Re: universal programmer

Source: http://sci.tech−archive.net/Archive/sci.electronics.design/2005−12/msg04219.html

- *From*: Dr. Anton T. Squeegee <SpammersAreJerks@xxxxxxxx>
- *Date*: Fri, 23 Dec 2005 17:53:16 −0800

In article <11qoqhlquql4kd3@xxxxxxxxxxxxxxxxxx>, "Abstract Dissonance"
<Abstract.Dissonance.hotmail.com> says...

> Since I'm trying to get into MCU it seems I will need a programmer to get
> anywhere. From looking online it seems that any "decent" programmer is
> pretty expensive(1k+) and those that are cheap seem to offer very few
> features and few chip support.

This is my understanding. I invite others to offer constructive
correction if I mis−speak.

There are two distinct classes of device programmer. The dividing
line is whether they're dedicated to memory devices, such as EPROMs,
EEPROMs, FLASH devices, etc., or whether they are true 'universal'
programmers with pin−driver architecture.

Dedicated programmers are fairly simple units, designed to be
inexpensive and to perform a limited range of functions for just a few
families of chips. This is because the programming algorithms required
for EPROMs and such are not particularly complex in themselves.

The next step up is an architecture called 'pin−driven.'
Programmers based on such technology are at least an order of magnitude
more complex −− and, consequently, more expensive −− than those that are
not pin−driver capable.

Neat thing about pin drivers is that they're software−controlled.
You can program any pin in the programming socket to do anything you
need it to do: Vcc, ground, address, data, programming voltage,
programming waveform, whatever. Pin driver technology also gives you
very precise control over signal timing.

This means that you can, in theory, set them up to handle
read/program for any imaginable programmable device. Functionality like
that does not come cheaply.

> I was thinking that it shouldn't be very difficult to program just about any
> chip by using a computer if the computer had "access" to all the pins on the
> chip. From looking at a few data sheets it seems that its very easy to

> program a MCU by simply handling the procedure through the software.

It's a lot more complex than it may appear at first glance. True "universal" programmers are, in fact, small dedicated−purpose computers designed to do exactly as you describe.

Example: I use Data I/O's Unisite and 2900 series programmers. They both require their own operating system, and they both contain all the elements of a microcomputer: CPU, memory, hard drive (in the case of the Unisites, anyway), and I/O functions.

The Unisi