



SISTEMA INTEGRADO

En este artículo presentamos un completo Sistema Integrado de Control con microcontrolador. Su programación es bastante sencilla ya que se realiza en lenguaje BASIC, permitiendo aplicaciones de todo tipo, desde aplicaciones básicas hasta aplicaciones de tipo industrial. Su sencillez de programación permite su utilización por quienes no son expertos programadores de microcontroladores en Assembler.

Un campo que siempre hemos potenciado en nuestra revista es el desarrollo con **microcontroladores**, ya que es la tendencia de diseño actual. En concreto hemos expuesto cursos completos, sistemas y aplicaciones para **ST6**, **ST7** y **PIC**, detallando todas sus características y procedimientos de programación.

De hecho muchos lectores han contactado con nosotros para agradecernos la gran claridad con la que hemos explicado la utilización de es-

tos microcontroladores, con los que muchos han logrado realizar sus propios **sistemas de control**.

No obstante las personas menos familiarizadas con el lenguaje Assembler y con limitadas posibilidades de desarrollar circuitos impresos nos han preguntado sobre la posibilidad de disponer de una **tarjeta** que, además del **microcontrolador**, incorpore las **interfaces** con el **exterior** y que sea **fácilmente programable**

para hacer más inmediata y sencilla la realización de **sistemas de control**.

Como es nuestra costumbre, respondiendo a vuestras demandas, exponemos en este artículo un **Sistema Integrado** con microcontrolador que incluye los circuitos de interfaz con los sensores y transductores integrados en una **única tarjeta**.

No obstante, para que pueda ser utilizada por todo el mundo, incluyendo aplicaciones industriales, hemos optado por un completo **Sistema Integrado** que proporcionamos **montado** y en conformidad con las **normativas CE**.

El **Sistema Integrado** que proponemos dispone de **relés** en sus salidas, lo que permite utilizarlo en todos los contextos en los que las instrucciones solicitadas por el sistema a controlar son del tipo **ON/OFF**.

Este sistema, desarrollado por **EPTAR S.R.L.** y que nosotros proporcionamos con la referencia **KM2107**, satisface prácticamente todas

las exigencias, y nos ha convencido no solo para la buena relación calidad-precio, sino especialmente para su versatilidad y por su sencillez de programación.

NOTA: Quienes deseen conocer la compañía **EPTAR**, empresa especializada en automatismos, puede visitar su página Web www.eptar.it.

CARACTERÍSTICAS del KM2107

El corazón del **Sistema Integrado KM2107** es un **microcontrolador Atmel ATmega8535L** tipo **AVR** (Advanced Virtual RISC) capaz de ejecutar una instrucción por cada impulso de reloj.

La interfaz del microcontrolador con el exterior está compuesta por:

- **6 salidas con relé.**
- **8 entradas** para señales digitales tipo **TTL**.
- **1 Entrada/Salida TTL/analógica.**
- **2 entradas analógicas** directas al convertidor **ADC** de **10 bits** incluido en el microcontrolador.

con **MICROCONTROLADOR**



Fig.1 Fotografía frontal del Sistema Integrado de Control KM.2107.

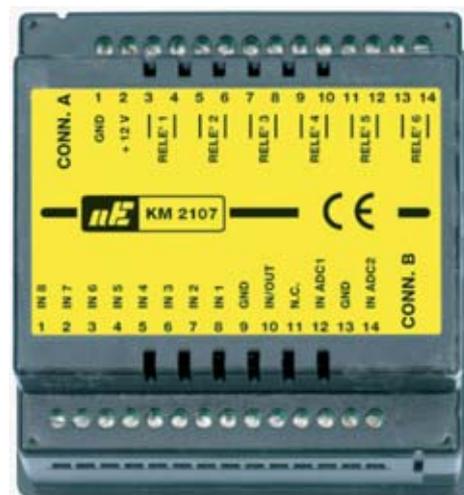


Fig.2 El Sistema Integrado KM.2107 visto por el lado de las conexiones.

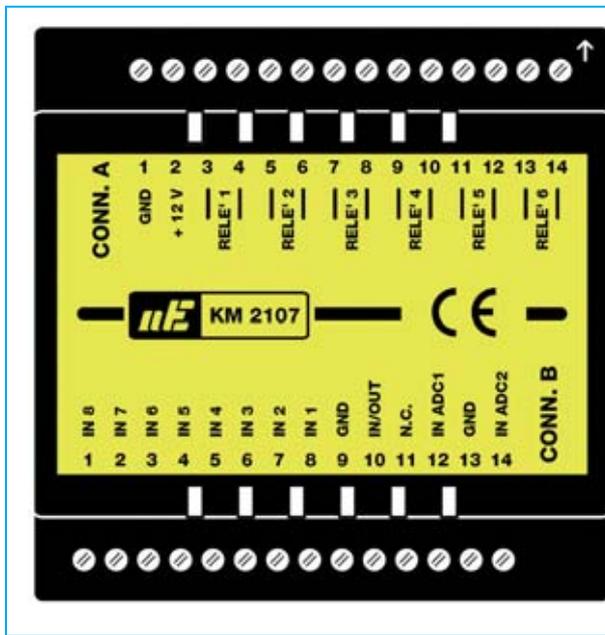


Fig.3 Esquema a tamaño natural de la parte posterior del Sistema Integrado KM2107 con sus dos conectores de 14 terminales. Al pegar la etiqueta incluida en el kit hay que hacer corresponder la flecha de la caja, aquí reproducida en color blanco, con el terminal 14 de CONN.A. Las correspondencias entre los terminales y las señales se describen ampliamente en el artículo.

El sistema es fácilmente **programable** ya que permite escribir programas en un lenguaje de **alto nivel** que no precisa un conocimiento profundo de las características del microcontrolador ni tampoco de sus instrucciones maquina.

El sistema se conecta directamente a un **ordenador personal**. La programación se realiza en el PC y se vuelca al Sistema Integrado **KM2107** mediante **puerto paralelo**.

CONECTIVIDAD EXTERNA

El Sistema Integrado se presenta como una caja cuyo panel frontal dispone de un **display** de **16x2 líneas** y **tres pulsadores** de membrana (ver Fig.1).

La parte posterior del contenedor dispone **dos conectores** de **14 contactos**. Las señales correspondientes a cada uno de sus terminales de conexión se muestran en la Fig.2.

Al conector **CONN.A** están conectados internamente **6 relés** mecánicos de **30V/DC - 230 V/AC 5 amperios**, con sus contactos normalmente **abiertos**.

Conector CONN.A

Terminales 1-2: Conexión de **alimentación**, que puede ser de **12 V/CA** o bien **12 V/CC**. En caso de alimentarse en continua el polo **negativo** va conectado al **terminal 1**, que representa la **masa (GND)** del **Sistema Integrado**.

Terminales 3-4: Salida del **RELÉ1**.

Terminales 5-6: Salida del **RELÉ2**.

Terminales 7-8: Salida del **RELÉ3**.

Terminales 9-10: Salida del **RELÉ4**.

Terminales 11-12: Salida del **RELÉ5**.

Terminales 13-14: Salida del **RELÉ6**.

Las **8** entradas digitales binarias **IN1-IN8** se encuentran en el conector **CONN.B**, con su potencial referido a la masa de alimentación. La **masa** de alimentación está disponible en los **terminales 9** y **13** del mismo conector, además de en el **terminal 1** de **CONN.A**.

También se encuentra en el conector **CONN.B** una entrada-salida TTL/analógica que hemos denominado **IN/OUT Analog/TTL**, y dos entradas analógicas denominadas **INADC1-INADC2**.

Conector CONN.B

Terminal 1: Entrada **IN8**.

Terminal 2: Entrada **IN7**.

Terminal 3: Entrada **IN6**.

Terminal 4: Entrada **IN5**.

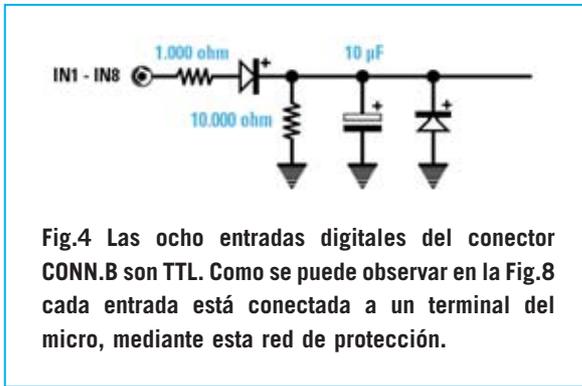


Fig.4 Las ocho entradas digitales del conector **CONN.B** son TTL. Como se puede observar en la Fig.8 cada entrada está conectada a un terminal del micro, mediante esta red de protección.

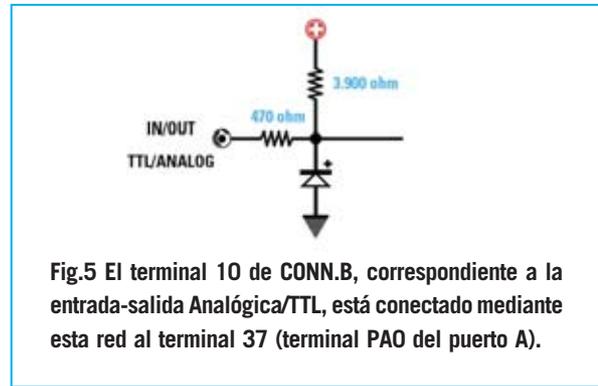


Fig.5 El terminal 10 de **CONN.B**, correspondiente a la entrada-salida Analógica/TTL, está conectado mediante esta red al terminal 37 (terminal PA0 del puerto A).

