



CHIPS Newsletter Vol 10

Tuesday, November 16, 2010 8:03 PM

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

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

CHIPS

Vol 10 / November 15, 2010



Official Newsletter of *Chuck Hellebuyck's* Electronic Products

<http://www.elproducts.com/>

Topic

Real Time Clock/Calendar Chip

RS232 Shield

SimpleC Sample

USB to RS232 Converter

Conclusion

It's another month and another newsletter. This month I introduce a few new products from Microchip along with a new shield for the CHIPINO module from chipaxe.com. I even show a SimpleC example of how to use the shield. Lets start with the new Real Time Clock chip from Microchip.

Real Time Clock/Calendar (RTCC)

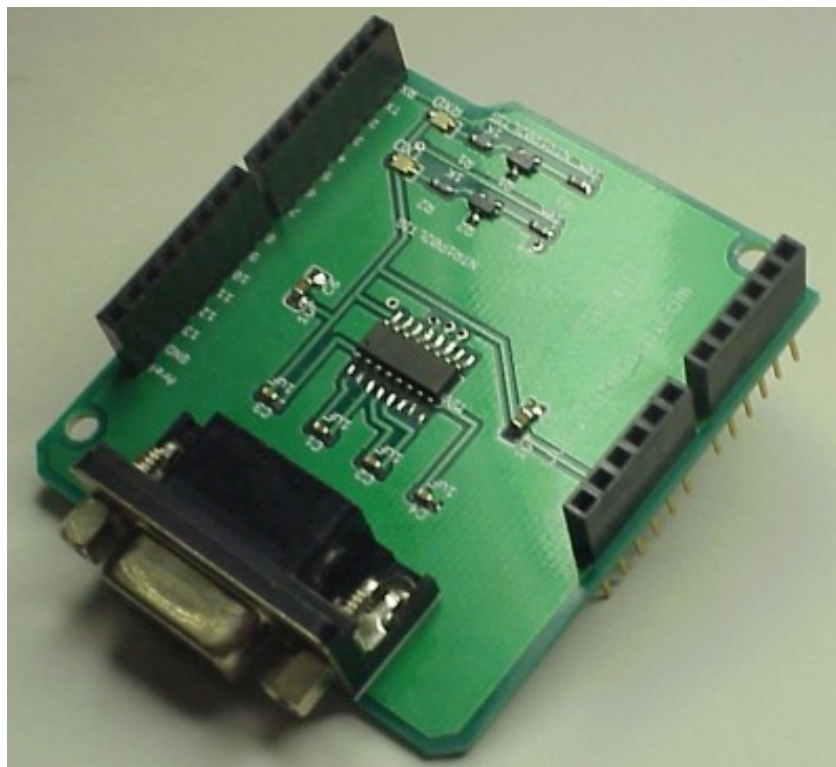
Microchip recently released their first stand alone RTCC chip under the part number MCP79410. It communicates and is controlled via I2C communication. I haven't had much time to try this out but I will in the future. I've used a PIC microcontroller as a clock by using the Timer 1 external clock oscillator option and a 32 khz crystal. In fact I wrote about it in the September 2006 Nuts and Volts Magazine and also republished it in my book "Getting Started with PICs - 2006". In my example I just created a clock using my original Ultimate OEM module and PICBASIC PRO compiler. This new RTCC offers many more features such as an alarm, calendar function, battery back-up so it will continue to run if main power is lost. If you get a chance to use this before me send me

to run on main power is best. If you get a chance to see this before me, send me an email about your application. I believe the chipaxe.com guys have a shield design in the works for this part also.



RS232 Shield

Speaking of shields, I received a sample of the RS232 CHIPINO shield from chipaxe to play with for my book "Beginner's Guide to Embedded C Programming - Volume 3". One of the SimpleC projects is to communicate via the USART of the CHIPINO. This new shield made the hardware connections very easy to setup.



