



Proton DS Lite: Un SISTEMA

Hoy en día los sistemas de desarrollo para microcontroladores incluyen diferentes aplicaciones capaces de comunicarse entre sí para realizar los proyectos completos de una forma ágil y efectiva. La suite Proton DS Lite incluye un conjunto de programas coordinados y estructurados para el desarrollo de sistemas basados en microcontroladores PIC.

Es indiscutible que en el mundo laboral tienen una gran importancia las **herramientas** utilizadas para desarrollar las tareas asociadas a los diversos tipos de trabajo, siendo también necesario tener los **conocimientos adecuados** sobre las características y el funcionamiento de las herramientas para obtener unas óptimas prestaciones.

Además, es importante que las herramientas utilizadas estén **integradas** entre sí de forma lógica, ya que de otro modo se corre el riesgo de no conseguir **resultados adecuados**, desperdiciando tiempo y recursos.

Por ejemplo, consideramos un electricista que cuenta con una nutrida colección de destornilladores con punta de estrella pero que tiene que trabajar sobre una serie de elementos dotados de tornillos con cabeza hexagonal. Evidentemente para poder realizar su **trabajo**, y **de forma óptima**, ha de contar con las herramientas adecuadas para este tipo de tornillos. Las cosas no son diferentes cuando se desarrolla **software** para **microcontroladores**. También en este caso la disponibilidad y el conocimiento de programas válidos e integrados permite conseguir **resultados óptimos** en tiempos razonablemente cortos.

Aquí entra en juego el nuevo sistema **IDE** (acrónimo de **I**ntegrated **D**evelopment **E**nvironment), es decir **Entorno de Desarrollo Integrado**, para microcontroladores **PIC**. Este término hace referencia a un conjunto que comprende **varias herramientas**: Editor, compilador, programador y otras utilidades, todas integradas en un mismo programa. En la jerga informática se utiliza el término **Suite** para referirse a un programa que integra varias aplicaciones, como por ejemplo la conocida suite **Office** de **Microsoft**, que integra los programas Word, Excel, Access, Powerpoint y FrontPage.

Este artículo es una introducción a un entorno de desarrollo para PIC, la **suite Proton Development**, un sistema de desarrollo bastante reciente, muy ágil y eficiente.

SUITE PROTON DEVELOPMENT

Se trata de un completo sistema de desarrollo resultado de la colaboración de tres importantes empresas del sector:

Lo más significativo de este sistema de desarrollo es, sin duda, el **lenguaje utilizado**: El **BASIC**. Se trata de una novedad muy importante para los programadores, tanto profesionales como aficionados. El menú **Help (Ayuda)** de **Proton DS Lite** dispone de una lista con todas las **instrucciones BASIC**, incluyendo explicaciones exhaustivas, ejemplos y referencias en línea de las instrucciones y de sus funciones relacionadas. También forman parte de la suite **Proton DS Lite** una nutrida serie de **programas de ejemplo** que, de forma sintética y directa, enseñan la sencillez y la flexibilidad de la sintaxis del lenguaje BASIC de la versión propuesta por **Crownhill** (estos ejemplos se encuentran en la carpeta **Samples Lite** de **Proton DS Lite**).

PROTON Plus

Como seguramente muchos de vosotros sabéis, **BASIC** es sin duda uno de los lenguajes de programación más sencillos, representando un lenguaje equilibrado entre el potente lenguaje **Assembler** y los más desarrollados, y complejos, **C**, **PASCAL**, **JAVA**, etc.

integrado para PIC

- **Crownhill Associated Ltd** (www.crownhill.co.uk) ha realizado el Compilador BASIC PROTON Plus.

- **Mecanique** (www.mecanique.co.uk) es la empresa diseñadora del Sistema Integrado de Desarrollo (IDE).

- **Labcenter Electronics Ltd 2004** (www.labcenter.co.uk) ha desarrollado Proteus Virtual Simulator Machine (VSM). Este artículo analiza la versión reducida, y **gratuita**, denominada **Proton DS Lite** (se puede descargar de la Web de **Crownhill**). Al tratarse de una versión gratuita tiene, como es obvio, **algunas limitaciones** sobre la versión comercial, si bien estas limitaciones **no reducen la potencia** del sistema integrado, ya que conciernen sobre todo al **código** y a los **modelos de PIC** que se pueden utilizar.

Desarrollado alrededor de los años 60, desde la década de los 80 se ha ido actualizando poco a poco. Adaptaciones actuales, como **Visual BASIC**, se utilizan hoy en día en ámbitos profesionales para desarrollar aplicaciones.

El **compilador BASIC** incluido en **PROTON Plus** genera el código fuente en **Assembler** (archivos con extensión **.ASM**), totalmente compatible con **MPASM**, es decir el Assembler de **MICROCHIP**. Además genera programas en **formato ejecutable** (archivos con extensión **.HEX**) listos para cargarse en un microcontrolador **PIC**.

Como hemos señalado, en la versión **LITE** este compilador presenta algunas **limitaciones** en el **código** y en los **modelos** de PIC soportados, que, como veréis, no reducen las prestaciones de las funciones básicas del sistema.