Terminal 5: Entrada **IN4**.

Terminal 6: Entrada **IN3**.

Terminal 7: Entrada **IN2**.

Terminal 8: Entrada **IN1**.

Terminal 9: Masa (**GND**).

Terminal 10: Entrada-Salida **IN/OUT Analog/TTL**.

Terminal 11: Terminal **libre** para facilitar la conectividad externa.

Terminal 12: Entrada **INADC1**.

Terminal 13: Masa (**GND**), replicada para facilitar la conectividad.

Terminal 14: Entrada **INADC2**.

En la Fig.3 están reproducidos los dos conectores con las correspondencias entre los **terminales** y las **señales**.

Las **8** entradas digitales (**IN1** a **IN8**) son de tipo **TTL** y cada una de ellas está conectada directamente a un terminal distinto del microcontrolador Atmel-AVR, a través de una red protectora (ver Fig.4).

El terminal **Analog/TTL** se conecta a través de la red mostrada en la Fig.5.

Las entradas **INADC1** e **INADC2** están conectadas al conversor **ADC** mediante dos redes de ajuste con una impedancia de entrada de **44.000 ohmios**. Estas redes permiten la conversión a 10 bits mediante el ADC interno del microcontrolador, ofreciendo valores digitales

incluidos entre **000000000** y **111111111** correspondientes a señales analógicas con valores incluidos entre **0** y **5 voltios**.

El conector **CONN.C**, situado en un lateral del contenedor, sirve para conectar el **Sistema Integrado** al **puerto paralelo** del **ordenador** para realizar la programación del microcontrolador.

MICROCONTROLADOR ATMEL-AVR

A continuación vamos a exponer las características principales del microcontrolador **ATmega8535L-AVR**, centrándonos en las características de necesario conocimiento para utilizar adecuadamente el **Sistema Integrado KM2107**.

No obstante quien desee obtener las características completas puede descargarlas gratuitamente de la Web del fabricante www.atmel.com.

NOTA: En el **CD-ROM** que se proporciona con el Sistema Integrado **KM2107** se incluye el **datasheet** completo del microcontrolador en formato **PDF**.

En primer lugar hay que decir que se trata de un microcontrolador con un **bus de datos** de **8 bits** y que dispone de **130 instrucciones maquina**.

El **reloj (clock)**, es decir el oscilador que establece el intervalo de tiempo entre la ejecución de una instrucción maquina y la siguiente, es de **8 MHz**. Puesto que el microcontrolador ejecuta una instrucción por impulso de reloj su rendimiento es de **8 MIPS (8 Millones de Instrucciones Por Segundo)**.

En cuanto al la **memoria** el microcontrolador dispone de:

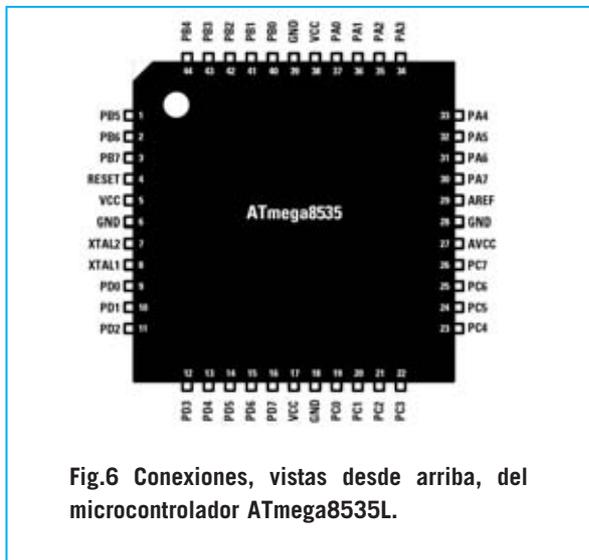


Fig.6 Conexiones, vistas desde arriba, del microcontrolador ATmega8535L.

- Una **memoria EEPROM de 512 Bytes** utilizados para almacenar los datos que han de ser mantenidos durante la ejecución del programa y que no han de ser borrados al desconectar la alimentación.

Como se puede observar en la Fig.6, el procesador **ATMEL8535L** dispone de **44 terminales**, de los que **4** (terminales 6-18-28-39) se utilizan para la conexión a **masa** y **5** (terminales 5-17-27-29-38) para la conexión de alimentación a **+5V CC**.

El **terminal 4** corresponde a la señal de **RESET**, mientras que a los **terminales 7-8** se conecta el **cuarzo** utilizado para la generación de la señal de reloj (clock).

- Una **memoria no volátil** (no pierde su contenido al desconectar la alimentación) de **8 KBytes** a disposición para el programa a ejecutar.

- Una **memoria SRAM volátil** (sí pierde su contenido al desconectar la alimentación) de **512 Bytes** utilizados para almacenar los datos que han de ser mantenidos durante la ejecución del programa y que son borrados al desconectar la alimentación.

Los restantes **32 terminales** constituyen **4 puertos** de comunicación con el exterior de **8 bits** cada uno.

Al **puerto A** están conectadas **5 entradas** de las ocho entradas **digitales** del Sistema Integrado (terminales PA1-PA5), las **2 entradas analógicas INADC1-INADC2** (terminales PA6-PA7) y la **entrada/salida TTL/analógica** (terminal PA0).

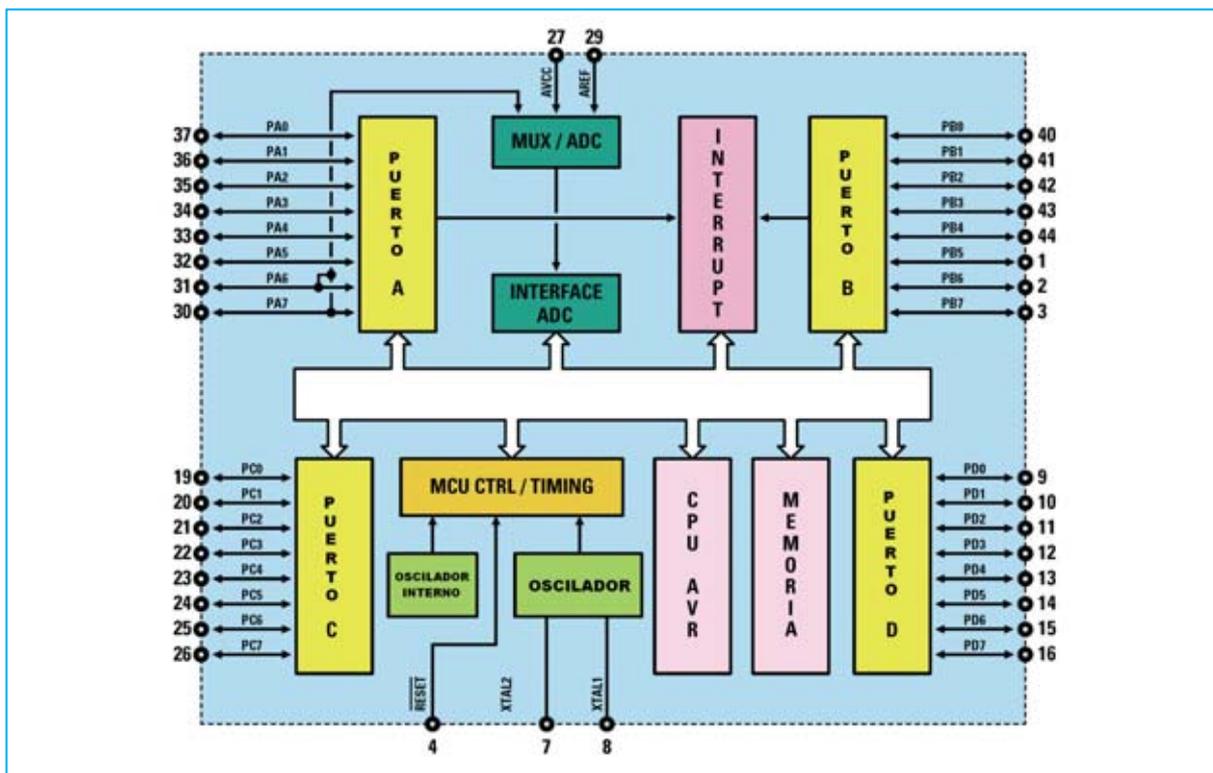


Fig.7 Esquema de bloques del microcontrolador Atmel ATmega8535L. En el CD-ROM que se adjunta con el kit se encuentra el datasheet completo del micro en formato PDF.

