



THE WCF EXPERIMENTER

EDITOR: GEOFF HAINES, N1GY

N1GY@ARRL.NET

Published Quarterly

Fall Edition

FROM THE EDITOR:

This is the Fall Edition of the WCF Experimenter and as I write this it is early in August. Our Section Manager, Darrell Davis KT4WX has been kind enough to submit his column on microcontrollers for this issue and we hope it will be a regular feature in the WCF Experimenter. I am still looking for articles of any size, on any subject within Amateur Radio. Please, if you have built or designed anything for your radio room or perhaps for a mobile installation, consider writing me at n1gy@arrl.net and tell me about it. I can help the ham unfamiliar with writing for publication and we can get your article into great shape in very short order. This issue, I have had to revisit some of my older articles simply because no one other than Darrell Davis, KT4WX, our Section Manager has submitted any material. I know that there is plenty of material out there from hams living in the WCF section. There has to be, I am sure that not everyone is an “appliance operator” and many of you must have built and or designed devices to improve your operating experience. I beg you to email me and describe those projects so together we can flesh out the info into an article for The WCF Experimenter or maybe even, dare I say it, QST. Believe me, there are plenty of articles in QST that came from a ham who didn't think it was worthy of publication.

BITS AND BYTES – All About Microcontrollers

By Darrell Davis KT4WX

ARRL WCF Section Manager and ARRL Technical Specialist

Welcome to the next installment of Bits and Bytes, our microcontroller column for THE WCF EXPERIMENTER. Now instead of editing the whole newsletter, I get to write my column in each issue (although I produce THE WCF PRESSER now, which is a general newsletter for the West Central Florida Section each month, so I just traded one newsletter for another). However being Section Manager has kept me fairly busy and the only regret I have, is I do not get as much project and workbench time as I used to have but that is how the cookie crumbles.

MICROCONTROLLER TERMS: The last time I dealt with RISC versus CISC. Let us get into a few more terms that will be unfamiliar to newcomers.

- **RAM: Random Access Memory**. This is the memory used by the program that is running on the microcontroller to store information that is being acted upon by the microcontroller program. This memory is what is called volatile memory and it only retains the data stored in it while the power is applied. Turn the power off and the contents in RAM are lost. Also another key fact to remember is that the main program that runs the microcontroller is not typically stored in RAM. Only the data the main program acts upon in some fashion is stored in RAM. Also another key of this type of memory is any portion of it is accessible in any sequence. It does not have to be read from beginning to end, which is a great time saver. There is an excellent Wikipedia entry on Random Access Memory at http://en.wikipedia.org/wiki/Random-access_memory.

- **ROM: Read Only Memory**: This is the memory used to store the main program that runs the microcontroller. This memory once programmed is either usually not changeable or readily changeable in the field once programmed. Originally in days gone by this type of memory was programmed onto integrated circuits by the manufacturer from source code provided by the end user and was not changeable in the field. Today most manufacturers use some form of EEPROM, which stands for Electrically Erasable Programmable Read Only Memory. They are programmed by the application of a programming voltage when being programmed and is not ordinarily used in normal operation. Once programmed the EEPROM will act like a ROM chip. There is an excellent Wikipedia article on Read Only Memory at http://en.wikipedia.org/wiki/Read-only_memory.

- **Clock Rate**: This term refers to how fast, in MHz usually, how fast the clock that runs the microcontroller operates. The faster the clock, the faster the microcontroller can fetch and execute instructions. There is an excellent Wikipedia article on Clock Rate at http://en.wikipedia.org/wiki/Clock_rate. Unlike desktop computers today, most microcontrollers have a clock speed of 4-24 MHz on 8 bit processors. Mid range microcontrollers (ARM Cortex M0-M3) operate in the 25-75 MHz range. High end microcontroller clock speeds can be in 100-300 MHz range. Remember, microcontrollers do not need high clock speed like a desktop computer to perform the task they are designed for very well.

The clock in a microcontroller is what sequences the microcontroller in its fetch and execute cycle. It is analogous to the timing belt or timing chain in an automobile. Without it a car cannot run and without a clock a microcontroller cannot run either. There would be nothing to sequence events to happen and the microcontroller would just sit there, use power, and do nothing useful.

In days gone by microcontrollers required a crystal or a RC network (Resistor-Capacitor) network to provide the clock, or timebase, to operate the microcontroller.

Today, many microcontrollers have their own internal clock or have the option in most cases to use the internal clock or to use an external clock (with a crystal or an RC network). This was designed by the manufacturers to reduce parts count and save cost. Only in more critical applications, is an external clock necessary.

To sum it all up: In the microcontroller world, ROM (or in our case EEPROM) is used as program memory or OS memory, RAM is used to hold the data the program is working with and in many cases I/O Ports are treated just like RAM when programming the microcontroller, and the clock speed is how fast the microcontroller clock is operating.