El **número máximo** de líneas de código BASIC que pueden ser escritas es **50**. Pasado este límite no se compila el programa entero. Quien piense que **50 líneas** de código BASIC son muy pocas, con el ejemplo siguiente, que gestiona el control de un pulsador, tal vez cambie de opinión:

Button PX,0,2,20,PVAR,1,PREM1

Con esta instrucción se configura el pin conectado al pulsador (**PX**) como entrada con pull-up interno (**0**), la gestión de los ciclos de espera (**2**), el valor de autorepetición (**20**) y el salto a la subrutina **PREM1** si se presiona (**1**) el pulsador. **¡Todo en una única línea de código!** Otra limitación es la **imposibilidad** de utilizar la **directiva INCLUDE**, por lo que si han sido realizadas partes del programa en archivos diferentes, el intento de unirlas a través de **INCLUDE** no funcionará.

Los microcontroladores que se pueden utilizar se reducen a los siguientes **modelos: 12C508,**

12F675,16F628A y 16F877 (la versión Lite **no** soporta los modelos de **16 bits**). Algunas **funciones trigonométricas con coma flotante** no funcionan ya que solo están disponibles en los PIC que trabajan a **16 bits**.

Por último, solo se puede trabajar con **dos frecuencias de reloj: 4 MHz y 20 MHz**. El resto de frecuencias (3.58MHz, 8MHz, 10MHz, 12MHz, 14.32MHz, 16MHz, 24MHz, 32MHz y 40MHz) solo están disponibles en la versión comercial.

A pesar de estas limitaciones insistimos en la posibilidad de generar un buen número de programas y proyectos. Además hay que tener presente que nuestro objetivo no es vender programas sino **enseñar** a realizar desarrollos para **microcontroladores PIC**.

PROTON DS LITE

Como ya hemos mencionado nuestro objetivo es exponer las posibilidades de este sistema