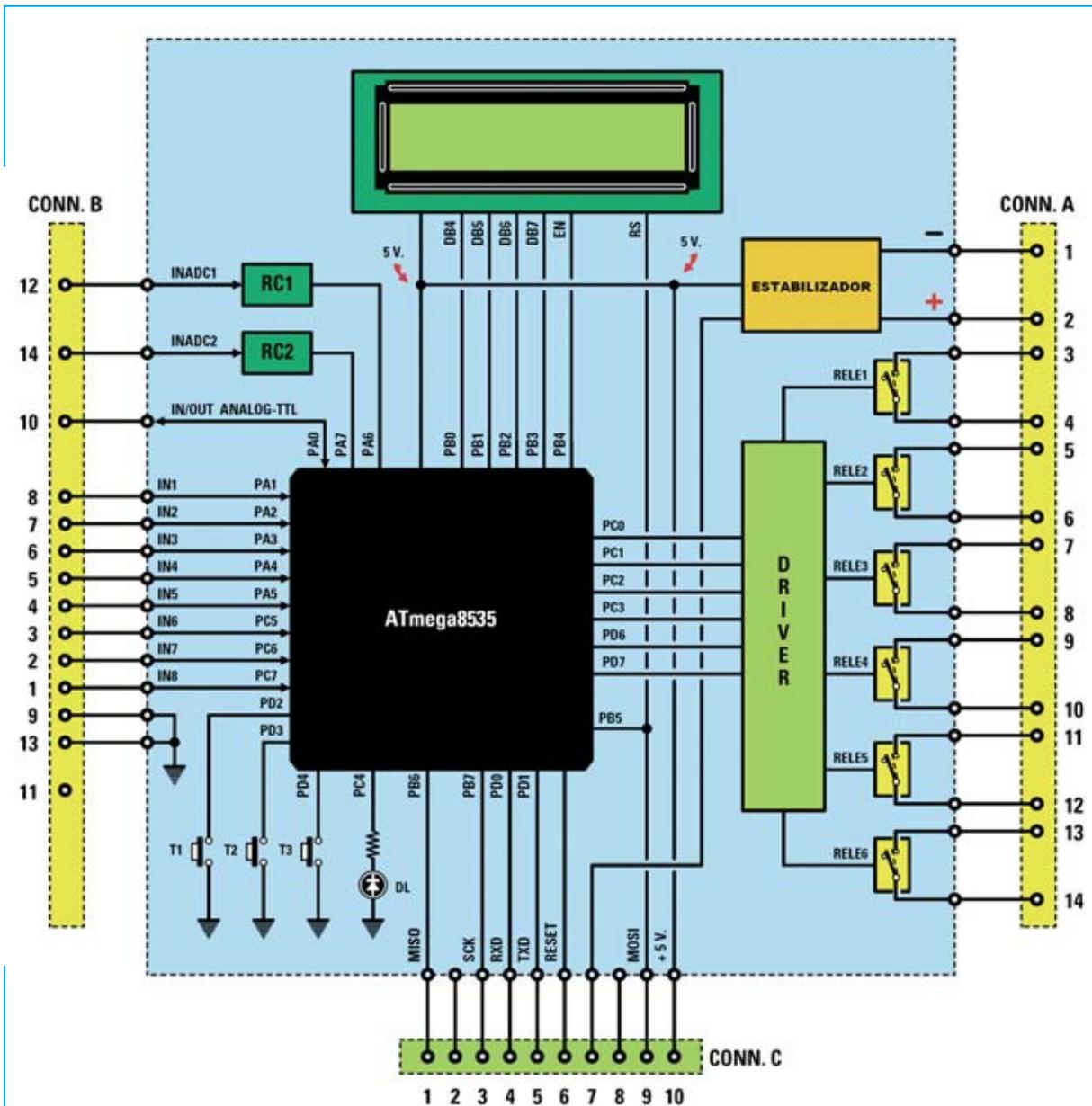


Fig.8 Esquema de bloques del Sistema Integrado KM2107. El micro dispone de 4 puertos de comunicación con el exterior que permiten administrar un display LCD con 2 líneas de 16 caracteres, seis relés, ocho entradas digitales binarias, dos entradas ADC, una entrada/salida Analógica/TTL, tres teclas y un diodo LED. A través del conector CONN.C es posible programar este sistema mediante un PC utilizando programas escritos en BASIC mediante BASCOM-AVR.

Los 8 terminales del **puerto B** (PB0-PB7) se utilizan íntegramente para la gestión del display y para la comunicación con el **ordenador** a través del cual se programa el micro.

Al **puerto C** se conectan 4 de los seis **relés** (terminales PC0-PC3), 3 de las ocho **entradas digitales** (terminales) PC5-PC7 y el **diodo LED** situado bajo el pulsador T1 (terminal PC4).

Por último, al **puerto D** se conectan 2 de los seis **relés** (terminales PD6-PD7) y los 3 **pulsadores** (terminales PD2-PD4). De los **restantes** terminales del **puerto D**, el terminal PD5 no se utiliza, y los terminales PD0-PD1 se utilizan para la comunicación con el ordenador personal.

Para no dar lugar a error aclaramos que para quienes programen el **Sistema Integrado** si-

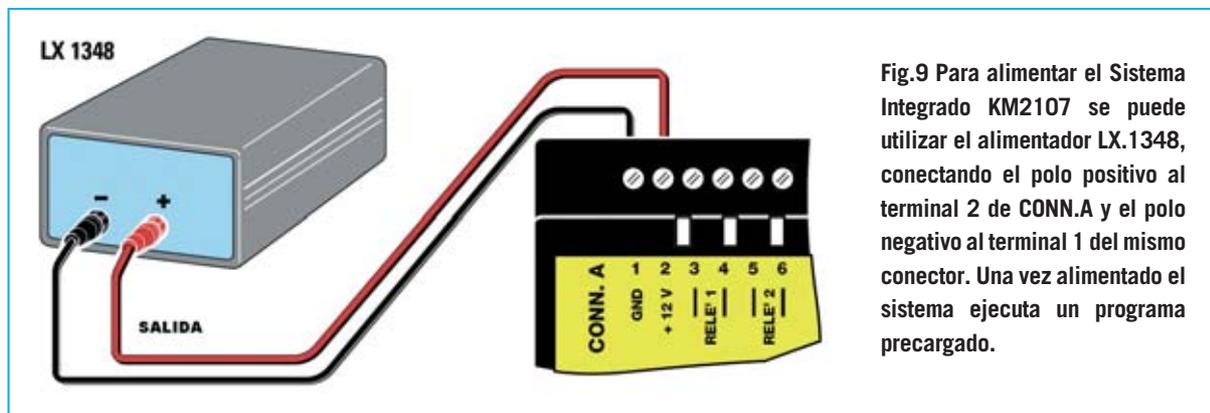


Fig.9 Para alimentar el Sistema Integrado KM2107 se puede utilizar el alimentador LX.1348, conectando el polo positivo al terminal 2 de CONN.A y el polo negativo al terminal 1 del mismo conector. Una vez alimentado el sistema ejecuta un programa precargado.

guiendo nuestras indicaciones no necesitan conocer toda la información que estamos exponiendo, simplemente es suficiente con conocer la forma de manipular la entradas y las salidas.

Esta información es útil para quienes deseen modificar el Sistema Integrado para hacerlo aún más versátil y que responda a aplicaciones **muy específicas**.

ALIMENTACIÓN y PRUEBA del SISTEMA INTEGRADO

En primer lugar hay que alimentar el **Sistema Integrado** conectándolo a un alimentador de **12 voltios** capaz de proporcionar al menos **250 mA**. Se puede utilizar, por ejemplo, nuestro alimentador **LX.1348** presentado en la revista **Nº171**.

Una vez alimentado el **Sistema Integrado** ejecuta un programa de **diagnóstico** que nosotros hemos **cargado**.

Se trata del programa **test.bas**, almacenado en la carpeta **Ejemplos** del **CDR2107** e instalado con el programa **BASCOM-AVR**. El programa sirve para probar el correcto funcionamiento de las tres teclas y de las entradas/salidas del **Sistema Integrado**.

En la primera línea del display aparece el texto **NuovaElettronica**, mientras que en la segunda línea aparecen secuencialmente los mensajes correspondientes a las pruebas que el microcontrolador está ejecutando sobre las entradas **IN1** a **IN8**.

Ya que, como hemos explicado, las ocho **entradas digitales TTL** normalmente están a nivel **bajo** (ver Fig.4), los mensajes mostrarán el estado lógico bajo de las entradas.

Para cambiar el estado lógico basta con proporcionar a la entrada deseada una señal TTL a nivel **alto**, señal que se puede conseguir conectando entre los **+12 V** del alimentador y la **entrada** una resistencia de **15.000 ohmios**.

Si, por ejemplo, se conecta la resistencia entre los **+12 V** y el terminal **8** de **CONN.B** (ver Fig.10), cuando el microcontrolador prueba la entrada **IN1** en el display se visualiza la indicación **IN1 alto**.

Para probar el funcionamiento de las teclas hay que mantener pulsada la **tecla T1** (ver **tecla** con forma de **flecha** en la Fig.1) hasta que aparezca la indicación **tasto1 premuto**. Si se mantiene pulsada también se encenderá el diodo LED situado bajo la tecla, luego se apagará.

Si se mantiene pulsada la **tecla T2** (ver **tecla +** en la Fig.1) aparecerá la indicación **tasto2 premuto**. Si se mantiene pulsada se inicia un ciclo de activación de los seis relés, al que seguirá un ciclo de desactivación.

Por último, presionando la **tecla T3** (ver **tecla -** en la Fig.1) aparecerá la indicación **tasto3**



Fig.10 Las ocho entradas digitales TTL normalmente están en estado lógico bajo. Para llevar las entradas a estado lógico alto hay que conectar una resistencia de 15.000 ohmios al terminal 8 de CONN.B, como se muestra en esta imagen para IN1.

premuta. Si se mantiene pulsada se inicia una prueba sobre las entradas **ADC**.

En el caso de que no se muestre ninguna indicación hay que **verificar** las **conexiones** efectuadas. Si tras verificarlas se determina que están correctas no hay que forzar el circuito, es más conveniente ponerse en contacto con nuestro Servicio Técnico.

EL COMPILADOR BASCOM-AVR

Para programar el **Sistema Integrado KM2107** se puede utilizar cualquier programa o entorno de desarrollo que permita editar el código fuente del programa, su compilación y el traslado del programa compilado a la memoria del microcontrolador **Atmel-AVR**.