In this world of USB everywhere an RS232 module may not be that useful but a simple USB to RS232 cable makes this adaptable to any computer. Most PCs

Simple USB to RS232 cable makes this adaptable to any computer. Most PICs have a USART but only a small group offer a USB peripheral so this is still a handy way to communicate with a PC from your embedded device. There are many devices designed with a DB9 RS232 connection so having this in your shield arsenal seems very logical to me. It can easily accept shields on top so this becomes just one layer in your CHIPINO project. In fact let me use this topic to introduce you to an example of how SimpleC can make programming easier. I'll show you an RS232 SimpleC code sample of how to control this shield. (If you weren't aware, SimpleC is the library of functions I show you how to build in my next book "Beginner's Guide to Embedded C Programming - Volume 3" hopefully released by the end of the year.)

SimpleC Sample

The project is created in MPLAB and the SimpleC library file is included with the project. To control the USART at 9600 baud the `init_serial(9600)` function is issued from the SimpleC library. This sets up the USART for 8N1 communication. From there a simple `printf` and `getch` functions control all the communication. The file asks for a keyboard input and then sends back what it received. This is why I like SimpleC so much. I can do simple projects like this in no time at all.

```

/*****
A simple demonstration of serial communications which
Uses the on-board hardware USART of the Microchip
PIC16F886 device. The project will use the printf
command to send a phrase and ask for you to send it
something via the keyboard and it will echo it back.
The program will keep asking for input until power
is lost.

```

Hyperterminal and RS232 module settings

Pin 0(C7) Rx - to Tx of RS232 module
Pin 1(C6) Tx - to Rx of RS232 module

Hyperterminal settings
Bits per second: 9600
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

```

*****/

```

```

#include "simplec1.1.h"

```

```

void main(void){

```

```

init_micro();      // Setup micro ports
init_serial(9600); // set up the USART baud rate to 9600

// *** Serial Communication Main Loop
while (1)
{
printf("Press a key and I will echo it back:\r\n");
char input = getch();// read a response from the user
printf("I detected [%c]\r\n",input);// echo it back
} // end while
} // end main

```

USB to RS232 Converter

Since I mentioned USB to RS232 conversion I have to mention the new USB to RS232 chip from Microchip. It's the MCP2200 and also comes in a nice development module. The module sells for \$23.99 and converts an RS232 DB9 connector to a mini USB connector. The kit includes the cable so you can connect the module to your PC USB port and then install the proper drivers that come with the module. This is an alternative to the FTDI type converters. You can get the MCP2200 for under \$2.00 to make your own converter board. The MCP2200 comes in a SOIC package so I find it easier to solder than an FTDI chip. One point to note though is the MCP2200 requires an external 12Mhz Crystal or Resonator so that adds some cost and space but still cheaper over all then an FTDI chip.