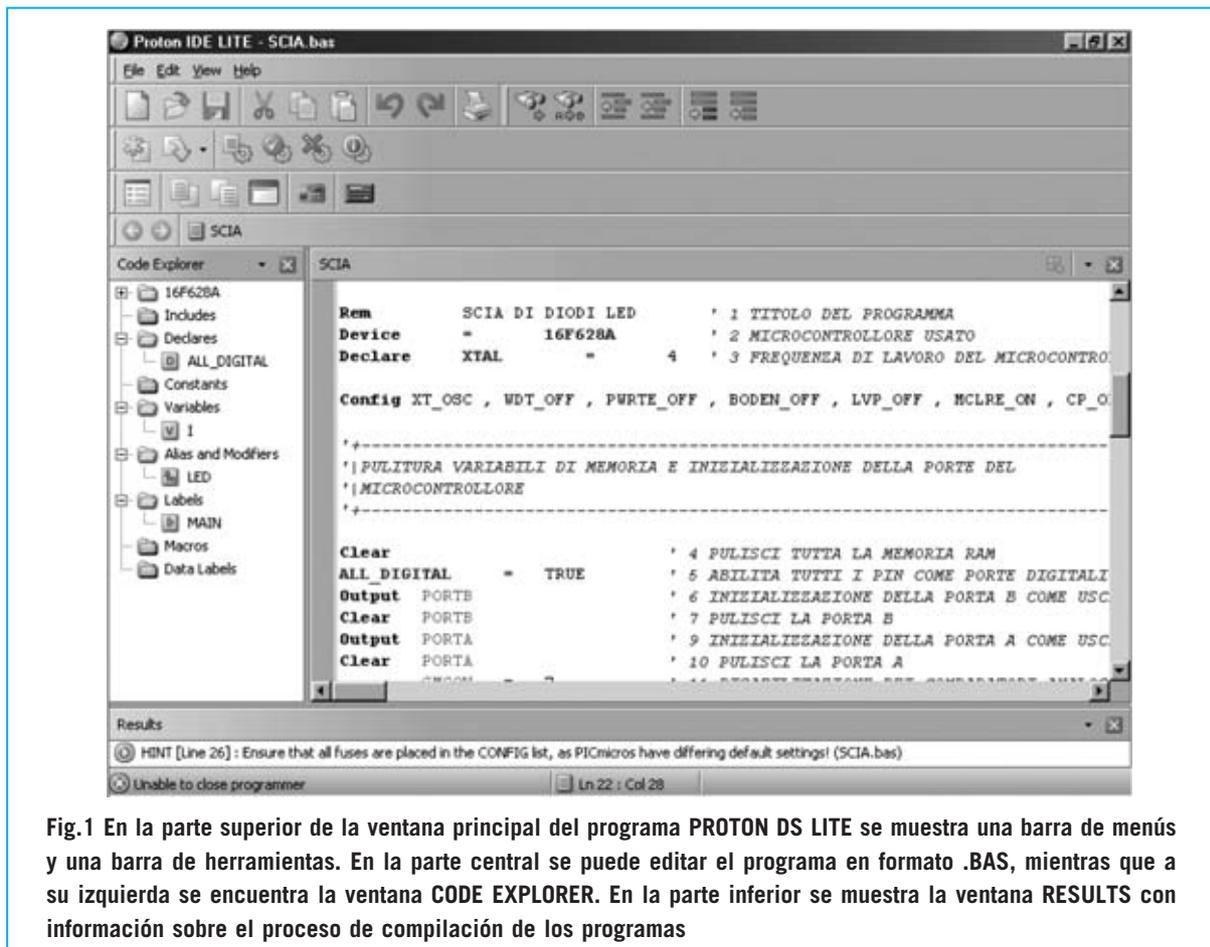


Fig.1 En la parte superior de la ventana principal del programa PROTON DS LITE se muestra una barra de menú y una barra de herramientas. En la parte central se puede editar el programa en formato .BAS, mientras que a su izquierda se encuentra la ventana CODE EXPLORER. En la parte inferior se muestra la ventana RESULTS con información sobre el proceso de compilación de los programas

```

SCIA
Rem      SCIA DI DIODI LED      ' 1 TITOLO DEL PROGRAMMA
Device  =      16F628A        ' 2 MICROCONTROLLORE USATO
Declare XTAL      =      4    ' 3 FREQUENZA DI LAVORO DEL MICROCONTRO

Config XT_OSC , WDT_OFF , PWRTE_OFF , BODEN_OFF , LVP_OFF , MCLRE_ON , CP_O

-----
'| PULITURA VARIABILI DI MEMORIA E INIZIALIZZAZIONE DELLA PORTE DEL
'| MICROCONTROLLORE
-----

Clear
ALL_DIGITAL = TRUE      ' 4 PULISCI TUTTA LA MEMORIA RAM
Output PORTB           ' 5 ABILITA TUTTI I PIN COME PORTE DIGITALI
Clear  PORTB           ' 6 INIZIALIZZAZIONE DELLA PORTA B COME USC
Output PORTA           ' 7 PULISCI LA PORTA B
Clear  PORTA           ' 9 INIZIALIZZAZIONE DELLA PORTA A COME USC
Clear  PORTA           ' 10 PULISCI LA PORTA A

```

Fig.2 Detalle de la ventana del editor. PROTON DS LITE dispone de un editor avanzado que facilita la escritura y la interpretación del código fuente, reconociendo los distintos elementos que forman el programa y utilizando colores diferentes para una óptima interpretación del código.

de desarrollo integrado para posteriormente mostrar cómo utilizarlo, por lo que en este primer artículo presentamos de forma muy general sus **características principales**, ya que en unas pocas páginas no se pueden mostrar todas sus posibilidades.

Como hacemos usualmente reproducimos algunas imágenes, en este caso del programa **Proton DS Lite**, para facilitar la asimilación de las descripciones.

La Fig.1 muestra la pantalla principal de **PROTON DS**. Incluye una ventana de **edición**, otra con el analizador de código (**Code Explorer**) y otra con los resultados (**Results**). A continuación realizamos una breve descripción de cada una.

EDITOR

En la parte central se encuentra la ventana de **edición** (ver Fig.2), es decir el lugar donde se puede escribir, visualizar, modificar y salvar en un archivo de texto el **código fuente** de las instrucciones que forman un **programa**.

En la Fig.2 se muestra uno de los ejemplos de programas **BASIC** contenido en el CDROM **CDR.1585 (SCIA.BAS)**.



Fig.3 En la ventana OPTIONS se pueden personalizar los caracteres y los colores del editor ajustándolos a las exigencias específicas de cada uno.

Se trata de un editor avanzado dotado de un gran número de características, que incluye un **anali-zador sintáctico** del programa editado capaz de analizar comandos ASM, comandos Proton BASIC, comentarios, variables, constantes, etc., mostrando los elementos con **colores diferentes** para facilitar la lectura del código fuente.

Su estructuración jerárquica en forma de árbol muestra el **modelo de microcontrolador** utilizado, **declaraciones, constantes, variables, ficheros incluidos, etiquetas, macros** y otros datos sin necesidad de ir explorando el código fuente. Haciendo click sobre los elementos el cursor del editor se posiciona rápidamente en el punto del **código fuente** donde se encuentran.

CODE EXPLORER

A la derecha del editor se encuentra la ventana **Code Explorer** (ver Fig.4). Se trata de una ventana muy importante ya que permite navegar de forma sencilla y clara dentro de la estructura del programa que se está editando.

Al hacer click en la carpeta situada al lado del nombre del **microcontrolador** se muestra una lista de sus **registros** (ver Fig.5), cuyo contenido detallado y explicado se puede visualizar haciendo click sobre el nombre del registro. Por ejemplo, en la Fig.6 se muestra el registro **Option_Reg**.