Existen varios productos que utilizan fuentes en **Assembler**, otros, en cambio, están basados en lenguajes de alto nivel como **C++** o **BASIC**, es decir lenguajes independientes del tipo de microcontrolador a programar y que, por lo tanto, permiten escribir programas para un microcontrolador sin tener que conocer en detalle el hardware y su conjunto de instrucciones en Assembler.

En este caso, una vez escrito el **código fuente** del programa con instrucciones de **alto nivel** un programa denominado **compilador** traduce el código de alto nivel a **código maquina**, el único código que el microcontrolador es capaz de procesar.

Nosotros hemos optado por el programa **BASCOM-AVR**, desarrollado por la empresa **MCS Electronics**. Este programa permite la **edición** de programas en **lenguaje BASIC**, realiza la **compilación** para obtener código para microcontroladores Atmel-AVR, posibilita la **depuración (debug)** de los programas, dispone de funciones de **simulación** y permite **enviar** el **código** generado al microcontrolador del Sistema Integrado.

El programa **BASCOM-AVR** ha de instalarse en un PC que disponga de las siguientes **prescripciones mínimas**:

- Procesador Pentium 166MMX o superior.
- Memoria RAM de 32 MBytes.

- Espacio libre en disco duro de 200 MBytes.
- Sistema operativo Windows 98 SE, Windows 2000 o Windows XP.
- Puerto paralelo Centronics.

La versión **Demo 1.11.8.1** de **BASCOM-AVR**, así como los **manuales de usuario**, pueden descargarse gratuitamente de la Web **www.mcselec.com**.

La única limitación de la versión **Demo** sobre la completa consiste en que solo se pueden crear programas con menos de **4096 líneas** de código en **lenguaje maquina**, número más que suficiente para la mayor parte de las aplicaciones del Sistema Integrado.

No obstante, ya que no todo el mundo puede conectarse a Internet, hemos incluido en el **CD-ROM CDR2107** la versión Demo del programa **BASCOM-AVR** y su **manual de usuario**. Además hemos incluido los siguientes archivos:

- **Datasheet** en formato PDF del microcontrolador **ATmega8535L**.
- El **artículo original** en formato PDF.
- Útiles **ejemplos** para entender fácilmente la programación en **BASIC**.
- **Aplicaciones** prácticas para el **KM2107**.

No vamos a profundizar en las características del lenguaje **BASIC** y del procesador **ATmega8535L**, ya que no es necesario. En cambio sí vamos a ampliar la descripción del **Sistema Integrado**, a detallar la instalación y la utilización de la plataforma **BASCOM-AVR** y a mostrar algunos ejemplos que permitirán utilizar el Sistema Integrado de forma inmediata.

LA LIBRERÍA IONE.BAS

La posibilidad de utilizar el **lenguaje BASIC** simplifica notablemente la programación del microcontrolador **Atmel-AVR**. No obstante es preciso conocer las conexiones entre el microcontrolador y el resto de componentes del **Sistema Integrado**.

Por ejemplo, para excitar el relé conectado a un terminal del microcontrolador hay que conocer la **dirección** del **puerto** de salida del microcontrolador a cuyo terminal está conectado, además del **identificativo** del terminal.

A	<pre>'NON MODIFICARE QUESTO FILE !!!!! 'contiene l'inizializzazione del basic plc 'e va sempre incluso all'inizio del 'programma principale</pre>	<p>El bloque contiene un comentario recordatorio de que este fichero ha de incluirse en los programas para el Sistema Integrado</p>
B	<pre>\$crystal = 8000000 'frecuencia del cuarzo</pre>	<p>Definición de la frecuencia de reloj del sistema (cuarzo conectado a los terminales 7-8)</p>
C	<pre>Config Lcdbus = 4 Config Lcdpin = Pin , Db4 = Portb.0 , Db5 = Portb.1 , Db6 = Portb.2 , Db7 = Portb.3 , E = Portb.4 , Rs = Portb.5 Config LCD = 16 * 2</pre>	<p>Instrucciones para configurar el display LCD. Gracias a ellas se puede escribir en el display con las instrucciones BASIC "Lcd" y "Locate"</p>
D	<pre>Tasto1 Alias Pind.2 'nome di T1 Tasto2 Alias Pind.3 'nome di T2 Tasto3 Alias Pind.4 'nome di T3 Led Alias Portc.4 'nome del led in T1 Rele1 Alias Portc.3 'nome del relè1 Rele2 Alias Portc.2 'nome del relè2 Rele3 Alias Portc.1 'nome del relè3 Rele4 Alias Portc.0 'nome del relè4 Rele5 Alias Portd.7 'nome del relè5 Rele6 Alias Portd.6 'nome del relè6 In1 Alias Pinc.5 'nome dell'ingresso IN1 In2 Alias Pinc.6 'nome dell'ingresso IN2 In3 Alias Pinc.7 'nome dell'ingresso IN3 In4 Alias Pina.5 'nome dell'ingresso IN4 In5 Alias Pina.4 'nome dell'ingresso IN5 In6 Alias Pina.3 'nome dell'ingresso IN6 In7 Alias Pina.2 'nome dell'ingresso IN7 In8 Alias Pina.1 'nome dell'ingresso IN8</pre>	<p>Instrucciones para asignar nombres a las direcciones correspondientes a los terminales de los puertos del micro</p> <p>EJEMPLOS</p> <ul style="list-style-type: none"> - Rele1: Primer relé - In3: Tercera entrada TTL - Tasto3: Tercera tecla - Led: Diodo LED situado bajo T1
E	<pre>Config Adc = Single , Prescaler = Auto</pre>	<p>Instrucciones para configurar el conversor analógico-digital</p>
F	<pre>Attivo Alias 1 'definizioni utili per Disattivo Alias 0 'accendere e spegnere i relè Acceso Alias 0 'definizioni per accendere e Spento Alias 1 'spegnere il led su T1 Premuto Alias 0 'definizioni utili per leggere Rilasciato Alias 1 'lo stato dei pulsanti Alto Alias 1 'definizioni utili per leggere Basso Alias 0 'lo stato degli ingressi</pre>	<p>Estas instrucciones asignan los valores 0 (Dissativo, Acceso, Premuto y Baso) y 1 (Attivo, Spento, Rilasciato y Alto) a diversas etiquetas para hacer la programación más sencilla</p> <p>De esta forma se puede escribir</p> <p>Rele1=attivo Led=spento Tasto1=rilasciato</p> <p>en lugar de</p> <p>Rele1=1 Led=1 Tasto1=1</p>

G	<pre>Declare Sub Init_basicplc Declare Function Adc1() As Word Declare Function Adc2() As Word</pre>	<p>Definiciones del nombre de la subrutina y de las funciones de los bloques I, L y M</p>
H	<pre>Init_basicplc 'inicializza la scheda e salta Goto Go_basicplc 'al main program dell'utente</pre>	<p>Este bloque, junto al bloque N, es el código del programa ioNE.bas, que ejecuta la subrutina Init_basicplc, configura los puertos del micro en relación a las especificaciones de los terminales del Sistema Integrado y salta al final del programa</p> <p>Incluir el archivo ioNE.bas en un programa significa ejecutarlo y permitir el uso de las funciones y los nombres definidos</p>
I	<pre>Sub Init_basicplc() 'routine di inizializzazione Porta = 0 '0000 0000 Ddra = 0 '0000 0000 Portb = 192 '1100 0000 Ddrb = 63 '0011 1111 Portc = 16 '0001 0000 Ddrc = 31 '0001 1111 Portd = 28 '0001 1100 Ddrd = 226 '1110 0010 Start Adc Cls End Sub</pre>	<p>Instrucciones de inicialización de los puertos del microcontrolador</p>
L	<pre>Function Adc1() As Word 'ADC1 è collegato alla Local X As Word 'porta A pin 7 X = Getadc (7) Adc1 = X End Function</pre>	<p>Las instrucciones definidas en este bloque y en el siguiente (M) permiten una gestión sencilla de las entradas INADC1 e INADC2</p> <p>Si, por ejemplo, se desea leer el número digital correspondiente al valor analógico presente en la entrada INADC1, basta con escribir:</p> <p style="padding-left: 40px;">Dim valor 1 as word valore1 = adc1 ()</p> <p>El valor presente en la entrada ADC1 es pasado a binario por el ADC y su contenido se lleva a la variable valore1, definida como word (2 bytes) puesto que el ADC es de 10 bits</p>
M	<pre>Function Adc2() As Word 'ADC2 è collegato alla Local X As Word 'porta A pin 6 X = Getadc (6) Adc2 = X End Function</pre>	<p>Las instrucciones de este bloque, junto a las del bloque L, permiten una gestión sencilla de las entradas INADC1 e INADC2</p>
N	<pre>Go_basicplc:</pre>	<p>Ver bloque H</p>

Fig.11 Listado de las instrucciones en lenguaje BASIC del programa IONE.BAS. Este programa contiene toda la información sobre las conexiones del microcontrolador.

Para simplificar al máximo programación del sistema hemos incluido la información relativa a las conexiones del microcontrolador en el ámbito del **Sistema Integrado** dentro del archivo de librería **IONE.BAS**, que para ser utilizado debe ser incluido en el programa. Los ejemplos mostrados aclaran el procedimiento de uso e inclusión de la librería.

NOTA: El archivo **IONE.BAS** está incluido en el **CDR2107**. Al tratarse de un archivo desarrollado por Nueva Electrónica no se encuentra en el programa **BASCOM-AVR** descargado de Internet.

No obstante para quienes no adquieran el **CDR2107** tienen disponible el archivo **IONE.BAS** en la sección **ZONA DE DESCARGA** de nuestra página Web.

