, the **GARA News and Information Net.** This net features *NewsLine* and is on the 145.150, W4GSO repeater. Roy Smith, N4BYU is always looking for net controls. Contact him if you would like to help.

TUESDAYS – at 8:30pm **The Triad SkyWarn Net** meets on the 147.225, K4ITL repeater, no tone required.

TUESDAYS – at 9pm, the **D-STAR Net** meets on 442.8625 (W4GSO B and Reflector 17C)

WEDNESDAYS – at 8:30pm, **The Guilford Amateur Society Rag Chew Net** holds their weekly net on the 145.250, W4GG repeater with an 88.5 Hz. tone. Jim Hightower, W4JLH is the net control.

THURSDAYS – at 9pm, The **Guilford County ARES Net** meets on the 145.150 repeater (100 Hz. tone).

OTHER ACTIVITIES :

FIRST MONDAY – The Guilford County A.R.E.S. monthly meeting is held at 1002 Meadowood St. off W. Wendover Ave, in the EMS building, beginning at 7pm.

THIRD MONDAY – at 6:15pm **The Guilford Amateur Society** holds their monthly meeting at Tex & Shirley's Restaurant in Friendly Shopping Center. Eat at 6:15pm and the business meeting begins at 7pm.

THURSDAY – at 11:15am, Greensboro Hams get together for lunch. Thursday lunch group is meeting at the K&W Cafeteria, 300 Forum VI Mall at Friendly Shopping Center. Talk-in is on the 145. 150, W4GSO repeater with a 100 Hz. tone.

EVERY FRIDAY – at 8pm (approximately) Greensboro Hams get together for coffee at Starbucks at Edney Ridge Rd.

The W4VEC Testing Schedule

Sept. to Dec. 2011

September 10, 2011
October 8, 2011
November 12, 2011
December 10, 2011

Location: #3 Centerview Dr,
Hickory Building,
Greensboro, NC 27407
Time: 9:00am
Contact: Glenda Nicholson
Phone: (336) 674-3810
E-mail: ag4nc@bellsouth.net

GARA REPEATERS

145.150 MHz - offset, 100 Hz. tone

442.8625 MHz. + offset, Digital D-Star

Greensboro Amateur Radio Association
P.O. Box 7054
Greensboro, NC 27417
www.w4gso.org

FIRST CLASS MAIL

The Official Publication of GARA