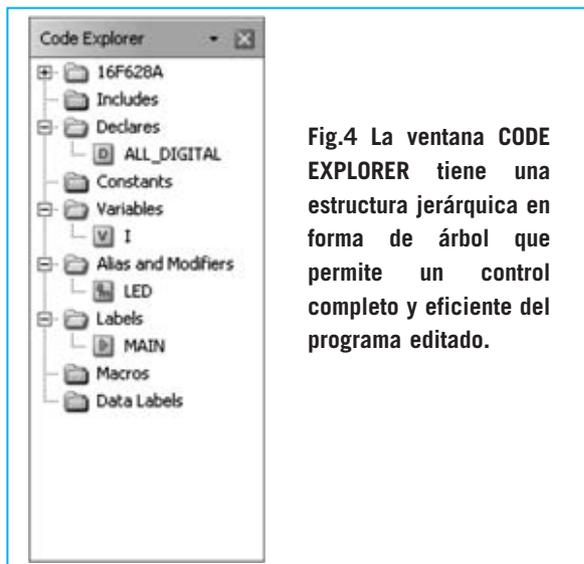


Fig.4 La ventana CODE EXPLORER tiene una estructura jerárquica en forma de árbol que permite un control completo y eficiente del programa editado.

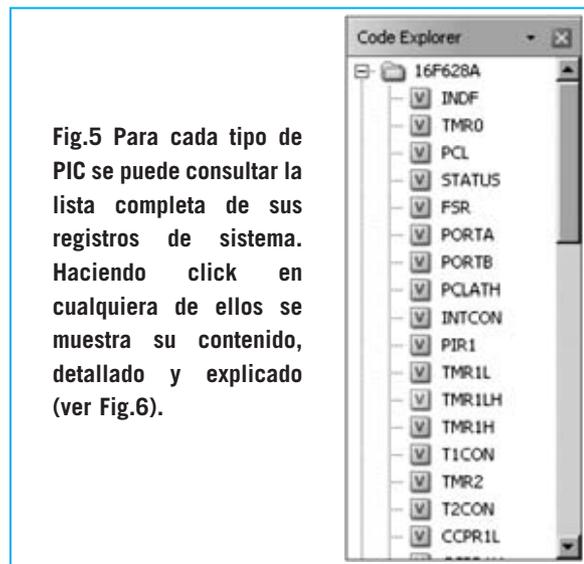


Fig.5 Para cada tipo de PIC se puede consultar la lista completa de sus registros de sistema. Haciendo click en cualquiera de ellos se muestra su contenido, detallado y explicado (ver Fig.6).

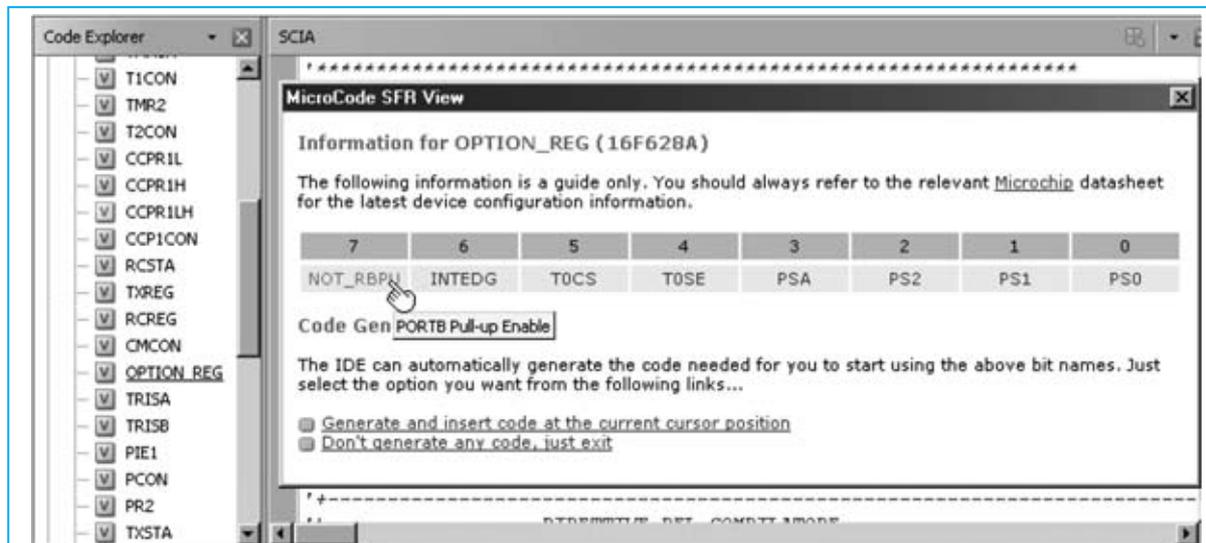


Fig.6 Esta ventana se abre haciendo click sobre el registro OPTION_REG. Se muestra el nombre, descripción y la funcionalidad de cada bit que lo compone. Ventanas análogas se muestran para cada uno de los registros de sistema.

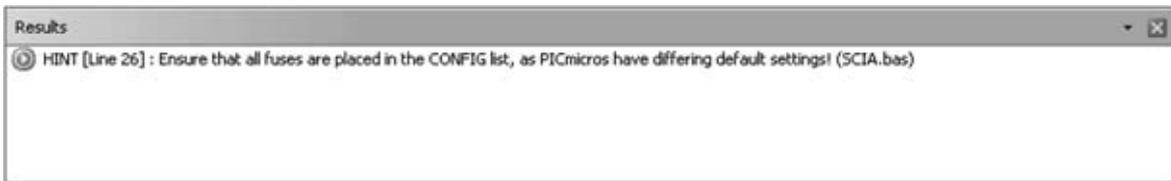


Fig.7 La ventana RESULTS no solo muestra los resultados de la compilación (ver Fig.9), también proporciona sugerencias (HINTS) y advertencias (WARNINGS).