Por ejemplo, sabiendo que a los terminales **3** y **4** del conector **CONN.A** está conectado el contacto normalmente abierto del **RELÉ1**, para cerrar este contacto es suficiente escribir la siguiente instrucción BASIC:

```
rele1 = attivo
```

Si, por ejemplo, se desea que cuando se presione la tecla **T1** se cierre el contacto del **RELÉ1** basta con escribir las siguientes instrucciones:

```
if tasto1 = premuto then
rele1 = attivo
endif
```

Si se quiere obtener un **ciclo** de apertura y cierre del **RELÉ1** con un tiempo de **apertura** de **2 segundos** y un tiempo de **cierre** de **3 segundos**, ciclo que se **inicia** la pulsación de la tecla **T1** y **termina** con la pulsación de la tecla **T2**, basta con escribir las siguientes instrucciones:

```
inicio:
  if tasto1 = premuto then
    rele1 = attivo
    waitms 3000
    rele1 = disattivo
    waitms 2000
  end if
  if tasto2 = premuto then
    goto fin
  end if
  goto inicio
fin:
```

En la práctica nadie adoptaría el código anterior por las siguientes consideraciones:

- Precisa que se mantenga pulsada una tecla hasta que se ejecute el procedimiento que controla su estado.
- No se considera el hecho de que la pulsación de la tecla puede generar rebotes, como por ejemplo varias pulsaciones consecutivas de la misma tecla.
- No se tiene en cuenta la liberación del pulsador y, por lo tanto, no discrimina entre la voluntad de pulsarlo y la eventualidad de una pulsación accidental. Resumiendo, no tiene en cuenta el tiempo de presión del pulsador.

No obstante para **sencillos sistemas de control** puede servir una rutina simple como la anteriormente expuesta.

Recordamos que este artículo no está enfocado para enseñar programación BASIC sino mostrar las posibilidades del **Sistema Integrado**, aunque es interesante tener en cuenta las cuestiones anteriormente expuestas y analizar los programas de ejemplo que proponemos.

Para ilustrar los contenidos del archivo de librería **IONE.BAS** vamos a apoyarnos en el listado de sus instrucciones en lenguaje **BASIC** (ver Fig.11). En esta figura hemos dividido el código en **bloques**, cada uno identificado por una **letra**, y complementado con una breve **descripción**.

NOTA: Como se puede observar hemos respetado el **diseño original**, si bien se puede rediseñar según las necesidades de cada uno. Por ejemplo se pueden **traducir** a otro idioma los valores de variables y constantes, como **tasto1** por **tecla1** o **key1**, o bien **premuta** por **pulsado** o **pressed** (si se traduce en la librería al programar hay que utilizar las expresiones traducidas, en este caso **pulsador1=pulsado** o **switch1=pressed**).

PROGRAMAS EJEMPLO

Creemos que el modo más sencillo para aprender a programar el **Sistema Integrado** es trabajar sobre un **programa de ejemplo**. Vamos a utilizar **EXE1.BAS**. Analizaremos el código y como trasladarlo al Sistema integrado para ver lo que hace. Todo paso a paso.



Fig.12 Después de introducir el CD-ROM CDR2107 en el lector CD/DVD del ordenador hay que hacer doble click en el icono MI PC del escritorio, y, a continuación, y hacer doble click sobre el icono correspondiente al lector CD/DVD.

Fig.13 En la ventana que se abre se puede ver el contenido del CDR2107. Además de algunas carpetas con archivos, se encuentra el archivo PRESENTAZIONE.HTM, sobre el cual hay que hacer doble click.



Fig.14 Como sugiere la imagen para continuar hay que hacer doble click en el botón ENTRA.

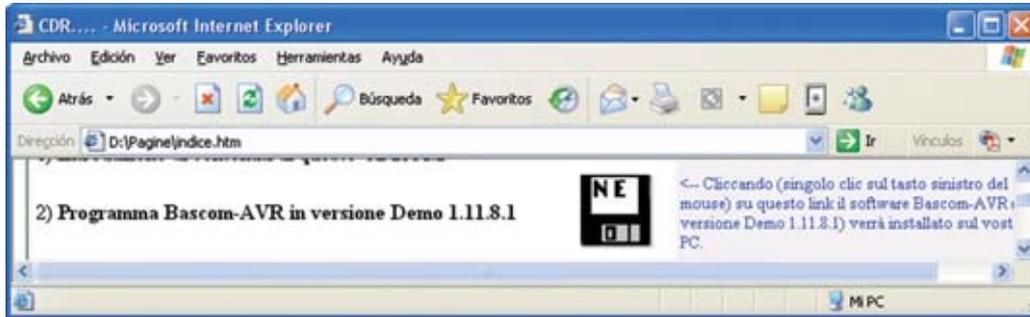


Fig.15 Como se explica en el artículo, el CD-ROM proporcionado junto al sistema KM2107 contiene, además del datasheet del microcontrolador ATmega8535L, cuatro ejemplos de programación en BASIC, dos aplicaciones prácticas y la versión Demo 1.11.8.1 del programa BASCOM-AVR. Para instalar el programa hay que hacer click en el icono con forma de disquete.

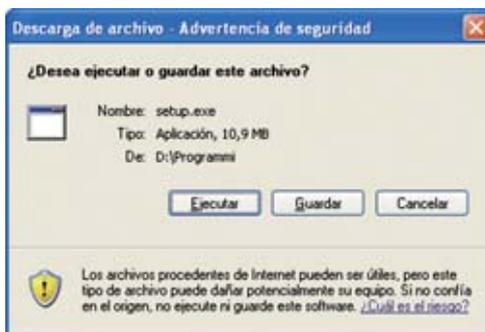


Fig.16 Como hacemos con la mayoría de nuestros CDs, también en este caso hemos simplificado lo máximo posible el procedimiento de instalación, que se inicia haciendo click en el botón EJECUTAR.



Fig.17 A Continuación hay que hacer click en OK. Durante el resto del proceso de instalación simplemente hay que pulsar en SI (YES) o en SIGUIENTE (NEXT).

Además de este programa el **CDR2107** dispone de más programas de ejemplo que vosotros mismos podéis analizar y utilizar para adquirir mas destreza con el **Sistema Integrado**.



Fig.18 El KM2107 incluye un cable de conexión al PC de un metro de longitud, mostrado en la fotografía de cabecera del artículo. Este cable está dotado con los conectores que se pueden apreciar en detalle en esta imagen.

Todos los programas de ejemplo están contenidos en la carpeta **Ejemplos** del **CDR2107**. No obstante también se instalan automáticamente en el directorio del programa **BASCOM-AVR** durante la instalación en vuestro ordenador.

Evidentemente antes de utilizar los programas de ejemplo es necesario explicar el procedimiento para instalar el programa **BASCOM-AVR** y la forma de realizar la conexión entre el ordenador y el **Sistema Integrado KM2107**.

INSTALACIÓN de BASCOM-AVR

El programa **BASCOM-AVR** se encuentra en el **CD-ROM CDR2107** incluido en el **Sistema Integrado KM2107**.

La instalación del programa está completamente guiada. En las Figs.12 a 17 se muestran las fases iniciales de la instalación. En las fa-

ses siguientes únicamente hay que hacer click en **ACEPTAR**, **OK** o en **SIGUIENTE (NEXT)**.

Es conveniente no modificar el **directorio propuesto** en la instalación ya que así será más fácil encontrar los programas de ejemplo que nosotros hemos desarrollado.

Las imágenes que acompañan la descripción de la instalación se han realizado en un ordenador **compatible IBM** con sistema operativo **Windows XP** y una tarjeta gráfica ajustada con una resolución de **800x600 pixels** con la combinación de colores estándar de Windows.

El **navegador** que hemos utilizado es **Internet Explorer**. No obstante el procedimiento de instalación es idéntico si se utiliza otro navegador.

Después de haber introducido el **CDR2107** en el lector CD/DVD de vuestro ordenador hay que hacer click en el icono **Mi PC** del escritorio de Windows.

En el caso de que no se realice automáticamente, depende de la configuración del ordenador, para visualizar el contenido del CD-ROM hay que hacer doble click sobre el icono correspondiente al lector en el que habéis introducido el **CDR2107** (en nuestro caso es **D:**). A continuación hay que hacer doble click sobre el archivo **presentazione.htm** (ver Fig.13). De esta forma se lanza el navegador y se visualiza la imagen le reproducida en la Fig.14.

Ahora hay que hacer click en **ENTRA** para acceder a la presentación del CD-ROM. Para instalar en el ordenador el programa **BASCOM-AVR** hay que hacer click en el icono con forma de **disquete** (ver Fig.15).

Con la imagen mostrada en la Fig.16 comienza el proceso de instalación. Para iniciarla hay que hacer click en **EJECUTAR**. El resto de ventanas solo han de confirmarse haciendo click en **ACEPTAR (OK)**. En la última ventana hay que pulsar en **FINISH**.

