

EDITOR: GEOFF HAINES, N1GY

N1GY@ARRL.NET

Published Quarterly

Fall Edition

From The Editor: Summer is almost over, although in Florida it never really goes anywhere. I need more articles about home-brew projects so we can encourage DIY Amateur Radio in West Central Florida. My email address is right there on the masthead so take keyboard in hand and start putting it down. Doesn't matter what "it" is, I can sort that out. That is my job as an editor.

A home-brewed Fume Extractor For The Workbench

Geoff Haines, N1GY

The other day, I was listening to a couple of local hams on the local repeater when the subject of the fumes from the job of soldering came up. The consensus was that for the occasional soldering job there was not much risk but that if one were soldering on a regular and frequent basis, certain precautions were wise to take. This started me to thinking about my own workbench habits. Since I am now building Power-pole power distribution blocks on a regular basis for sale, my exposure to the toxic elements of the smoke and flux coming off of the soldering tools and projects has risen significantly. I immediately began an on-line search for soldering fume filters etc.

While I found sources for bench-top fume hoods immediately, the prices were reason enough to pursue a more home-brewed solution. I also remembered that my son had given me a batch of computer fans several years ago. I checked and sure enough I had two identical 3.5" fans that operated on 12 volts DC nominally. I wired them in parallel and using my bench power supply fired them up. They both ran beautifully and very quietly as well.

Looking around my radio room, I found a cardboard box of the right shape, size and rigidity, it turned out to be the box my antenna analyzer came in. Using a compass and a utility knife to draw and then cut two circular openings in the front of the box, I then mounted the fans to the inside of the front using 4-40 1/2" bolts and nuts. Using the utility knife again, a rectangular opening was cut in the back of the box in preparation for an activated charcoal filter panel. The filter material is available very inexpensively from several on-line sources.

I wired an on/off switch to control the fans and added a short run of red/black zip cord in 16 gauge to connect to the bench supply via Power-Pole connectors. A shopping trip to the local craft store resulted in a panel of what the retailer calls "plastic canvas". This is a 1/8" mesh plastic material which made a perfect front fascia for the fume extractor to keep errant fingers away from the fan blades. I wrapped the entire box in duct

tape to strengthen the cardboard, of course leaving the front and back clear. After painting the front of the box black, a piece of the "plastic canvas" was cut to fit the front of the box and secured in place on all four sides with more duct tape. As I write this, I am awaiting the arrival of the filter material from Amazon, so as soon as it arrives and is installed I will add a photo of the filter on the back of the enclosure. Other than the filter material and the plastic canvas not one penny had to be spent on this project. The cost of a commercially available fume extractor of similar size and functionality would have cost over \$50.00 even from Amazon, considerably more from some other retailers. I think I spent less than \$5.00 for the whole project, even accounting for the parts that I already had on hand.



. The photo to the left shows the front of the fume extractor. The "plastic canvas" can be seen covering the fans to prevent errant fingers from getting into the fan blades. I used screws to attach the mesh as the duct tape kept coming loose of the plastic material.



The photo to the left shows the back of the extractor with the addition of the activated charcoal filter. Without the filter the fans are just moving the fumes around the room, not removing them from my environment. I still have lots of filter material to replace this piece on a regular basis

As I mentioned above, the filter material arrived today, on a Sunday no less, and it was soon installed on the back of the fume extractor. I decided to test the unit with all of the changes that have been made over the past few days. I fired up my new 200 watt soldering iron and let it heat up until the tip began to smoke in the usual manner. The smoke was speedily sucked through the extractor and none of it managed to get through the activated charcoal filter in the back. UPDATE: It is now Tuesday afternoon and the power supply (really more of a power brick) arrived from Amazon. I replaced the coaxial power connector

with a Powerpole set and the fume extractor is now officially finished. I did ponder the possibility of replacing the cardboard box with one assembled using ABS 1/8" sheet plastic but decided to leave that for another time when the cardboard has worn out. JOB DONE!

WELL, NOT QUITE! After using the new fume extractor for a few days, I realized that the upper fan does very little of the work because it is above the point at which the fumes are generated. The bottom fan does most of the work. A redesign was in order. Same two fans, same wiring but now the two fans are in tandem, one behind the other. The filter material and the screen in the front were re-sized and in fact they were bolted or otherwise secured to the fan bodies themselves. The fans were bolted together using the same hardware that used to hold them to the cardboard box. When it was done and tested, the wiring looked a tad exposed so I added three ABS plastic panels to the top and right and left sides to neaten up the exterior. The results are below.