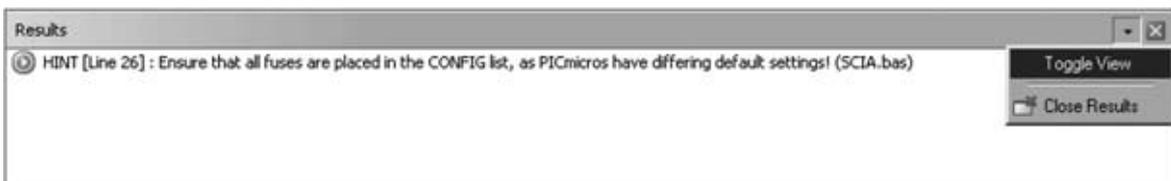


Fig.8 Para ver la información correspondiente a la compilación hay que hacer click en la pequeña flecha situada en la parte superior-derecha y seleccionar la opción TOGGLE VIEW.

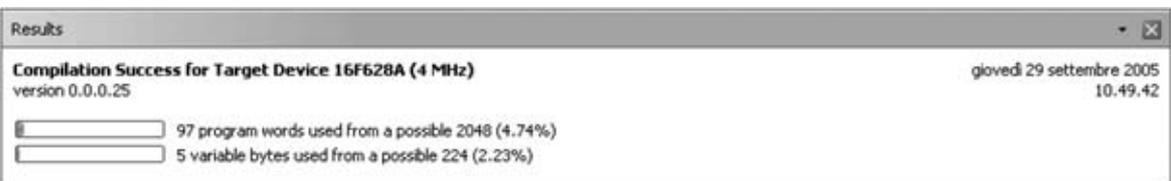


Fig.9 En este caso la compilación se ha realizado sin errores. El programa proporciona información relativa a la memoria ocupada por el programa (4,74 %) y por los datos (2,23 %).

RESULTS

En esta ventana, que se encuentra en la parte inferior de la pantalla principal, se muestra la información correspondiente al **resultado** de la **compilación BASIC**.

Si procede se muestran mensajes de advertencia (**Warnings**) y/o sugerencias (**Hints**) referentes a las instrucciones afectadas (hemos reproducido en la Fig.7 un mensaje que **sugiere** al programador que revise la configuración).

Cuando los programas contienen **errores de compilación**, identificables por la inscripción **Error**, se muestra toda la información necesaria para identificar el tipo de error y las líneas incorrectas.

Cuando la compilación acaba **correctamente** se muestra información sobre el espacio de **memoria ocupado**, tanto de **programa** como de **datos** (ver Fig.9).

MICROCODE LOADER

Seleccionando la función **Loader** del menú **View** (ver fig.10) se abre una ventana con el programa **MicroCode Loader** de **Mecanique** (ver Fig.11).

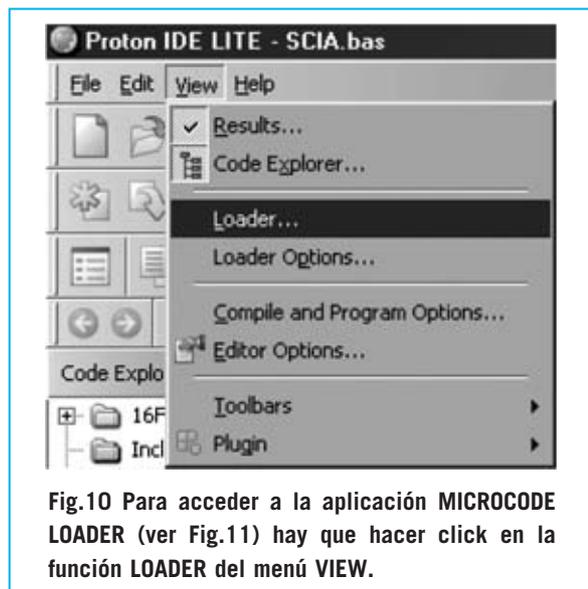


Fig.10 Para acceder a la aplicación MICROCODE LOADER (ver Fig.11) hay que hacer click en la función LOADER del menú VIEW.