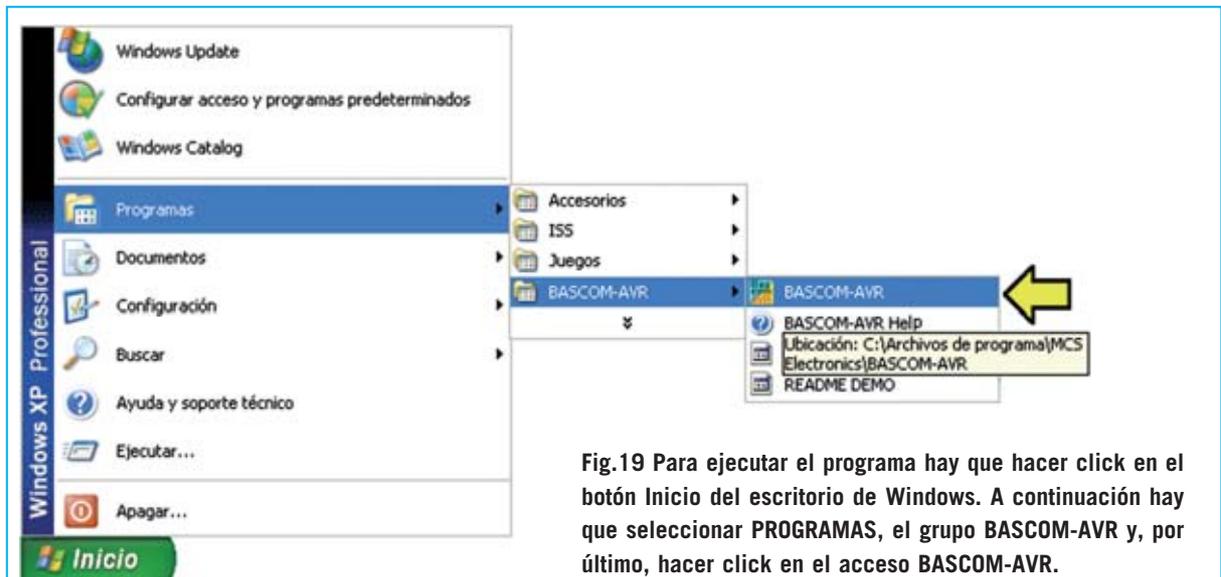


Fig.19 Para ejecutar el programa hay que hacer click en el botón Inicio del escritorio de Windows. A continuación hay que seleccionar PROGRAMAS, el grupo BASCOM-AVR y, por último, hacer click en el acceso BASCOM-AVR.

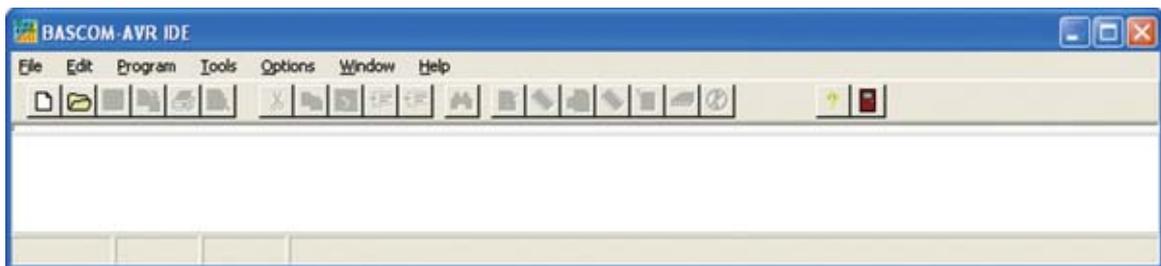


Fig.20 Aspecto de la ventana principal del programa BASCOM-AVR versión 1.11.8.1.



Fig.21 Cada vez que se ejecute el programa hay que ajustar la configuración haciendo click en la opción PROGRAMMER del menú OPTIONS.

de ejemplo en formato .BAS que también están disponibles en la carpeta **Ejemplos** del CDR2107, y que posteriormente analizamos.

CONEXIÓN del SISTEMA INTEGRADO al PC

Para conectar el Sistema Integrado **KM2107** al PC hay que utilizar el cable con sus correspondientes conectores incluido en el kit, conectándolo a **CONN.C** del Sistema Integrado y al **puerto paralelo** del ordenador (ver Fig.18).

El conector hembra del cable solo se puede conectar a **CONN.C** en un **único sentido**, por lo que es imposible equivocarse.

Llegado este punto el programa **BASCOM-AVR** está instalado en la carpeta:

C:\Archivos de Programa\MCS Electronics\BASCOM-AVR

En la misma carpeta también se ha instalado el archivo **IONE.BAS** y todos los programas

UTILIZACIÓN de BASCOM-AVR

Para lanzar el programa **BASCOM-AVR** hay que utilizar el botón **Inicio** de Windows siguiendo las indicaciones mostradas en la Fig.19. Una vez lanzado el programa muestra el aspecto que se evidencia en la imagen de la Fig.20.

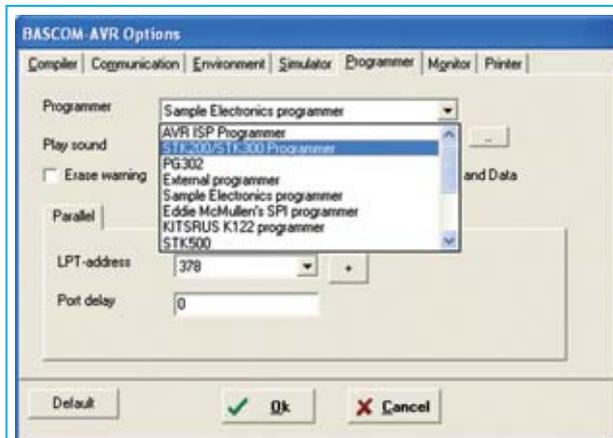


Fig.22 Hay que abrir la lista de selección PROGRAMMER.

Fig.23 En la lista de selección hay que elegir STK200/STK300 PROGRAMMER.



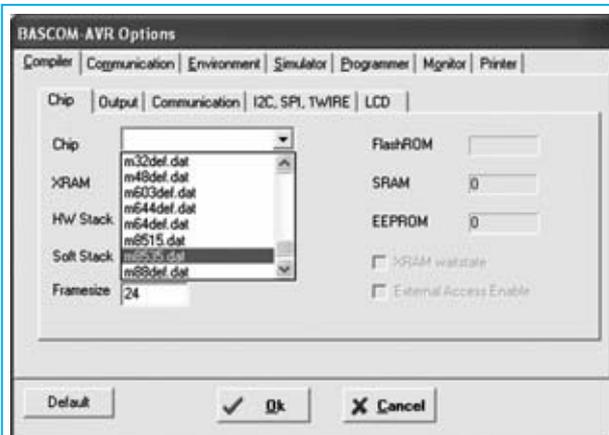
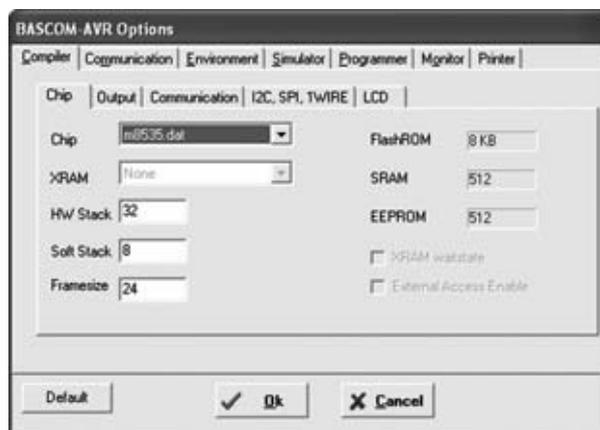


Fig.24 A continuación hay que seleccionar la pestaña COMPILER y abrir la lista de selección CHIP.

Fig.25 En la lista de selección hay que elegir M8535.DAT. Para confirmar los ajustes hay que hacer click en OK.



Para poder programar el **Sistema Integrado KM2107** hay que **configurar** necesariamente algunos parámetros de **BASCOM-AVR**, operación que ha de realizarse después de cada carga del programa.

zar dos ajustes (ver Figs.22-25): En la pestaña **Programmer** seleccionar el **Programador** (Programmer) **STK200/STK300** y el **puerto paralelo** utilizado, mientras que en la pestaña **Compiler** hay que seleccionar el **chip m8535.dat**.

Hay que seleccionar la opción **Programmer** del menú **Options** (ver Fig.21). Aquí hay que reali-

Para confirmar las selecciones efectuadas hay que hacer click en el botón **OK**. Ahora el pro-

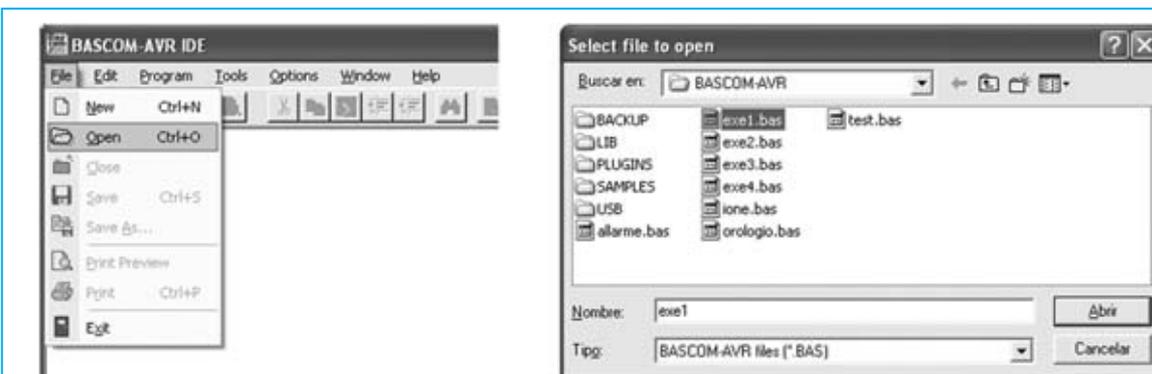


Fig.26 Para abrir uno de los programas que acompañan a BASCOM-AVR hay que seleccionar la función OPEN del menú FILE. Cuando se abre la ventana mostrada a la derecha hay que seleccionar el archivo .BAS (fuente escrito en BASIC). En el artículo analizamos el listado de EXE1.BAS, aunque en el CD-ROM hay más archivos.