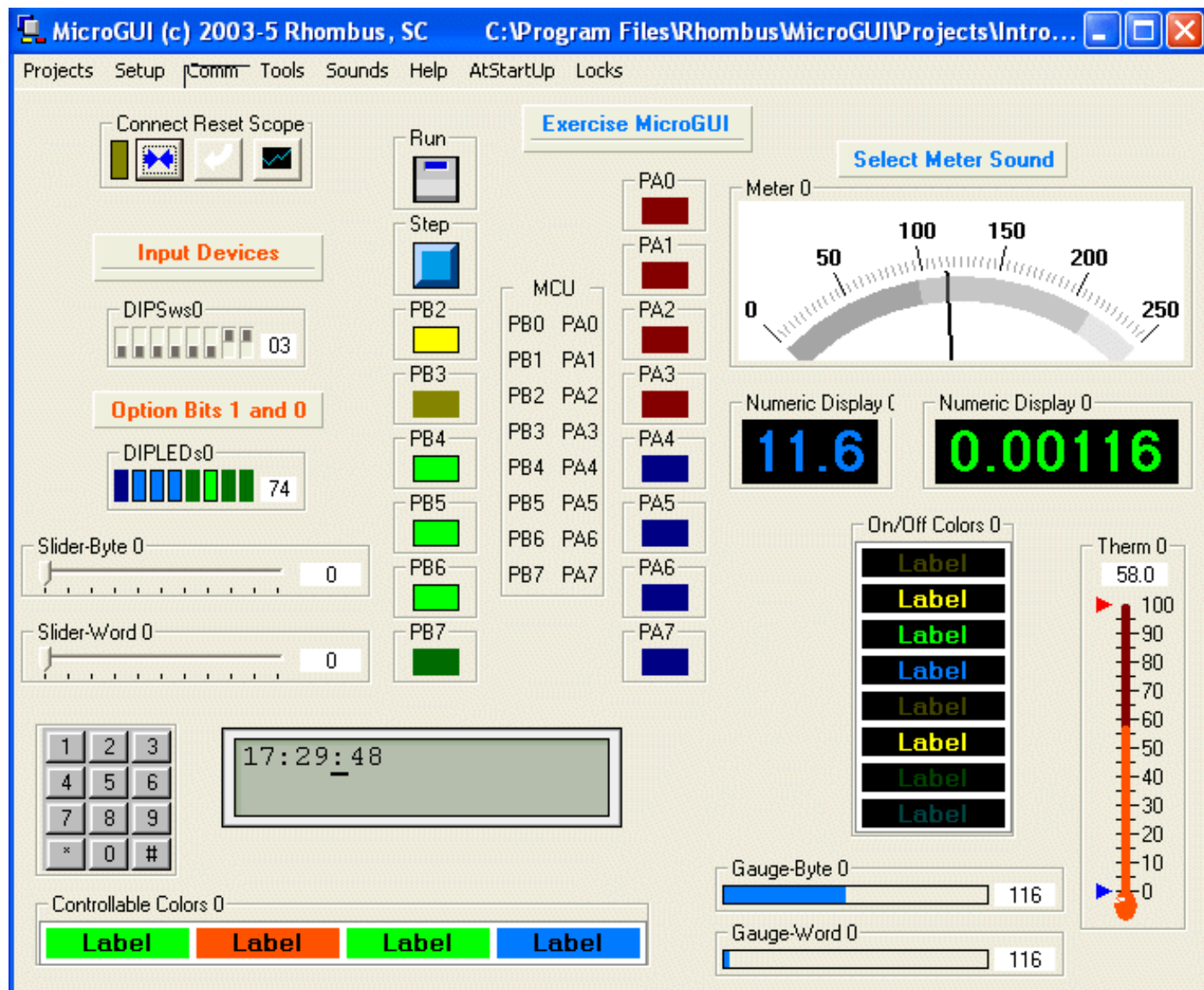
MicroGui

Many years ago I played around with the MicroGui software from Rhombus-Tek.com. The full URL is:

http://rhombus-tek.com/Orig_uGUI.html

I haven't tried this with the RS232 shield setup but I plan to do that. The

MicroGui is a great graphical interface you can adapt to your project. I have used it successfully with PICBASIC PRO in the past and then got lost in other projects. During this newsletter write-up I visited the MicroGui site again and see they still offer it in a trial version you can download for free with a 30 day limit. Even purchasing it is inexpensive at under \$20.



If you try this out, email me a description of your project. I'd really like to see how people are using this software.

Conclusion

I guess this was a RS232 dedicated newsletter with a little RTCC thrown in for variety. I still find it fascinating when I can control a microcontroller module from a PC with very little code. Everybody has a PC these days so if you can create

a PC with very little code. Everybody has a PC these days so if you can create an RS232 module with USB connectivity and then control it from a PC, you have so many projects you can create with this. If you have any questions or comments send them to me at chuck@elproducts.com. I really would like to see what people are doing with their embedded RS232 applications and for those getting started with CHIPINO email me your feedback there as well. I'm told its already off to a great start as it officially launched less than two months ago.

[Unsubscribe from CHIPs Newsletter](#)