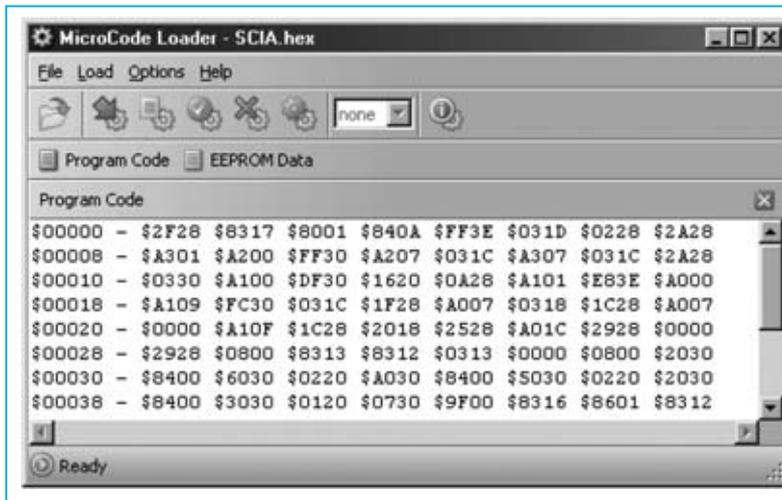


Fig.11 La aplicación MICROCODE LOADER comunica el ordenador, a través de un puerto serie, con el programa BOOTLOADER del PIC, permitiendo de esta forma su programación sin necesidad de utilizar programadores externos.

Las series **16F87x** y **18Fxxx(x)** de los micros **PIC** son capaces de escribir en su propia memoria utilizando un **puerto serie**, sin necesidad de un dispositivo de programación externo. Esta operación se puede realizar gracias a un pequeño programa residente precargado en estos micros denominado **bootloader**.

Este programa reside en las primeras **256 posiciones** de memoria en los **PIC 16F87x** y en las primeras **336 posiciones** en los **PIC 18Fxxx(x)**.

MicroCode Loader es un programa que se instala en el ordenador y que, a través de un puerto serie **RS232 (COM:)**, se comunica con el programa **bootloader** del microcontrolador, pudiendo **cargar** así de una forma sencilla los programas realizados.

La versión **Proton DS Lite** permite la utilización de los micros **16F876**, **16F877**, **18F252** y **18F452**, existiendo un **bootloader** específico para cada modelo y frecuencia de reloj utilizada. Los programas bootloader específicos para estos cuatro modelos de PIC se encuentran en el directorio **MCLoader** de **Proton DS Lite**.

PLUGINS del IDE

Proton DS ha sido realizado teniendo en cuenta una característica muy importante: La flexibilidad.

La posibilidad de incorporar **plugins**, que en la jerga informática se suele utilizar para referirse a **accesorios añadidos**, permite incorporar a

este **IDE** nuevos programas, utilidades, funciones y aplicaciones diseñadas por terceros y, por qué no, por vosotros mismos.

Para añadir estos **accesorios añadidos (plugins)** se utiliza la función **Plugin Editor** (ver Fig.12). Además del editor, la versión **Lite** contiene unos cuantos **accesorios estándar** incorporados como **plugins**:

- ASCII Table**
- IDE Assembler View**
- IDE HEX View**
- IDE Serial Communicator**
- Labcenter Electronics PROTEUS VSM Programs**

A continuación exponemos una explicación breve, pero sin duda importante e interesante para los programadores, de las posibilidades de cada uno de estos **plugins**.

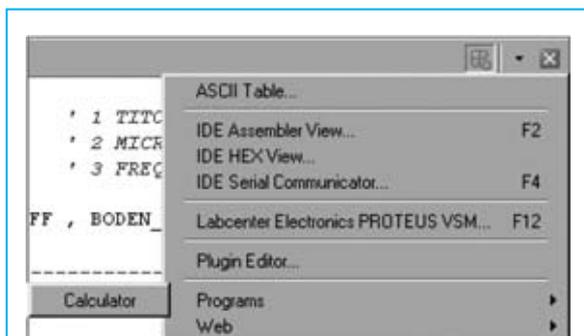


Fig.12 Haciendo click en el icono PLUGIN se abre un menú con la lista de accesorios añadidos (plugins) instalados. De forma estándar se incluyen varios plugins en PROTON DS LITE.

PROTEUS VSM (Labcenter Electronics)

Producido por Labcenter Electronics Ltd 2004, Proteus VSM es una suite completa para el desarrollo de esquemas eléctricos (schematic capture), incluyendo un simulador modelo SPICE y el novedoso VSM (Virtual System Modelling), una tecnología que permite la simulación de proyectos que utilizan microcontroladores, entre los que se encuentran una gran variedad de micros PIC. Por supuesto además del microcontrolador se

administran el resto de componentes electrónicos, tanto analógicos como digitales.

Con los modelos VSM del microcontrolador es posible realizar depuraciones virtuales del firmware, es decir del software incorporado en el hardware, para controlar los valores de los registros o el contenido de la memoria de programa (program memory), introduciendo puntos de ruptura (breakpoints) o utilizando el modo de depuración paso a paso (step by step).



Fig.13 Haciendo click en LABCENTER ELECTRONICS (ver Fig.12) se abre una ventana con la lista de las tarjetas de hardware virtual (Virtual Hardware Boards).