Fig.27 Aspecto del archivo EXE1.BAS escrito en lenguaje BASIC para compilador BASCOM-AVR. La primera instrucción indica al compilador que incorpore el código fuente contenido en IONE.BAS, incluyendo así todas sus funcionalidades.

grama **BASCOM-AVR** está listo para la programación del **Sistema Integrado KM2107**.

PRIMER EJEMPLO (EXE1.BAS)

Para aprender a utilizar el programa **BASCOM-AVR** y poner a prueba las posibilidades del Sistema Integrado proponemos en primer lugar utilizar el programa de ejemplo **EXE1.BAS**.

Se trata de un programa escrito en **BASIC** para el compilador **BASCOM-AVR** y muy útil para comprender los métodos de control de los **relés, pulsadores, display LCD** y del **diodo LED** situado bajo la tecla **T1** del **Sistema Integrado KM2107**.

En primer lugar, desde la ventana principal de **BASCOM-AVR**, hay que seleccionar la función **Open** del menú **File** (ver Fig.26). En la ventana que se abre hay que seleccionar el archivo **EXE1.BAS** y pulsar en el botón **Abrir**. Auto-

máticamente se muestra una ventana similar a la mostrada en la Fig.27.

Para trasladar el programa al Sistema Integrado tiene que **compilarse** previamente, es



Fig.28 Para trasladar al micro un programa escrito en código fuente en primer lugar hay que compilarlo, haciendo click en la función **COMPILE** del menú **PROGRAM**.

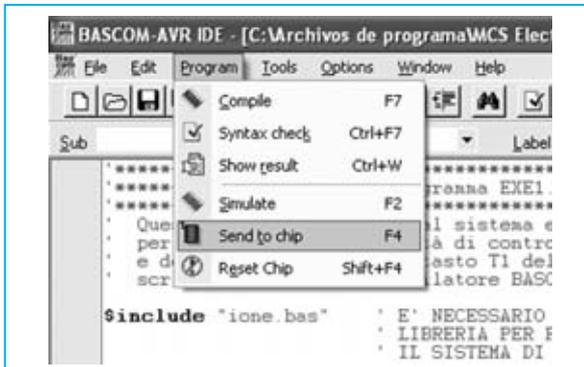


Fig.29 Si el proceso de compilación no indica ningún error se puede proceder a la programación del micro. Para realizar esta operación hay que seleccionar, en primer lugar, la función SEND TO CHIP del menú PROGRAM.

crea el archivo **EXE1.HEX** en lenguaje máquina para procesadores AVR.

Durante el proceso de **compilación** se ejecuta un módulo **depurador** (debug), a través del cual BASCOM-AVR genera, en su caso, un listado de errores, identificándolos adecuadamente.

Una vez compilado el programa, sin errores, hay que **volcarlo** al Sistema Integrado. Para realizar esta operación hay que seleccionar la función **Send to chip** del menú **Program** (ver Fig.29).

De forma automática se abre una nueva ventana en la que se puede observar el contenido de las diferentes **áreas de memoria** del microcontrolador.

decir traducirse al lenguaje máquina del microcontrolador. Para realizar esta operación hay que seleccionar la función **Compile** del menú **Program** (ver Fig.28). Automáticamente se

En el caso de que aparezca un mensaje de error del tipo **Could not identify chip...** significa que **no** hay comunicación entre el PC y el **Sistema Integrado**.

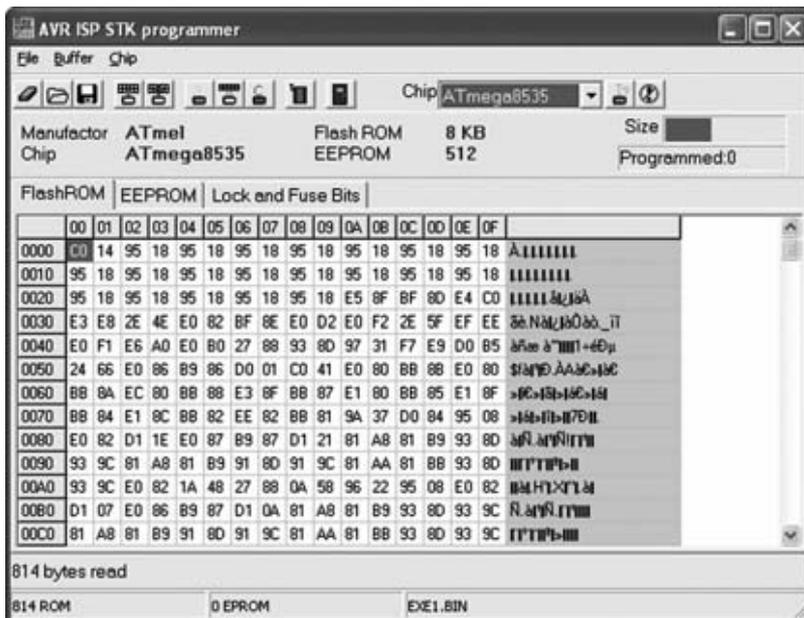


Fig.30 Al ejecutar la función SEND TO CHIP se abre una ventana con el contenido de las diferentes áreas de memoria.

Fig.31 Para iniciar el proceso de programación hay que seleccionar la función **AUTOPROGRAM** del menú **CHIP**.

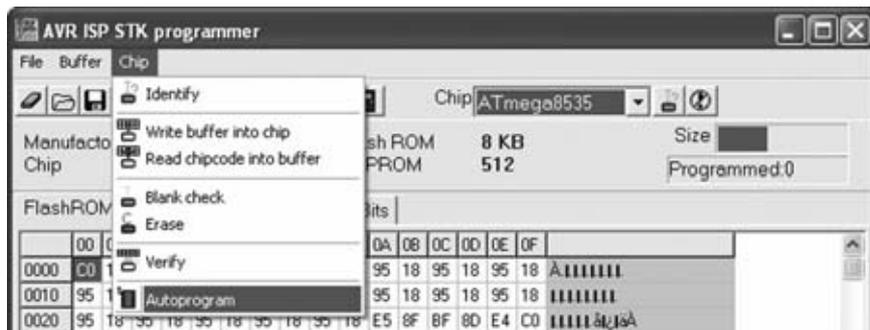




Fig.32 Una vez terminado el proceso de programación el sistema KM2107 inicia la ejecución del programa. En este caso aparece en el display una indicación con las funciones realizadas por las teclas.

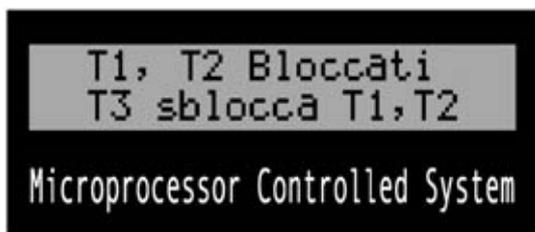


Fig.33 Mediante la pulsación de la tecla T3 el sistema valida, en caso de estar inhibido, el uso de T1-T2, o inhibe, en caso de estar validado, el uso de T1-T2.

El motivo seguramente sea que no se han realizado correctamente las **conexiones**, que habría que repasar, o que no se ha **configurado** adecuadamente, como se expuso con anterioridad, el programa BASCOM-AVR. También sería conveniente revisar el correcto funcionamiento del **puerto paralelo** del PC.

NOTA: Para revisar el correcto funcionamiento del puerto paralelo del PC se puede utilizar nuestro **Téster para puerto paralelo LX.1588** (revista N°241).

Para programar el microcontrolador, una vez situados en la ventana mostrada en la Fig.31, hay que seleccionar la función **Autoprogram** del menú **Chip**. De esta forma BASCOM-AVR inicia la programación del microcontrolador. Al terminar esta operación se ejecuta automáticamente el programa.

En este caso el **display LCD** muestra el contenido que se puede observar en la Fig.32, indicando las funciones realizadas por las teclas **T1** y **T2**.

Presionando **T1** se activa RELÉ1 y se enciende el diodo LED situado bajo T1, mientras que presionando **T2** se desactivan el RELÉ1 y el LED.

Presionando **T3** se bloquea, o desbloquea, la utilización de T1 y T2 (ver Fig.33).

Analizamos seguidamente el listado del programa (ver Fig.34) para ver como se realizan estas funciones.

Las primeras líneas, precedidas de una **comilla**, son **comentarios** que explican las funciones desarrolladas por el programa. De hecho el símbolo comilla (') indica que lo que hay a continuación no ha de procesarse, son comentarios.

La primera línea de código efectivo es:

\$include "ione.bas"

No se trata de una instrucción para el microcontrolador sino de una instrucción para el compilador que, de esta forma, añade en la cabecera del archivo **EXE1.BAS** el contenido del archivo **IONE.BAS**, incluyendo así todas las definiciones, variables, procedimientos y funciones contenidas en su interior.

La instrucción **Cursor off** hace que el cursor del display no sea visible. Obviamente puede volver a ser visible mediante la instrucción adecuada..

La sintaxis de las instrucciones puede ser consultada en el menú **Help** de BASCOM-AVR. La ayuda ofrecida es muy útil porque proporciona definiciones y sintaxis de todas las instrucciones BASIC fácilmente localizables en orden alfabético (**Bascom Language References**). Además se proporcionan útiles ejemplos de utilización de las instrucciones.

La siguiente línea (**Prg:**) se trata de una etiqueta (label), es decir una referencia introducida para volver a este punto del programa cuando interese, en este caso mediante la última instrucción del programa (**Goto Prg**). De esta forma se repiten indefinidamente las instrucciones contenidas entre la línea siguiente a la etiqueta **Prg:** y la penúltima línea.