The on-off switch and the LED were both moved to the front of the extractor. The slight OOPS next to the switch was the result of an errant drill (certainly not the operator's fault!!!!) The ABS panels on the sides were cut back at the rear corners to allow room to replace the filter material as necessary. The entire unit is much shorter and smaller overall, allowing me a better view of the tools stored behind it on the tool rack wall. The air flow is also improved by having both fans pulling on the same column of air.

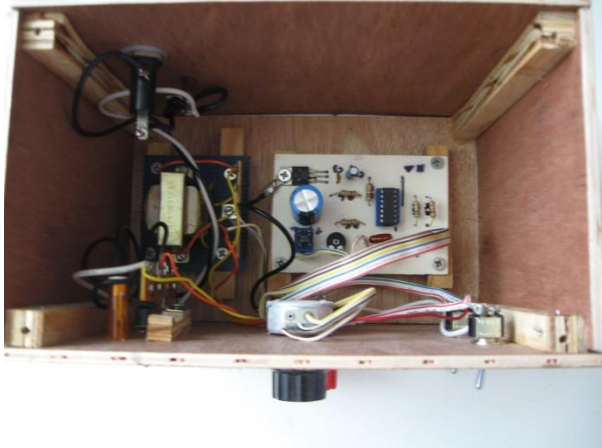
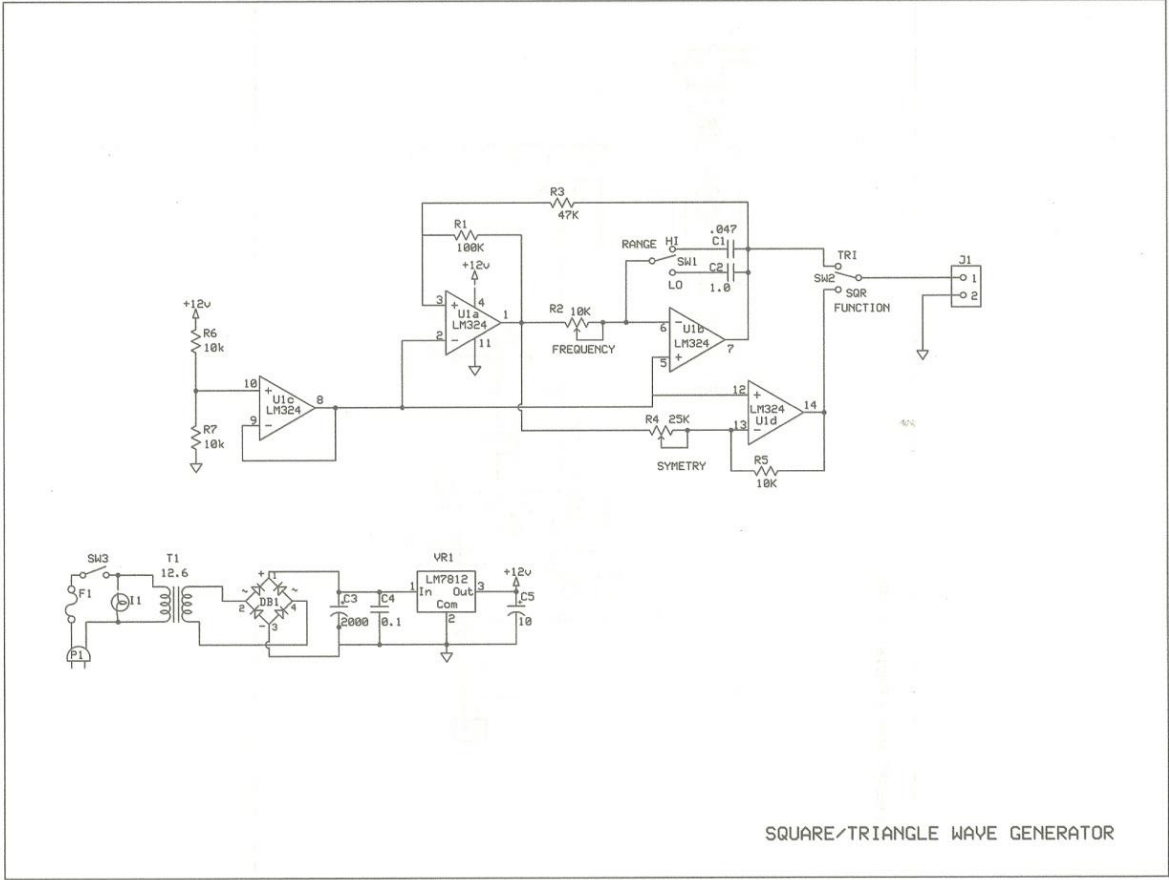
The point of this project is that if you spend even a fair amount of time soldering at the bench you are inhaling a bunch of toxic gases and fumes given off by the solder, the flux and even the tip of the soldering iron itself. Better to suck those fumes away from you and through an activated charcoal filter and keep your lungs cleaner as a result. If anyone would like to duplicate this project, I have a massive amount of the activated charcoal filter material on hand and will be happy to provide it free of charge to anyone who asks for some. Just send me an email (see the masthead) and let me know what size piece you need.

