

Picaxe microcontrollers 101

GQRP Rishworth Convention

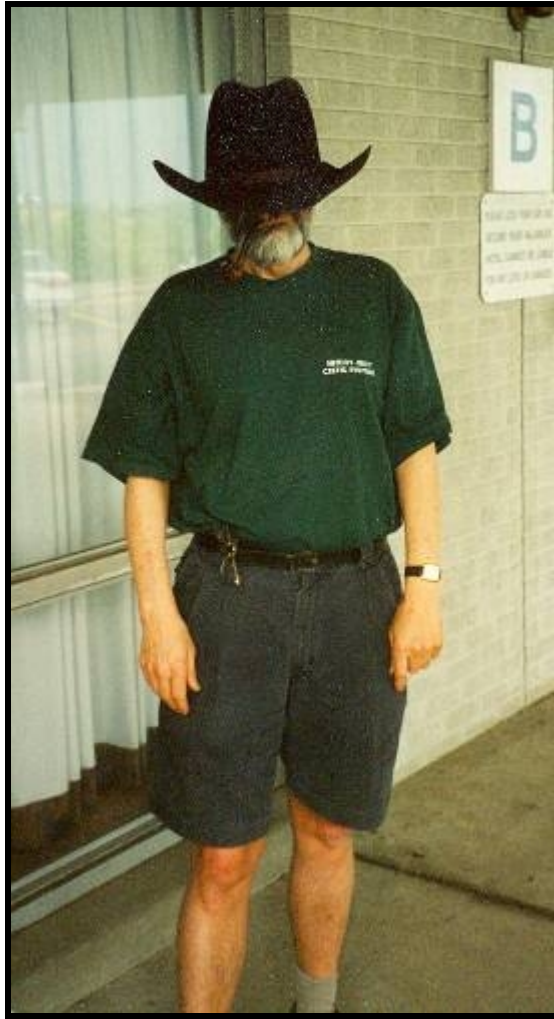
Presentation by W1REX

I once attended a seminar back in my home country where a gentleman from this country, whom shall remain nameless, introduced the audience to the old English word Pelf and how it applied to QRPers.

(here is my recollection of the main jist of the introduction as it might differ from the actual presentation as I forgot to take a steno pad and even if I did have a steno pad, I can't write in shorthand (or read it for that matter) but, even if I did have a steno book and could write and read shorthand, I would have lost the notes by now anyway!)

Pelf n. the collection of stuff that is not really needed but you collect up anyway just to have it, often applied to booty or ill gotten gains (as derived from the word pilfer)

QRP is like anti-pelf; as good QRPers are often known to go to extremes to do more...with a lot less!



The Pelfman disguises his identity using local headgear...