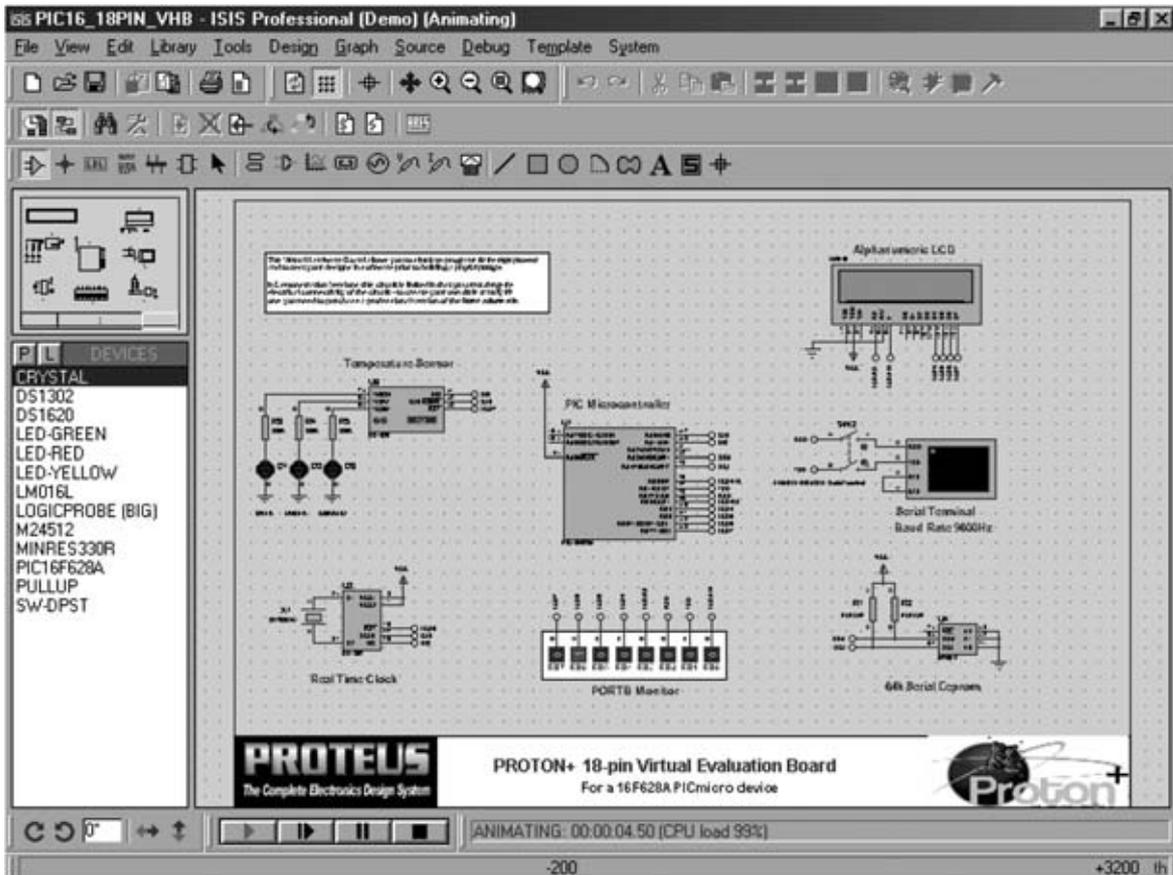


Fig.14 Las tarjetas de hardware virtual (VHB) incluyen modelos animados de periféricos y reproducen de forma virtual tanto el hardware analógico como el digital. Gracias a estos modelos es posible simular el funcionamiento en tiempo real y depurar los programas.

| CHAR | DEC | HEX | BIN | Description |
|------|-----|-----|----------|---------------------|
| NUL | 000 | 000 | 00000000 | Null character |
| SOH | 001 | 001 | 00000001 | Start of Header |
| STX | 002 | 002 | 00000010 | Start of Text |
| ETX | 003 | 003 | 00000011 | End of Text |
| EDT | 004 | 004 | 00000100 | End of Transmission |
| ENQ | 005 | 005 | 00000101 | Enquiry |
| ACK | 006 | 006 | 00000110 | Acknowledgment |
| BEL | 007 | 007 | 00000111 | Bell |
| BS | 008 | 008 | 00001000 | Backspace |
| HT | 009 | 009 | 00001001 | Horizontal Tab |

Fig.15 Haciendo click en el plugin ASCII TABLE (ver Fig.12) se abre una herramienta indispensable para cualquier programador: Una tabla ASCII.

```

Assembler - SCIA.asm
File Edit View Help
SCIA
SCIA
1 /-----
2 / Code Produced by the PROTON+ LITE Compiler. Version 3.1.1
3 / Copyright Rosetta Technologies/Crownhill Associates
4 / Written by Les Johnson. July 14 2005
5 /-----
6 NOLIST
7 #include "C:\IC-PROG\PRG DEMO BASIC\SCIA LED\SCIA.PBP"
8 LIST
9
10 _I = 36
11 #Define LED PORTB
12 F2_SOF equ $ ; SCIA.PRP
13 F2_EOF equ $ ; SCIA.PRP

```

Fig.16 Al Hacer click en el plugin IDE ASSEMBLER (ver Fig.12) se abre esta ventana.