73

A Square/Triangle Wave Generator by KG4BZW

I had a need for a square/triangle wave generator to check on some audio projects. One way to accomplish this relatively inexpensively is to generate a square wave and then apply it to an integrator thus producing a triangle wave. There are several ways to generate a square wave such as overdriven amplifiers, using the ubiquitous NE555 timer, etc. By using two OP AMPS, you get both a square wave and a triangle wave from the same circuit. The OP AMP I chose to use is an LM324 quad comparator. The way a comparator works is that the output is determined by comparing the inverting input (-) and the non-inverting input(+) then either saturates to the positive rail (+ Vcc) or the negative rail (- Vcc) depending on the difference between the inputs. Generally, one input is a reference input and the other is a signal input. Referring to the schematic diagram on Page 5, U1a is configured as a closed-loop Schmitt Trigger comparator. This gives a good stable and controlled rectangular wave output. The non-inverting input, U1a pin 3, is the signal input and the inverting input, U1a pin 2, is the reference input. U1b is configured as an integrator producing the triangle wave. A portion of the triangle wave is fed back to U1a pin 3 as its input signal. The frequency of the integrator is determined by the RC time of R2 and C1 or C2. The square wave and the triangle Peak to Peak (P-P) outputs are not of equal amplitude, the triangle wave being less than the square wave. U1d acts as an attenuator to allow adjustment amplitude of the square wave output. Since the phase relationship of the two outputs are 180 degrees, U1d also inverts the square wave to restore the phase relationship. U1c is configured to provide virtual ground for the generator and a reference input for the comparator. The pics below the schematic diagram show the inside view and the front panel of the generator.

Since the frequency of the generator is in the audio range, I didn't have to worry about RFI so I made a wooden box from 1/4" Luan paneling, screwing all sides with small wood screws (thus allowing easy disassembling for repair), to house the generator and mounted the components in it (see photo of inside view). The pointer is fashioned from a piece of aluminum. The dial was calibrated by observing the square wave form on an oscilloscope and noting the frequencies. I then laid out the front panel and used my word processor to construct the labeling. After laminating one side, I glued it to the front panel. If one desires, an additional amplifier can be added for the output between S2 and J1. I found the output (approximately 3.5 V P-P) sufficient for my purposes. 73's and happy constructing



Introduction To The PICAXE

By Darrell Davis KT4WX - ARRL Technical Specialist
6350 Mills Road, Fort Meade, FL 33841-9584
Email: kt4wx@arrl.net Phone (863) 245-9923

ABSTRACT

The purpose of this paper is to introduce the amateur radio operator to designing microcontroller based projects with the PICAXE microcontroller. Being the PICAXE microcontroller is programmed in BASIC programming language, the ease of learning to use the device for the newcomer as well as the radio amateur who has programmed in BASIC programming in times past, is much easier. This paper will explain what PICAXE microcontrollers are and how to get started in using them in project designs.

Keywords: PIC, PICAXE, Microcontroller, BASIC, Programming.

I. INTRODUCTION TO THE PICAXE

History of the PICAXE Microcontroller: The PICAXE microcontroller was first introduced in 1999 by Revolution Education Ltd. (also known by the abbreviation Rev-Ed), a British company. The PICAXE microcontroller is a PIC Microcontroller pre-loaded with a BASIC interpreter developed by Rev-Ed. Since that time Rev-Ed. Has made many improvements to the BASIC interpreter and has added more PICAXE controllers to their product line as well.