Below are a couple of diagrams to illustrate again that a microcontroller has all of these in one chip versus having them on several chips, a great advantage:

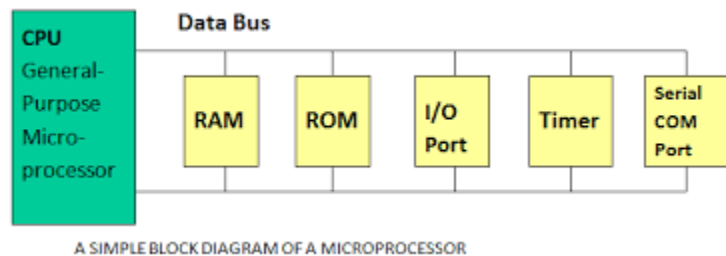
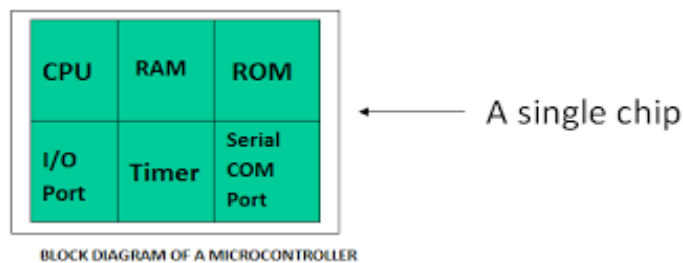


Figure 1 – A microcontroller versus a microprocessor block diagram. Courtesy of <http://www.learningembedded.com>

FEATURED MICROCONTROLLER - PICAXE: We will introduce a new microcontroller family each time we have Bits and Bytes. This month we will introduce the PICAXE. What is a PICAXE you say? The PICAXE is a family of microcontrollers that are based upon the Microchip series of 8 bit PIC Microcontroller. The PICAXE is a BASIC interpreter already programmed onto a PIC Microcontroller that looks to execute a program upon power being applied.

When you write a program for the PICAXE, you would use the PICAXE Programming Editor, available for download from the PICAXE website at <http://www.picaxe.com>, to write, debug (find errors or bugs), and test your BASIC program. The PICAXE Programming Editor is an IDE (Integrated Development Environment) developed by Revolution Education Ltd., the developers of the PICAXE environment, to do everything for PICAXE development in one application. The PICAXE Programming Editor has a nice simulator built in as well, for simulating programs running before you ever load the program onto a PICAXE. This package is for the Windows Operating System, but there are packages as well there for download for Linux and Mac platforms.

Once you have initially debugged your program you use the PICAXE Programming Editor, with a simple interface circuit, to write your program to PICAXE Microcontroller. Then built the circuit on a breadboard to test your circuit and program. Then you will go through no doubt numerous program writing, debugging, load, and testing cycle before you get your program just like you wanted.

If you cannot wait until my next column, you can view my "Introduction to PICAXE" PowerPoint slide presentation from the ARRL West Central Florida Section website at http://arrlwc.org/download/wcftechconference_2015/IntroductionToThePICAXE.odp. It will give you an Introduction to the PICAXE, how to get started, a sample program, and resources on how to get you started.

In my next column we will continue talking about the PICAXE, which is a great place for beginners with microcontrollers to get started, some more terminology we will define to increase your skills if needed, and any breaking news in the microcontroller world you need to be aware of. That is all I have for you in this installment. Have a great summer, stay cool if you can, and keep your soldering iron hot until next time. 73!

A Different Take on The Go-Box

Geoff Haines, N1GY

A few days ago, I purchased a new radio from an old friend, John Bee, N1GNV of Quicksilver Radio Products. The new unit was the JT-270M from Jetstream. A little research showed this to be a rebranded radio made in China by a company called Leixan (I think) and called the VV-898. The radio is a dual band unit that puts out only 10 watts on high power and 4 watts on low. In this day and age, that may not sound like much but it is priced at less than one third of what dual banders cost from the big three. When I fired it up and got it programmed, I could hit all of my normal local repeaters full quieting. I was not really surprised by that since I can hit most of them with my 5 watt HT. Still double the power of an HT in a package that is in some cases smaller than some older HTs, is nothing to sneeze at.

Since I recently had to purchase a so-called "mobility scooter" because of medical problems, I thought that the unit would work well on that "vehicle". The scooter

manufacturer warns against any two way radio transmissions near the scooter because according to them "RFI might interfere with the proper operation of the scooter". I did a short test anyway under what I considered adequate safe conditions and found that the RF from the radio did nothing to the scooter. However, the RFI hash from the scooter wiped out any possibility of hearing any received signals from another radio. I will bet that the cost of eliminating the RFI from the scooter played a major part in their warning against two way radios on or near the scooter. In any case, I do not plan to operate while in motion anyway so as long as the scooter is not moving I have no problem with radio operations.

The next phase of the project was where and how to mount the radio and antenna. The scooter came with a standard size basket that clips to the front of the scooter. Made of heavy duty plastic resin, the basket was easily able to hold the radio, the antenna system and a 7 AH gel cell 12 volt battery and it's associated charger. I was also able to fit the radio manual and still have room for plenty of other bits and pieces like forms, log book, rain suit etc.