**Another picture of the Pelfman with his identity
concealed.....(by me)**

**Now as a whole, the entire industrialized world
runs on pelf, and pelf especially runs rampant in
America....**

**Americans have accumulated so much pelf that
an entire business exists just to perpetuate the
practice....**

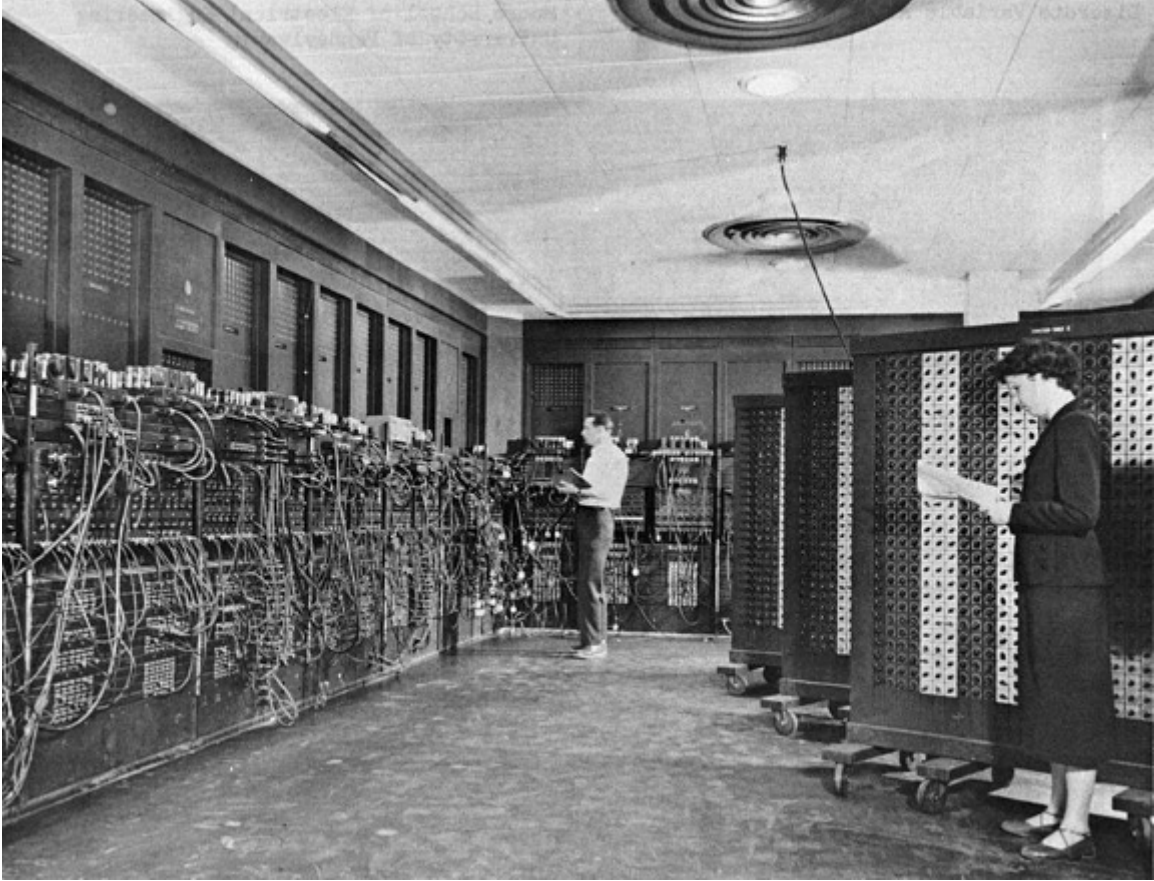


It seems that America wants to be #1 (as always) in the Pelf Race and has pulled out all the stops to remain #1. The National Rural Pelfication Act was passed in Washington D.C. to spread the practice of Pelf into the very heartland of America as exemplified by this business in my home town of Limerick Maine.....



OK! Let us get to The Picaxe microcontroller and what it can do for you....

Lets take a walk down memory lane and look at how electronic computers have evolved over the years...



1946

ENIAC

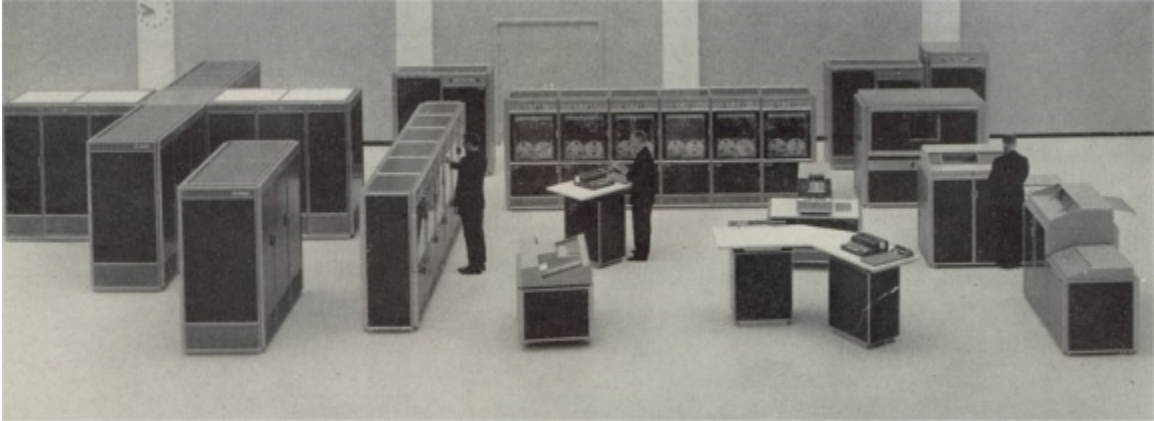
First programmable computer



1956

WHIRLWIND

First direct keyboard entry computer



1966

GE625 Timesharing Mainframe

This machine ran the original
Dartmouth College BASIC which
allowed hundreds of users to type in
and run BASIC programs
simultaneously.

I was one of the lucky ones!