Advantages of using the PICAXE versus the PIC or other microcontrollers: (1) For the amateur radio operator who is new to programming microcontrollers, the PICAXE is a great way to get started in programming with microcontrollers, because it is programmed in a customized version of BASIC and is easier for newcomers to get started with.. The PIC Microcontroller by Microchip, the AVR microcontroller by Atmel Semiconductor, or other 8 bit microcontroller lines are programmed with a derivative of the C or C++ programming language. Usually the manufacturers of these microcontrollers even supply a well designed Integrated Development Environment (IDE) program that will run on a IBM PC Compatible computer running the Windows OS. This IDE will allow the programmer to write, run simulations, debug, and program the target microcontroller, via an external programmer that is attached to the serial port or a USB of that computer. The external programmer is either supplied by the manufacturer of the microcontroller or by a third party. However many amateur radio operators are new to programming and have never learned C or C++, which is a wonderful programming language. However, C or C++ is intimidating to the newcomer and can discourage them from further attempts at building with microcontrollers.

(2) For the veteran amateur radio operator, who has used BASIC in the late 1970s through the early 1990s, this is familiar ground, and can make their start with microcontrollers much easier. Also the PICAXE is a great way to get a young person started into programming microcontrollers and once they are comfortable may wish to move up to using C or C++ in future projects.

(3) Because the PICAXE is a stand along microcontroller, it can be integrated into projects more seamlessly than other microcontroller platforms that are modular in nature. However there are breakout boards and even some modular forms of the PICAXE available for designers as well.

(4) Most PICAXE designs are available primarily in Dual In Line (DIP) packaging for those who are new to microcontrollers and are not comfortable yet with surface mount soldering or have yet acquired the skills to solder Surface Mount Technology (SMT) devices onto printed circuit boards. This also makes breadboarding new designs much easier for the newcomer as well. However all of the PICAXE line is available in SMT packaging as well for those who wish to use surface mount devices.

(5) The PICAXE BASIC Interpreter provides commands that are easy to use, to control all the underlying PIC Microcontroller general purpose I/O ports and custom ports such as the SPI and I2C serial ports, A/D converters.

(6) The PICAXE controllers are well documented in several well done manuals in PDF format.

II. PICAXE PRODUCT LINE AND BASIC SPECIFICATIONS

As of this writing, there are currently seven PICAXE based IC available for use in new designs. PICAXE microcontroller based IC has been produced in several versions. Microchip still produces the PIC Microcontroller based microcontroller itself. After fabrication and packaging (installing the IC dies into its DIP or SMT package), Microchip then programs the “bootstrap firmware” onto the PIC Microcontroller and then it is shipped out to Rev-Ed., the end customer. Then Rev-Ed ships the PICAXE controller to distributors and makes them for sale directly to customers via their website. All of the microcontroller manufacturers for large production runs from a single customer will offer this service or pre-loading the customer written program onto the target microcontroller after production before the microcontroller order(s) are shipped to the customer for incorporation into their end product. For smaller quantities, the manufacturers will sell them directly to distributors and the distributors sell to small or mid quantity customers.

Earlier versions of the PICAXE had the standard PIC microcontroller model number engraved on it. Recent and current versions of the PICAXE have the PICAXE microcontroller model number engraved on the package by Microchip when it is packaged, which for the end user leads to less confusion. If a designer were to have standard PIC

microcontrollers and PICAXE controllers on his or her workbench, there would be no confusion as to which controller is which.

The DIP products are standard 300 millimeter packages with the exception of the 40 pin version which is a standard 600 millimeter package, which is common. The SMT products are in SOIC format. The DIP product will therefore work well with common breadboards making prototyping very straightforward. The DIP and SMT are produced in 8, 14, 16, 18, 20, 28 and 40 pin packages, , which is part of the DIP industry standard.

Listed below is a chart showing the specifications of the PICAXE Microcontrollers. The chart show the program memory, scratchpad RAM size, number of I/O pins available and the clock speed of the particular microcontroller. These charts I produced from data was taken from the PICAXE Manuals.