Con **Cls** se limpia el display de su contenido en ese momento.

La instrucción **Lcd** muestra en el display el texto contenido entre las comillas, en este caso **Att. Di-**

```

'*****
'***** Programma EXE1.BAS *****
'*****
'Questo programma, adatto al sistema embedded KM2107, serve all'utente
'per comprendere le modalità di controllo di relè, tasti, display LCD e
'del led posto sotto il tasto T1 del sistema KM2107 tramite un software
'scritto in basic per compilatore BASCOM-AVR

$include "ione.bas"          ' E' NECESSARIO INCLUDERE IN OGNI PROGRAMMA QUESTA
                             ' LIBRERIA PER POTER UTILIZZARE IL PROGRAMMA CON
                             ' IL SISTEMA DI CONTROLLO EMBEDDED KM2107

Cursor Off

Prg:

Cls                          ' Pulizia display
Lcd "Att. Disat Block"       ' Invio di stringa al display
Locate 2 , 1                 ' Definizione di posizione
Lcd "Rele Rele T1,T2"       ' Invio di stringa
Wait 1                       ' Attesa di 1 secondo
Do
  If Tasto1 = Premuto Then   ' Se il tasto1 (T1) viene premuto
    Rele1 = Attivo          ' Il rele1 viene attivato
    Led = Acceso            ' e il led sotto T1 viene acceso
  End If                    ' ad indicare l'attivazione del rele
  If Tasto2 = Premuto Then
    Rele1 = Disattivo
    Led = Spento
  End If
  If Tasto3 = Premuto Then   ' ciclo
    Wait 1
    Cls
    Lcd "T1,T2 Bloccati "
    Locate 2 , 1
    Lcd "T3 sblocca T1,T2"   ' ciclo di attesa della pressione di T3
    Do
      Loop Until Tasto3 = Premuto
    Goto Prg
  End If
Loop
Goto Prg

```

Fig.34 Para seguir las explicaciones de las instrucciones contenidas en el archivo EXE1.BAS en esta figura hemos reproducido el listado original completo. Recordamos que el programa BASCOM-AVR dispone de una completísima información sobre la sintaxis de todas las instrucciones BASIC que incluye una colección de muchos ejemplos.

sat Block. Esta instrucción acepta tanto el texto de forma explícita entre comillas como una variable, cuyo contenido se visualiza en el display.

El texto indicado se visualiza en el display a partir de la primera posición (línea 1 - columna 1) o de la posición indicada por la instrucción **Locate** seguida del número de línea y del número de columna separados por una coma. La instrucción siguiente **Locate 2, 1** refleja precisamente lo anteriormente expuesto.

Después de las instrucciones **Lcd** y **Locate** en el display aparecen las indicaciones reproducidas en la Fig.32.

La instrucción **Wait 1** provoca que el Sistema Integrado no realice ninguna operación durante **1 segundo**. También existe la instrucción **waitms**, que suspende la ejecución del programa durante un tiempo expresado en milisegundos (**ms**).

A continuación se realiza un bucle sin fin correspondiente a las instrucciones contenidas entre las instrucciones **Do** y **Loop**. Estas instrucciones verifican el estado de las tres teclas.

Cada verificación es realizada entre las instrucciones **If** y **Then**: Hasta que las condiciones indicadas son verdaderas, y por lo tanto el resultado de la verificación es positivo, se eje-



Fig.35 Para escribir vuestros primeros programas os aconsejamos salvar con otro nombre un archivo fuente ya existente, modificarlo, y salvarlo con otro nombre a través de la función **SAVE AS...** del menú **FILE**. Cuando se abra la ventana mostrada a la derecha únicamente hay que modificar el nombre del archivo, dejando inalterado el valor de la carpeta destino para que el compilador encuentre el archivo **IONE.BAS** y no genere errores.



Fig.36 Este mensaje aparece cuando se utiliza la función **SAVE** del menú **FILE** y no se ha modificado el nombre del archivo originalmente abierto. En este caso hay que hacer click en **NO**.



Fig.37 Este mensaje aparece siempre que se guarde con un nombre diferente un archivo ya existente. El programa pide confirmación para abandonar el archivo original y así poder trabajar con el nuevo archivo. En este caso para continuar hay que seleccionar **YES**.

cuta el código correspondiente, es decir el incluido entre las instrucciones **Then** y **End if**.

EDICIÓN

Aconsejamos utilizar los ejemplos que hemos desarrollado como base para nuevos programas. Se pueden editar, hacer modificaciones y salvar con otro nombre para no escribir sobre el original. Esta operación se realiza mediante la función **Save as...** del menú **File** (ver Fig.35).

Es recomendable realizar las copias donde se encuentra instalado el programa, por defecto es:

C:\Archivos de programa\MCS Electronics\BASCOM-AVR

Sino se realiza así en la compilación del nuevo programa se producirán errores imputables al hecho de que el compilador no logra encontrar **IONE.BAS**.

Si se utiliza la función **Save As...** se abre una ventana que pregunta en inglés si se quiere abandonar el archivo origen para trabajar con el archivo recién renombrado. Hay que hacer click en **Yes** (ver Fig.37).

NOTA: Si apareciera un mensaje de **atención**, caracterizado por un triángulo amarillo con un signo de admiración, probablemente habréis olvidado escribir el nuevo nombre. En este caso hay que seleccionar **No** y dar un nombre al archivo.

En el nuevo código fuente hay que eliminar todas las partes del viejo programa que no sirven para nada, con la **excepción** de la instrucción **\$ include "ione.bas"**, que no hay que borrar nunca (a no ser que no se utilicen sus definiciones, variables y constantes).

Obviamente antes de terminar hay que **salvar** el **código fuente** escrito con la función **Save** o **Save As...** del menú **File** para almacenar su contenido.

CONTENIDO del CD-ROM

En el CD-ROM **CDR2107** y en la carpeta donde se encuentra el archivo **EXE1.BAS** hay varios ejemplos cuyo funcionamiento está descrito ampliamente en el propio CD-ROM.

En la misma carpeta también se encuentran **dos proyectos** realizados por nosotros y sugerencias para realizar aplicaciones para el Sistema Integrado.

En las próximas líneas se exponen de forma muy resumida estas aplicaciones. Los detalles también se encuentran en el CD-ROM **CDR2107** incluido en el **Sistema Integrado**.

RELOJ de PRECISIÓN con KM2107

Utilizando el **cuarzo de 8 MHz** incorporado en el **KM2107** es posible transformar el **Sistema Integrado** en un preciso reloj.

El programa en BASIC **OROLOGIO.BAS** además proporciona un válido punto de partida para realizar programas que precisen gestión de tiempo.

Por ejemplo se pueden realizar **temporizadores** o **despertadores** provistos de sensores.

SISTEMA de ALARMA con KM2107

Programando el **Sistema Integrado** con el programa BASIC **ALLARME.BAS**, incluido en el CD-ROM, el **KM2107** queda transformado en una **centralita de alarma**.

El programa posibilita la utilización de **numerosas funciones**:

- Gestión de **sensores perimétricos**, implementados con sensores magnéticos a colocar en puertas y ventanas.
- Gestión de **sensores de presencia**, implementados con sensores infrarrojos.
- Gestión de una **línea de retardo** que desactiva la alarma con el correspondiente código de desbloqueo.
- Gestión de una **línea auxiliar** para alimentar el sistema en el caso de que los intrusos corten los cables.
- Gestión de un **sistema de alimentación híbrido** que permite el funcionamiento de la instalación de alarma en caso de corte de suministro eléctrico.

- Posibilidad de efectuar **test periódicos** sobre el funcionamiento de la instalación sin molestar a los vecinos.

Los detalles sobre el esquema eléctrico, conexiones a los sensores, batería de respaldo, sirena y todas las instrucciones se encuentran disponibles en el CD-ROM.

NOTA A LOS LECTORES

Como habréis comprobado vosotros mismos el **Sistema Integrado KM2107** se adapta a numerosas situaciones y es fácilmente programable en lenguaje BASIC.

Estamos seguros de que este nuevo sistema de control tendrá muchos adeptos dadas sus características. Por esta razón estamos desarrollando una **sección propia** en nuestra **página Web** en la que publicaremos librerías, utilidades y proyectos desarrollados por nosotros y los más interesantes de los que vosotros nos enviéis.

En este espacio podréis obtener ideas y sugerencias para desarrollar vuestros **propios proyectos**. No obstante también publicaremos proyectos completos que consideremos interesantes en la edición impresa de revista.

PRECIO del SISTEMA INTEGRADO KM2107

KM2107: Precio del Sistema Integrado de Control con el programa **TEST.BAS** cargado para probar el funcionamiento de las teclas y las entradas/salidas (ver Figs.1-2), **montado** y equipado con el **cable de conexión al PC** (ver Fig.18), incluyendo el **CD-ROM CDR2107** que contiene, además del **artículo original** en formato PDF, el **datasheet** siempre del microcontrolador **ATmega8535L**, **4 ejemplos** de programación en BASIC, algunas aplicaciones y la versión Demo 1.11.8.1 de **BASCOM-AVR**386,80 €
LX.1348: Precio de los componentes necesarios para la realización del **alimentador estabilizado a 12 voltios** publicado en la **revista N°171**24,07 €
MTK07.03: Precio del mueble de plástico estándar para el alimentador **LX.1348** ..12,10 €
LX.1348: Circuito impreso8,02 €

ESTOS PRECIOS NO INCLUYEN I.V.A.