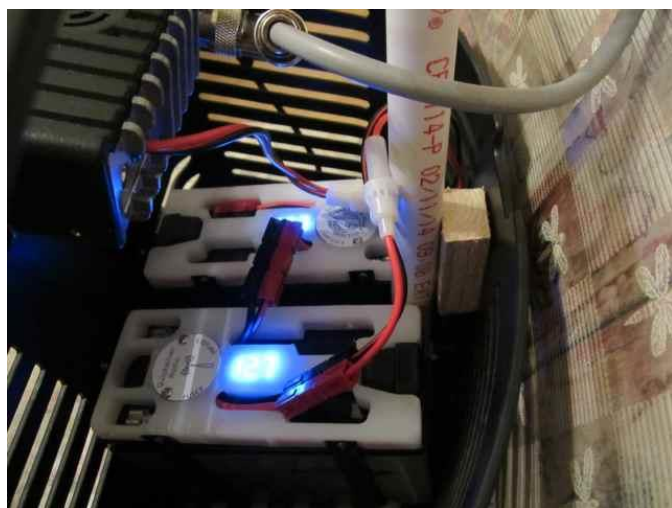
The design of the antenna went through several iterations. My first attempt was a small "Hershey kiss" type mag mount dual band antenna placed on a metal device that added some radials and made of steel was a good mount for the antenna. This worked well for the 70 centimeter band but was a no-go on 2 meters. Then I tried a vertical dipole using two dual band whips made by OPEK. The results were worse but I soon figured out my error. I had not built the antenna with enough separation between the lower element and the coax to the radio. My third attempt went much better. I used some PVC tubing (1/2" ID) to place the antenna elements some 12' ahead of the place where the coax came down to the radio. The results were very acceptable. On 2 meters the SWR was less than 1.7 to 1 across the entire band, getting as low as 1.2 to 1 around 146 MHz. On 70 centimeters the story was much the same. The SWR was well within acceptable limits (1.6 to 1 or less) across the part of the band that I planned on using.

I used several methods to mount the various components of the new Go-Kit radio system. The radio itself, mounted on the supplied bracket, was secured to the back wall of the basket with zip ties. This may sound inadequate, however, the entire radio weighs around 11 ounces. I used the supplied mic clip on the outer left side of the basket to make the mic easily accessible. The batteries and charger sit in the bottom of the basket and the antenna system is screwed to the front of the basket. I used 1/2" PVC pipe to construct an 18" vertical mast topped by a 13" horizontal strut that terminates with the upper and lower elements of the dipole. Coax is run through the horizontal strut and fitted with a ring terminal on both the shield and the center conductor. Each ring terminal is secured with a nut to one of the antenna elements. The same nut also secures the element to a flat panel of ABS plastic sheet about 1 1/2" wide and 5" long. The ABS is screwed to the top of the horizontal strut. For storage, a "Velcro" tie strip was placed around the vertical mast and can be wrapped around the strut when it removed from the top of the mast. Because of where this Go-Kit is stored in my vehicle, I did not feel it necessary to disassemble the antenna further.

Using this assemblage in my driveway, away from the house, I was able to hit all of the local repeaters and simplex was also full quieting. I consider this more than adequate for our local ARES operations since our team has tested the area and found that a 5 watt hand held is more than adequate for operations in our area. That is not to say that all operators could hit everything with 5 watts, but all could hit enough either simplex or repeater to get their messages transmitted to the EOC. The extra 5 watts in my set up just adds a little extra power.



This photo shows the radio installation in the basket, but the antenna shown did not work well and was replaced by a better set up that works fine.



This photo shows the dual battery set up. The white panels on top of each 7 AH Gel-cell

contain two Power Pole leads, one of which has a 5 amp blade fuse in line. The other lead is unfused and is used in this project to place the two batteries in parallel and create a 14 AH source. These unfused leads are connected via a Power Pole “y” so as to permit recharging both batteries with one charger. These battery toppers also have a switchable voltmeter built in so one can check the condition of the battery with the flick of a rocker switch

The new antenna pictured below used two dual band stingers assembled as a vertical dual band dipole. Shown on an interim mount here, the antenna was eventually mounted on a vertical PVC mast that plugs into a socket at the rear of my mobility scooter and gives excellent service in that position.



The basket in which the radio and power are mounted is one of two baskets I have for the scooter. The other, stock basket is used when I do not need to have a radio with me like shopping or attending an outdoor craft fair or similar event. This antenna has been superseded by mounting the antenna on top of a telescopic painters pole mounted at the back of the scooter behind the seat. You can view the new antenna on my web site

at www.n1gy.com

While my setup is designed around my need for a mobility scooter, anyone needing a portable setup with twice the power of the usual HT can easily create a package that works for them using one or two batteries and the JT-270M. The radio has 199 memories available and is quite easy to program with the included software and cable. One caveat, the Jetstream software does not work with Windows 8.1, only with Windows 7 or previous. On the other hand, the RT Systems software for the JT-270M does work fine with 8.1 so that is another option.

If anyone has any questions about the JT-270M they can email me at n1gy@arrl.net and I will be happy to provide any info I have.