Processor	Program Memory	I/O Pins	RAM	Clock
PICAXE 08M2	2048 Bytes (2K)	6	128 Bytes	4-32 MHz
PICAXE 14M2	2048 Bytes (2K)	12	512 Bytes	4-32 MHz
PICAXE 18M2	2049 Bytes (2K)	16	256 Bytes	4-32 MHz
PICAXE 20M2	2048 Bytes (2K)	18	512 Bytes	4-32 MHz

Figure 1:PICAXE Basic Line of Microcontrollers

Processor	Program Memory	I/O Pins	RAM	Clock
PICAXE 20X2	4096 Bytes (4K)	18	256	4-64 MHz
PICAXE 28X2	4096 Bytes (4K)	22	1280	4-64 MHz
PICAXE 40X2	4096 Bytes (4K)	33	1280	4-64 MHz

Figure 2: PICAXE Advanced Line of Microcontrollers

The devices vary for 8 pin versions with 2 Kilobytes of program memory, 128 bytes of “scratchpad” of RAM ,a 4-32 MHz clock, and 6 I/O pins, all the way to the 40 pin version with 4 Kilobytes of program memory, 1280 bytes of “scratchpad” RAM, 4 -64 MHz clock, and 33 I/O pins. The Program Memory is comparable to the hard drive space on a desktop PC and the RAM is comparable to the physical memory on a desktop PC. When selecting a PICAXE for a new project, you select the one that will best fit your needs. Some may be wondering, “That is not much memory”. Remember, you are dealing with what is called an embedded system in the electronics industry. Many embedded devices do not have a Graphical User Interface (GUI), or even have a display at all. Therefore most embedded systems do not require much memory to do a dedicated task, unlike a desktop PC, which has a lot more overhead.

All PICAXE microcontrollers, like there PIC microcontroller “lineage” are TTL level devices. They will operate on a supply voltage of 3-5 Volts quite satisfactorily. However do not exceed 5VDC on the supply or port pins or you can, and possibly will, damage the device. This is the most important consideration when using these microcontrollers.

III. PICAXE BASIC COMMANDS

Figure 3 and Figure 4 contain a list of the more commonly used BASIC commands that are available in the PICAXE Basic Interpreter. The PICAXE Manuals list all the available commands which are simply would take too long to list here. You may look at the entire list of commands also at the PICAXE website at <http://www.picaxe.com/BASIC-Commands>. Most commands are in operation/operand syntax.

COMMAND	DETAILS	COMMAND	DETAILS
Pause	Pause for a defined number of milliseconds	For/Next	Repeat a loop for number of time. Return from loop
Pauseus	Pause for a defined number of mcroseconds	If / Else	Conditionally execute program code. Alternative
Wait	Delay for a number of seconds	EndIf	End of If / Else statement
Sleep	Sleep for a period of time	Gosub / Return	Call a routine Come back from routine
Doze	Reduce pwr consumed for a short period of time	Goto	Continue program execution from label

Figure 3: PICAXE Commands – Time Delays and Variables

+COMMAND	DETAILS	COMMAND	DETAILS
High	Switch an output pin on	Pwmout	Generate a continuous pulse width modulation
Low	Switch an output pin off	Pwmduty	Set duty cycle of a pulse width modulation output
Toggle	Switch an output between on and off	Input	Set a pin as in input
Sound	Generate simple sounds	Output	Set a pin as an output
Button	Detect and debounce a pushbutton switch	If Pin	Respond to the state of an input pin

Figure 4: PICAXE Commands – Digital Input and Output Commands.

IV. PICAXE PROGRAMMING AND SETUP

PICAXE Programming Setup: The PICAXE microcontrollers are all programmed with a three pin serial connection. Serial programming is ubiquitous today with virtually all microcontrollers. In the PICAXE documentation these pins are referred to as SERIN, SEROUT, and 0V. The following is the function of those pins:

- **SERIAL IN** Program data going from the host PC to the PICAXE.
- **SERIAL OUT** Communications data going from the PICAXE back to the host PC.
- **0V** Signal Ground

Figure 5 below illustrates how to setup the PICAXE for programming. The beauty of this setup is it allow for In Circuit Serial Programming (ICSP). With ICSP, you can upload an updated or revised program without ever removing the PICAXE from the circuit. One 10K Ohm resistor must be placed across the SERIN and 0V pins. Then between the 10K resistor and the SERIN pin, a 22K Ohm resistor must be placed as well. When building any PICAXE circuit you must include two resistors. Without these resistors the PICAXE could go into program mode by accident once the PICAXE is installed into a circuit. So be sure to include these resistors even if you do not plan to make provision for ICSP. You may program a PICAXE with a standard RS232 serial port or with a USB port.

