YAHOO! MAIL Classic

CHIPS Newsletter Vol 5 Thursday, April 29, 2010 12:07 AM

From: "chips@elproducts.net" <chips@elproducts.net>

To: "Chuck Hellebuyck" <chuck@elproducts.com>

CHIPs newsletter

Vol 5 / April 28, 2010

Official Newsletter of *Chuck Hellebuyck's* Electronic Products

http://www.elproducts.com/

Topics

Archive of MPLAB

When writing my books I always try to use the latest version of MPLAB. By the time the book actually goes to print and then sold, the release level of MPLAB may be changed. When you consider my Beginner's Guide to Embedded C book is almost 2 years old, you can easily understand why so many updates to MPLAB have occurred since the book was written. If you ever want to use the exact same versions I used, check out the Microchip archives at the link below. You can get all the older versions of MPLAB there.

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023073

Development Boards

I used the low pin count 20 pin development board in my Beginner's Guide to Embedded C books since it came with the PICkit 2 programmer in the PICkit 2 Starter Kit. I'm looking at moving up to a bigger part for future books, specifically in the 28 pin package size. I'm happy to report I have several

choices that I like when it comes to 28 pin development boards. The first is the Microchip 28 pin version of the starter kit board shown below in Figure 1. The picture is a little fuzzy bit you can see it has a switch, four LEDs and a potentiometer just like the 20 pin board. The problem with this board for my purposes is the area below the chip is too small for a breadboard like I used in volume 2. I like to have a breadboard area for expansion.

Figure 1: Microchip 28 pin Board

My next option is an early prototype from my friends at CHIPAXE.com. They took the Parallax Stamps in Class board and made a 28 pin version that can be plugged into a PICkit programmer. This one has a breadboard area built-in and headers for the I/O pins. It also has a 5v regulator and 9v battery clips so it can be powered away from the PICkit programmer. You can see this one below in Figure 2. I already had developed some simple projects with this when I took the picture.

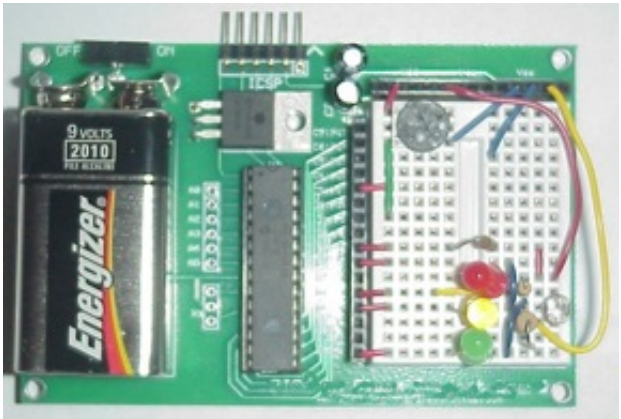

Figure 2 : CHIPAXE.com 28 pin board

The final one I'm considering is something I helped create with my friends at CHIPAXE.com. They sent me an early prototype of our creation which is an Arduino style board that plugs into a PICkit programmer. With this option you plug in a breadboard to build your circuit which can be one of those Arduino shields (plug-in board) that fits into the top of the connectors on this board. The board sparked a whole new concept for my embedded C books called Simple-C (which I'll talk about later) so this may be the board of choice.

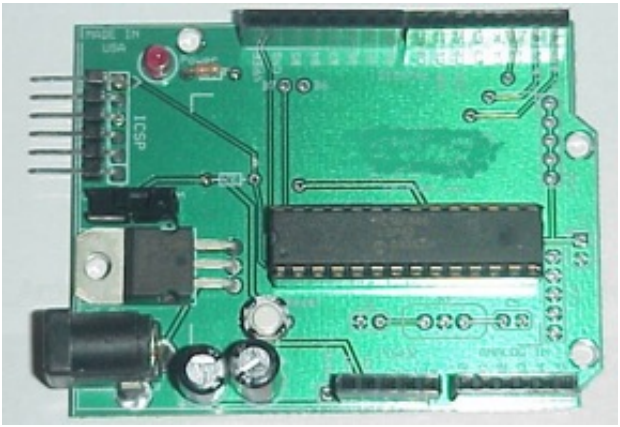

Figure 3 : Arduino Style Board from CHIPAXE.com

You can buy the Microchip board but you can't buy these CHIPAXE boards yet. I'd like your feedback on which one you like for any 28 pin development. Email me at chuck@elproducts with your opinion.

16F88

Speaking of development, if I had to do it over again I think I would have chosen an 18 pin PIC16F88 part for my Embedded C books. It also has a Microchip development board that can fit a small breadboard and it can debug in-circuit without an adapter. That is a big advantage over the 16F690 which needs an adapter for debug. This is a great little part to add to your toolbox of parts. You even gain some RAM. If you have a program that requires quite a bit of variable space, then the 368 bytes of RAM in the PIC16F88 vs. the 256 bytes in the PIC16F690 can make a difference. The PIC16F88 also supports self-write memory, so you can program it via a serial port bootloader. This can be a nice feature if you like bootloaders. The more I work with bootloaders though, the more I prefer a direct programmer like PICkit. It's less complicated to use a programmer and the price of a PICkit 2 or PICkit 3 or any of their clones are very affordable. A bootloader requires you to pre-program the part with the bootloader software and then your configuration bits are locked in. Direct programming allows more direct control and you can debug in-circuit.