Fig.17 Por cada instrucción BASIC se generan las correspondientes instrucciones en Assembler.

```

Assembler - SCIA.asm
File Edit View Help
SCIA
SCIA
48 F1_000039 equ $ ; in [SCIA.BAS] CMCON = 7 ' 11 DISA
49 movlw 7
50 movwf CMCON
51 F1_000053 equ $ ; in [SCIA.BAS] LED = 1
52 movlw 1
53 movwf PORTE
54 MAIN
55 F1_000061 equ $ ; in [SCIA.BAS] For I = 1 To 7 Step 1
56 movlw 1
57 movwf _I
58 fr8lb2
59 movlw 8
60 subwf _I,W

```

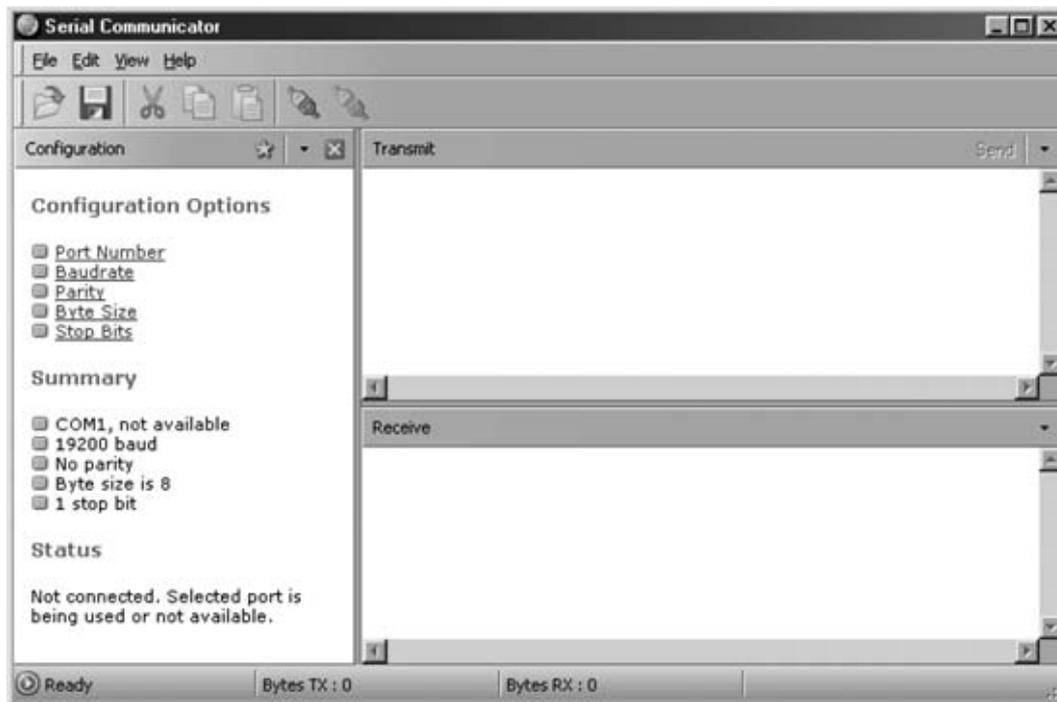



Fig.19 Al click en el plugin IDE SERIAL COMMUNICATOR (ver Fig.12) se lanza una aplicación que permite la conexión entre un puerto serie del ordenador y un dispositivo hardware externo.

Una vez ajustado el protocolo se puede **salvar** la configuración en un área denominada **Favorites** (Favoritos) que posibilita una **carga rápida** de las opciones guardadas.

- **Transmit**: Aquí se introducen los datos a transmitir al dispositivo conectado al puerto serie del ordenador.

Además de los **caracteres alfanuméricos** comunes existe la posibilidad transmitir una serie de **caracteres de control**.

- **Receive**: En este área se muestra la información transmitida por el dispositivo externo al PC.

Con esta aplicación es muy sencillo realizar pruebas en los desarrollos que prevén el intercambio de datos entre un **microcontrolador PIC** y un **ordenador personal**.

PROGRAMS (Calculator)

Cuando se instala **Proton DS Lite** se genera como **plugin estándar** un enlace a la **calculadora** de **Windows**.

REQUISITOS MÍNIMOS

La versión **Lite** de **Proton DS** precisa, como mínimo, un procesador **Pentium** o compatible, una memoria **RAM** de **32 MB**, aunque es recomendable disponer de al menos **128 MB**, y de **100 MB** de espacio libre en el **disco duro**.

Los **sistemas operativos** soportados son: **Microsoft Windows 98SE, ME, 2000** y **XP**.

CONCLUSIÓN

En próximos artículos explicaremos la forma de **instalar** y **configurar Proton DS Lite**. Además expondremos cómo realizar **compilaciones** de programas en **BASIC**, **simulaciones** con tarjetas **VHB** y la **programación** final del **PIC**.

PRECIO del CD-ROM

El programa **Proton DS Lite** versión **1.036** es una de las aplicaciones incluidas en el CDROM **CDR.1585**, proporcionado junto con tarjeta para PIC **LX.1585** presentada en esta misma revista. Quienes deseen adquirir únicamente el **CDROM** pueden solicitarlo al precio de**13,60 €**