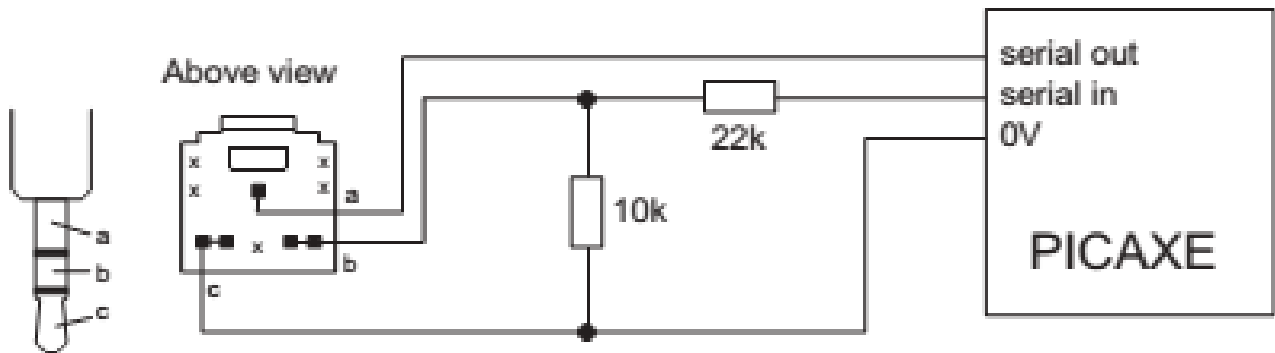


Figure 5: Serial PICAXE Configuration

Programming with an RS232 Serial Port: If you use an RS232 serial port, connection is straightforward. SERIAL IN would go to the TXD pin and SERIAL OUT would go to the RXD pin and 0V would go to Ground.

Programming with a USB Port: If you use a USB port, you will need a USB to RS232 Serial Port Adapter. The price of these adapters today is anywhere from \$5-\$20 USD. Also you will have to install the appropriate drivers in your computer to support the USB to

Serial converter. Then you connect the output of the USB to Serial converter to the PICAXE in the same manner as a regular serial port connection.

PICAXE Setup: Remember the following precautions when setting up a PICAXE project

1. Do not power the PICAXE with more than 5VDC period!!! Supplying more than 5VDC to the PICAXE can damage or destroy the PICAXE.

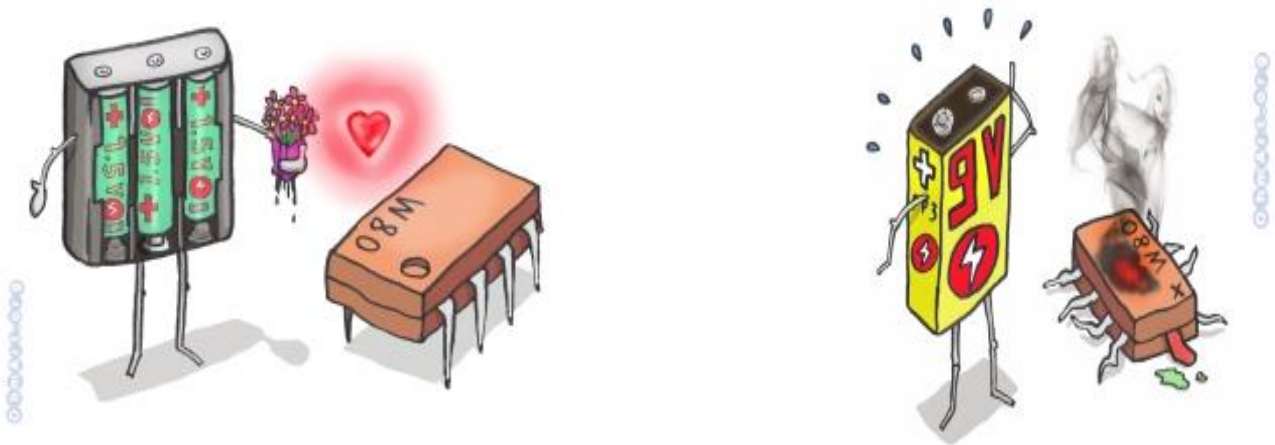


Figure 6 – A happy PICAXE running on 5VDC or less and a fried PICAXE after 9VDC is applied to it.

2. Each output can only source or sink about 20 mA of current. The maximum that the device should source or sink is 90 mA of current. If you exceed these limits you can damage the PICAXE output ports. You can drive an LED or LED segment directly without any issues but you would not be able to power a relay coil or a motor directly. If you need to control heavy current loads you must use an interface circuit that will not load the microcontroller but will source or sink the current itself.