1976

**typical (?) S100 bus 'home'
computer system**

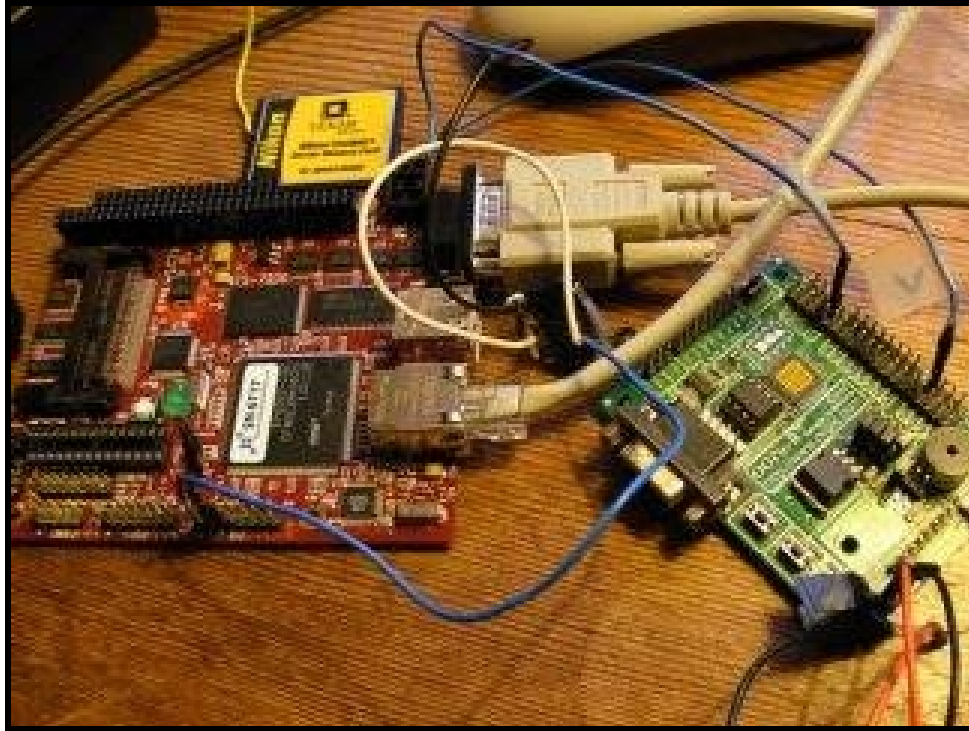
**Mine cost \$9500 and took me 6
months to build...**



1977

The ***FIRST*** 'true' Personal Computer?

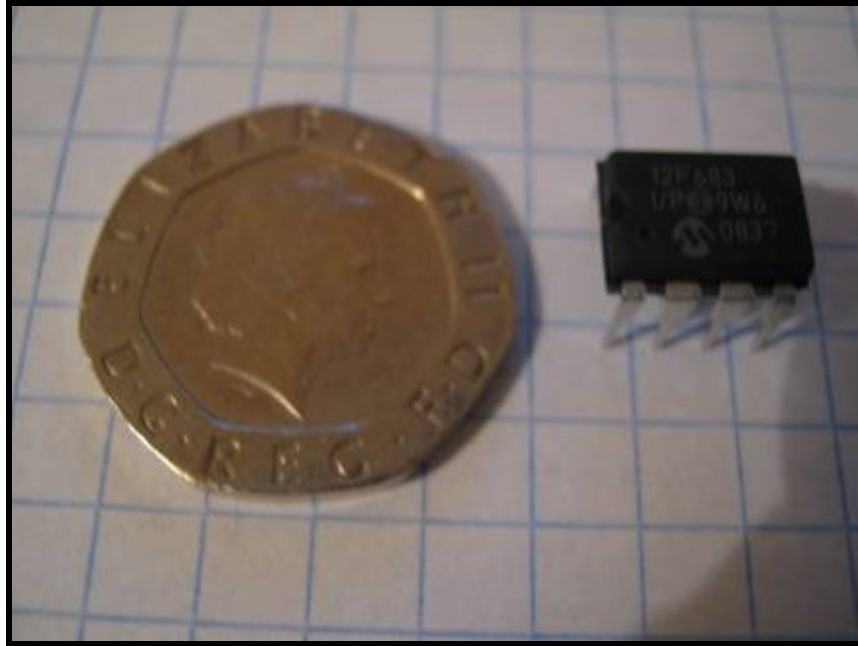
Apple II with 4K memory, cassette tape storage system and video output to monitor or color TV = \$3000.00



1997

Single Board Computers

Dedicated processing in a box
\$400.00?