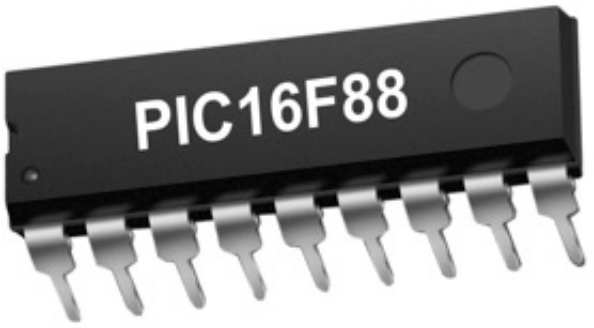

Figure 4: PIC16F88

PIC16F88 Features:

* 7 Kbytes/4 K-word program memory (14-bit address)
* 368 RAM
* USART
* 7 10-bit ADCs
* 3 timers
* 1 Capture/Compare/PWM peripheral
* 2 comparators
* Internal oscillator up to 8 MHz
* 256 bytes of EEPROM
* Self-write Memory

Amicus18

I've tried most of the BASIC compilers and modules available to the hobbyist or even professional. A more recent product I've played with is the Arduino module but unfortunately it's not programmed in BASIC and it's also not Microchip PIC based and I prefer to work with PICs. Well now there is a solution for us PIC users. It's the Amicus18 module from Crownhill Associates in the UK. The module is built to the same connector footprint as the Arduino but is based on the PIC18F25K20. The module has its own bootloader and can be programmed with the free Amicus compiler which is a lite version of the Proton BASIC compiler limited to the PIC18F25K20. The compiler download includes its own integrated design environment (IDE) similar to BASIC Stamp, Basic Atom and others. So you get a free compiler, free IDE, no programmer required because of the bootloader and USB connectivity. In fact the module can be powered from the USB port so you don't even need an external power source to get your first LED to flash. The IDE also allows you to add a programmer

option so programming the PIC18F25K20 with a PICkit programmer is an option. This is the path being taken by one of the new Amicus resellers in the USA BeginnerMicro.com. My friends at CHIPAXE.com and BeginnerElectronics.com joined forces on this one to launch a new website BeginnerMicro.com with a main focus on this platform.

The Amicus compiler is actually a very powerful compiler with floating point math, easy single command control of ADC, PWM, LCD, interrupts, Graphical LCD and more. I am working with the guys to create project books based on the Amicus18 so look for those later in the year. One thing nice about the Amicus compiler is the software is very similar to the PICBASIC Pro compiler I use in several of my books so writing code for this module is much easier for anybody familiar with PICBASIC Pro, BASIC Stamp or Basic Atom. The Proton compiler is also available for a reasonable price to move beyond the Amicus and 18F25K20 so anything developed for the Amicus can easily be ported over to larger or smaller parts in the Microchip PIC family if you purchase the Proton compiler which BeginnerMicro will also offer. One note though, the Amicus runs at 3v not 5v because the 18F25K20 is a 3v microcontroller. This can add a little confusion to the beginner when they try to connect circuitry designed for 5v. I'm told BeginnerMicro will help you through all this with application notes and other sources of information.

Figure 5: Amicus18 Board

Simple-C

From my experience with the BASIC compilers and the various C compilers, it was clear to me that a need to simplify C was desired by those just getting started programming microcontrollers in C. My Beginner books were my first attempt at doing that but I still recognized that the model of pre-written commands or functions, which is the basis of the various BASIC compilers, was needed for the C world. The Arduino software is a version of C but built to be

simplified into to pre-written functions just like a BASIC compiler already has. This has been attempted before with many C compilers having pre-written libraries of functions. The CCS C compiler actually does a great job of this while other C compilers don't. The problem with many of these compiler libraries is they are typically written for someone who already understands the C language very well. If you are a beginner then you can't benefit much from them.

The Arduino software did a good job of making that beginner connection and has contributed to its success. I wanted to do the same for Microchip PIC users. My next Beginner's Guide to Embedded C book (Volume 3) was already going down the library path when it occurred to me that I was creating the same basic structure for Microchip PICs as Arduino did. I wasn't copying it exactly but I was creating a common set of functions to speed up the book's project development and in the process created a beginner's set of pre-written libraries. I'm still working out the kinks but look for this simplified version of C in my next book. From there it will launch a whole series of project books to help my readers produce C based projects much quicker and easier.

Conclusion

I've been so busy with projects, I completely lost track of time and realized I haven't released a new newsletter in over a month. In fact I missed the whole month of March and now it's almost the end of April. I do plan to release these more often so please be patient with me. The Simple-C concept is really working out well. I will also expand it into the 18F parts in the future but volume 3 will stick with the 16F family based on reader feedback. I am planning on launching a magazine in the future targeted toward the beginner microcontroller market. Anyone signed up for this newsletter when the magazine launches will get a special bonus for being the early fans of my work. I will talk more about this in the future because I'm still working out the magazine details but it will be unique and bound like a book rather than the typical magazine. Think of it as a book that reads like a magazine.

Please send me any feedback, questions and comments on anything discussed here or any ideas you have for future topics to chuck@elproducts.com. I do read them all and try to answer all your questions though if you want me to fix your code or write you a custom version, I probably won't have the time.
See you in the next release which will hopefully be quicker.