3. Reset pin (if equipped) must be terminated to the supply voltage through a 4.7 K Ohm or a 10 K Ohm resistor. Unreliable operations will be the result if you do not pull the reset pin to the supply voltage.

PICAXE Programming Software: Rev-Ed has produced some great software for working with the PICAXE and some of it is oriented towards young people new to programming. Here is a rundown of the programming software packages as of this writing.

- **PICAXE Editor Version 6** – This is the main software for programming the PICAXE. It is a GUI based editor with a built in PICAXE Simulator. It is only available for the Windows OS.
- **Blocky for the PICAXE** – A visual block and text editor for PICAXE programming. It is available for the Windows, Linux, Mac, iOS. And Android operating systems.
- **AXEPad** – A cross platform development tool for the PICAXE. It is available for the Windows, Linux, and Mac OS.

PICAXE Programming Manuals: Rev-Ed has done an excellent job in producing user manuals for the PICAXE.

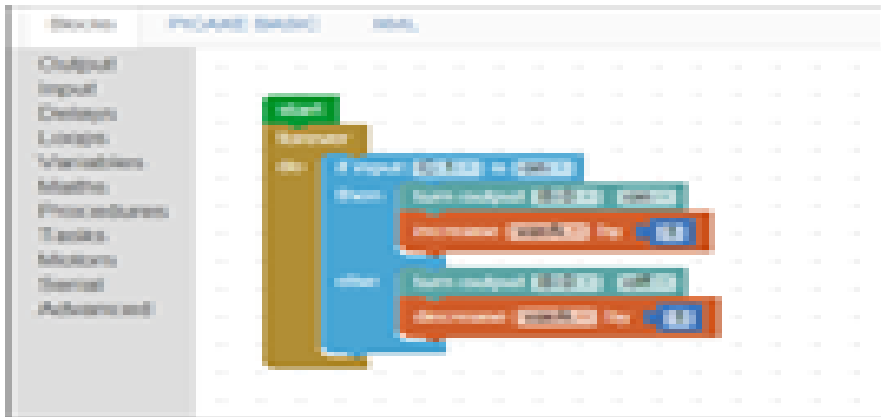
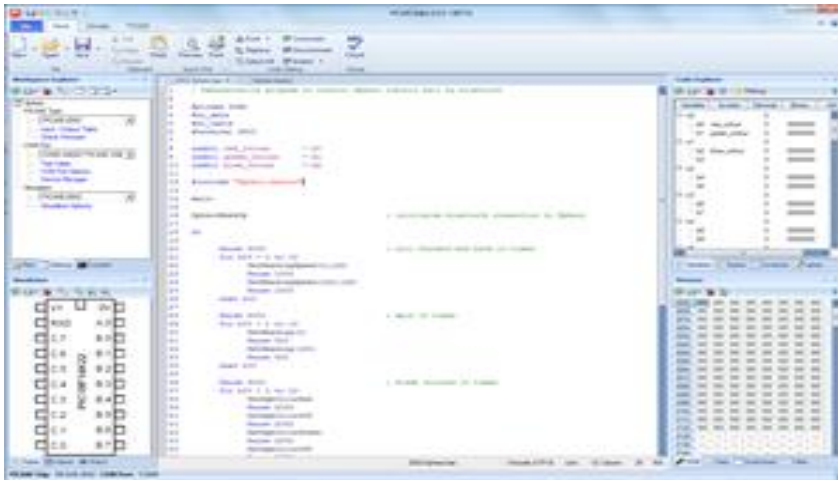


Figure 7: The PICAXE Programmer 6, Blockly for the PICAXE, and AXEPAD for the PICAXE Software Screenshots.

V. SAMPLE PICAXE PROGRAM

Listed below is the equivalent of the “Hello World” program for the PICAXE. This program is taken for the PICAXE Manuals. In computer programming, the first program that a programmer often learns is how to make the computer screen say “Hello World”. This program is a very simple program but illustrates the basic components of a PICAXE program and how easy it is to program these devices. The program is referred to as “Blinky”.