2009

World's simplest computer in a chip
Picaxe 08M

and
of special interest to QRPer...

It is **CHEAP!**

@ \$4.00 each!

What is a Picaxe?

The Picaxe 08-M is an electronic microcontroller chip where a user can change the functionality of the chip by sending it a sequence of commands.

Picaxe 08-M Resources

Program lines:	80
RAM memory for variables:	14 bytes
Bit variables:	16
Byte variables:	14
Word variables:	7
Storage variables:	48 bytes
I/O lines:	5
Total available memory	256 bytes
Integer only math	

Did I mention that the Picaxe microcontroller is CHEAP?

So what does that mean for you the skeptical and VERY frugal QRPer?

80 lines of program code means that the Picaxe can hold up to 80 command sequences in its memory.

14 bytes of RAM variables indicates that a program in the Picaxe micro has 14 separate and distinct memory locations to hold changing numbers.

14 byte variables or 7 word variables means that those 14 memory locations could hold either 14 numbers between 0 and 255 (BYTE) or 7 numbers between 0 and 65535 (WORD) or a combination of both.

48 bytes of storage indicates that there is another 48 locations in memory where numbers between 0 and 255 can also be stored.

5 I/O lines tells us that there are 5 TTL level interface locations that can be configured as either inputs or outputs or a combination of both.

(A couple of these lines can also be configured as analog input lines for measuring analog voltages.)

The total amount of memory locations for your program and data is 256.

Integer math means that the micro only knows how to perform math operations on whole numbers. The concept of decimal parts and/or fractions is NOT supported by the micro.

Would you like a translation?

$$7 / 4 = 1$$

NOT 1.75!

The Picaxe 08- is a small easy to use microcontroller that can perform small tasks, whatever they are..?

And did I mention another feature attractive to QRPer's?

The PICaxe microcontroller is cheap!

So lets forget about all the specifications, limitations and technical junk and move on to what it can do for you....the frugal QRPer.



Here is a complete Picaxe 08M microprocessor system from QRPme. The tiny board is called a Maine BUGS board. It contains a serial downloading circuit, microprocessor, battery, speaker, temperature sensor and two LEDs.

Here is a simple little Picaxe program:

```
Blink: High 0  
      Pause 500  
      Low 0  
      Pause 500  
      Goto Blink
```

A Picaxe micro containing this program would toggle it's input/output line #0 every 500 milliseconds. If an LED was hooked up to the I/O line, it would blink on and off every 500 milliseconds...

Not all that useful unless you want a blinking LED for something...

Now that wasn't too hard to understand!

The Picaxe language, or set of command words, was developed for programming beginners and is used extensively in schools around the world. The Picaxe microcontroller was developed here in the UK and is the primary micro used in UK school systems. Most commands are simple English words that cause the micro to act in a manner indicated by the meaning of the word....

Did I mention that the Picaxe micro is simple and easy to use.....AND at approx. \$4.00 each is pretty cheap?

Here are some typical commands in the Picaxe programming language:

Branch

Button

Count

Lookup

Nap

Read

Sound

Sleep

Toggle

Wait

Write

These words have everyday meanings that most everyone would understand. The Picaxe micro performs actions very similar to the meanings of the words...

For example:

Count counts the pulses on an input line.

Lookup looks up a number in a table.

Nap causes the micro to take a short 'do-nothing' nap.

Sound command causes the micro to make a sound.

Sleep causes the micro to take a slumber longer than a short nap.

Toggle causes the micro to change the state of a logic line: from high to low or low to high.

Wait causes the micro to wait a specific length of time as specified in milliseconds.

Another simple program might be:

Noise: Sound 4,(120,10)

 Sound 4,(0,1)

 Sound 4,(100,10)

 Sound 4,(0,1)

 Goto Noise

If a speaker was properly hooked up to a Picaxe running this program, the speaker would emit sounds akin to an European Police horn.....

If you wanted to test a set of headphones or small speaker element, you could use this program in a Picaxe:

```
Test: For b0=0 to 125 step 5
      Sound 4,(b0,2)
      Next b0
      Pause 1000
      Goto Test
```

A stepped tone audio sweep signal would be produced on Picaxe I/O line #4 starting at the lowest frequency, ending at the highest frequency with 26 discrete frequency steps.