Figure 8 shows a simple circuit for a PICAXE 08M2. The port numbers referenced are pins on the PICAXE 08M2. The following simple program turns the LED on for one second and then turns the LED off for one second and then repeats endlessly until you turn the power off or reset the PICAXE.

```
main:      high 4
           pause 1000
           low 4
           pause 1000
           goto main
```

Figure 8: PICAXE BASIC Program for “Blinky”

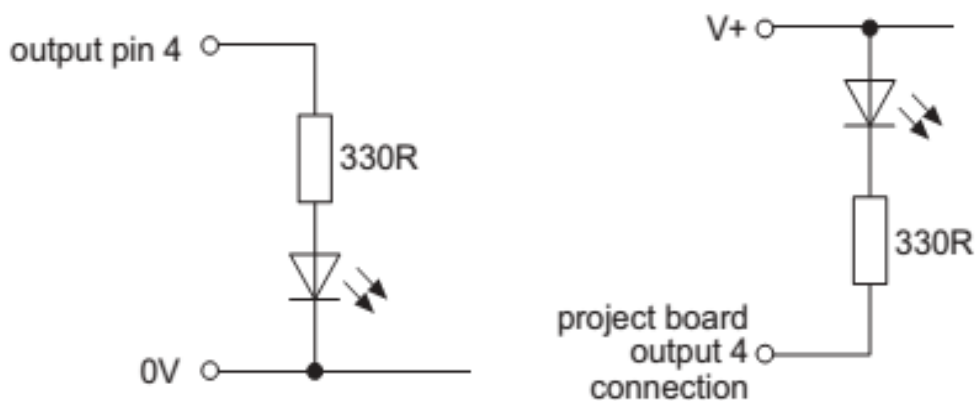


Figure 9: PICAXE Circuit options for “Blinky”
Page 16

All PICAXE programs must contain a MAIN function or routine. The MAIN Function calls upon all other functions and should continuously loop or your device will crash while running.

The MAIN functions does the following in this program: The command HIGH takes pin 4 of the PICAXE 08M2 to V+, which is the supply voltage. Then the PAUSE command pauses the PICAXE for 1000 milliseconds or 1 second. Then the command LOW takes pin 4 of the PICAXE 08M2 to 0V, which is ground. Again the PAUSE command pauses the PICAXE for 1000 milliseconds or 1 second. The GOTO commands takes the PICAXE back to the beginning and starts the MAIN program all over again.

A good microcontroller program, being it is a standalone device, will continuously run until it is turned off or the device is reset.

VI. PICAXE RESOURCES

Are you ready to dive into the world of microcontrollers with the PICAXE? The following is a list of resources to help you get started programming and using the PICAXE. If I receive good feedback from this presentation, I will endeavor to write a book titled something like, "PICAXE for Ham Radio".

BOOKS:

- Ham Radio for Arduino and PICAXE by Leigh L. Klotz Jr. WA5ZNU ISBN-13: 978-0872593244. (Ths book has some real good PICAXE projects that you may be interesting in building.)
- Programming and Customizing the PICAXE Microcontrollers by David Lincoln ISBN-13: 978-0071457651

INTERNET:

- PICAXE Website: <http://www.picaxe.com> (Well built and good material here.)
- PICAXE Programming Software: <http://www.picaxe.com/Software> (Goes over the various software packages availabe and all but one are for FREE download.)
- PICAXE Hardware Manuals: <http://www.picaxe.com/Getting-Started/PICAXE-Manuals/> (There are five PDF manuals total and they are very well done: Manual #1 - Getting Started, Manual #2 - BASIC Command list, Manual #3 - PICAXE Interfacing, PICAXE #4 – Flowcharts, and Manual #5 – Blocky. They are also for FREE download.)
- PICAXE Wikipedia Page: <http://en.wikipedia.org/wiki/PICAXE> (Basic introduction to the PICAXE microcontroller)

- **YouTube Video Tutorials:**
 - <https://www.youtube.com/user/PICAXEdotcom> (Main PICAXE YouTube Channel.) This YouTube channel was recently added.
 - <https://www.youtube.com/user/azadude18/videos> (Produced around 2010 but are good videos to help get you started)

REFERENCES

PICAXE. Revolution Education Ltd., n.d. Web. 16 Aug. 2016. <<http://www.picaxe.com>>.

"PICAXE Manual." *PICAXE*. Revolution Education Ltd., n.d. Web. 16 Aug. 2016. <<http://www.picaxe.com/Getting-Started/PICAXE-Manuals/>>.

3rd Annual TECHCON (formerly the WCF Section Technical Conference)

To be held at the Sarasota Red Cross HQ on Cantu Court in Sarasota. The date is February 25, 2017. Looking for speakers for the event.

Please contact Darrell Davis, KT4WX via the ARRLWCF web page

73 de Darrell KT4WX