I know! I slipped you a Mickey there...

What the heck is b0 you say?

b0: the internal Picaxe name for the 1st memory location available inside the Picaxe micro for saving variable data used by a Picaxe program.

Variables are one of the most confusing things when it comes to programming a Picaxe or any other computer for that matter!

Variables and program data memory can be compared to a set of lockers.



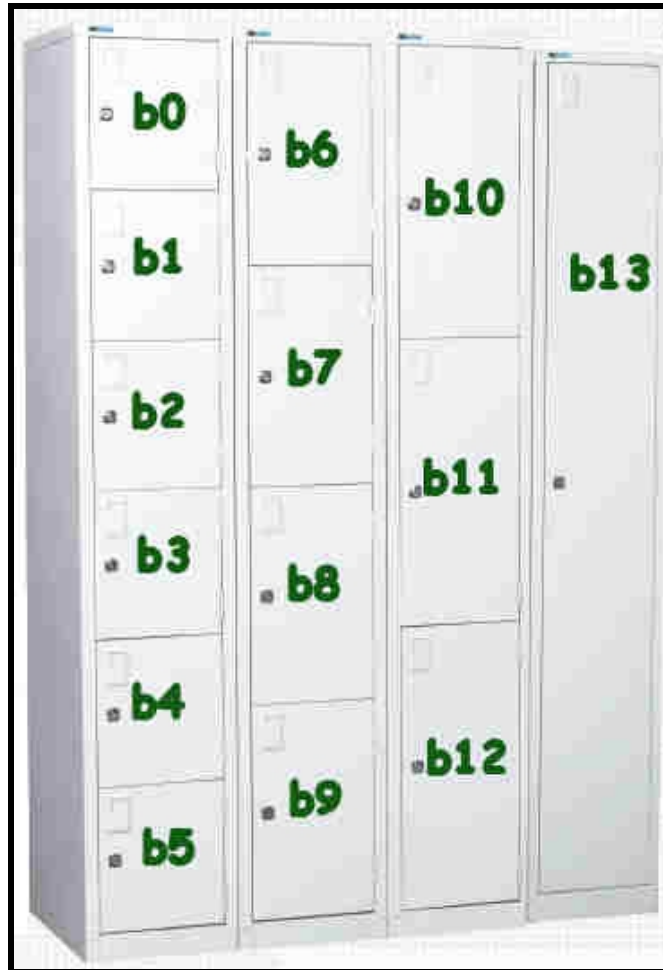
What do you use a set of lockers for?

To store Stuff!
**To preserve and protect your
personal pelf!**

Every time you open the locker up,
mess around inside and then close it
up, chances are, the pelf contained
therein will be different from when
you open up the locker. The contents
of the locker are always changing...
or variable!

The Picaxe 08M microcontroller has
14 locations, or lockers, where your
program can hold stuff/pelf...
variables...(changing numbers)

The Picaxe has some ready-made nametags for these lockers and they are b0 through b13.



Now that is just fine and dandy, the lockers already have some tape on them with the names b0 through b13

BUT,

that doesn't help you at all in keeping track of what your program is keeping track of inside each locker....

So you can make up your own names, using the Symbol command!

This helps you, and others, understand what is going on inside your program AND allows you to edit and change your program easily.

Take this Picaxe program:

```
Symbol FREQ = b0
```

```
Symbol DELTA = 5
```

```
Symbol LEN = b1
```

```
    LEN = 2
```

```
Test: For FREQ = 0 to 125 Step DELTA
```

```
    Sound 4,(FREQ,LEN)
```

```
    Next FREQ
```

```
    Pause 1000
```

```
    Goto Test
```

Using the Symbol command is like slapping a piece of masking tape over the original piece of tape and writing your own name on the locker.

One locker is named FREQ and contains a number that determines the frequency of the sound. Another locker is named LEN and holds a number representing how long you want each tone to sound... You can also take a number like 5 and apply the name DELTA to it. Just like when we name the number 3.14159 PI.

So you can see that the Picaxe micro is..... **CHEAP!**

The commands are simple easy to understand words.

Variables and Symbols are fairly simple concepts and allow you to easily keep track of things in you program.

Even school children, little boys and girls in the 5th grade, can get the Picaxe micro to do things.

YOU should try it!

THANKS FOR ATTENDING!

WIREX