

# MANUAL DE USUARIO

## AUTÓMATAS SERIE P18



[www.CSR-ingenieria.com](http://www.CSR-ingenieria.com)



# ÍNDICE

|                                       |    |
|---------------------------------------|----|
| 1 – Introducción .....                | 2  |
| 2 – Especificaciones técnicas.....    | 3  |
| 3 – Descripción de los pines .....    | 5  |
| 4 – Lenguajes de programación .....   | 9  |
| 5 – Programación del autómata.....    | 14 |
| 6 – Comunicaciones inalámbricas ..... | 15 |
| 7 - INFORME DE CONFIDENCIALIDAD ..... | 18 |
| 8 - DIMENSIONES.....                  | 19 |

## **1 – Introducción**

La familia de autómatas de la serie P18 ha surgido para satisfacer las necesidades de aquellas personas que buscan realizar automatizaciones de todo tipo con la máxima libertad de programación.

El autómata consta básicamente de un microcontrolador central programable y unos periféricos auxiliares de entrada/salida (relés, transistores, etc.) que obedecen las órdenes de este para manejar dispositivos externos.

Para lograr la máxima flexibilidad de programación se ha escogido como microcontrolador uno de la familia de MICROCHIP, en concreto el PIC18F2550. Este combina las mejores prestaciones en comunicaciones (USB, I2C, EUSART, LIN, etc.) con toda la comunidad online con la que cuentan todos los productos de MICROCHIP.

De esta forma la familia P18 intenta atraer tanto a los profesionales que buscan automatizaciones de calidad a bajo costo como a los aficionados a los microcontroladores de MICROCHIP para que se animen a dar el salto desde los PICs de gama baja hasta uno con características profesionales facilitándoles un producto terminado con sus protecciones (carcasa, alimentación, fusibles, filtros EMI, etc.) y sus comunicaciones (USB, RS-232, RS-485, GPRS, RF, etc.) para que solo tengan que preocuparse de crear programas en el mismo entorno en el que venían haciéndolo con anterioridad (MPLAB).

**2 – Especificaciones técnicas**

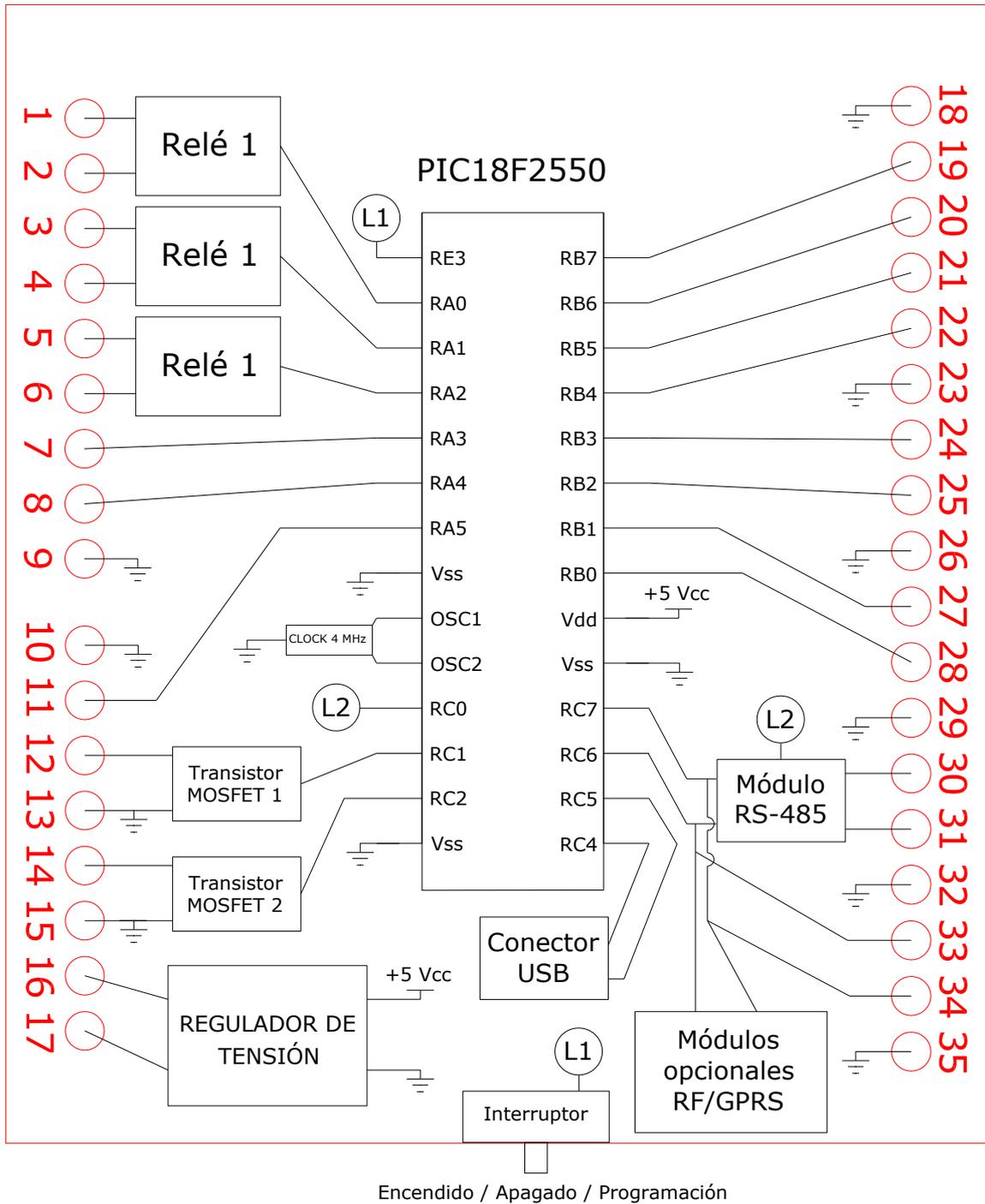


Figura 1 – Esquema general del autómata

El núcleo del autómata es un microcontrolador PIC 18F2550 con las siguientes características principales:

- Memoria de programa FLASH de 32 Kbytes
- Memoria RAM 2048 bytes
- Memoria EEPROM de 256 Kbytes
- 7 entradas/salidas digitales/analógicas de 10bits de resolución
- 2 salidas PWM de 10 bits de resolución
- 4 timers
- Comunicaciones EUSART, SPI, I2C, USB, LIN
- Watchdog timer programable

#### Nota

Para el óptimo conocimiento del microcontrolador debe consultarse el “datasheet” adjunto en la documentación del autómata.

Este microcontrolador interactúa con el resto de periféricos incluidos en el autómata:

- 3 relés con valores máximos de conmutación de 10 A y 250 Vca
- 2 transistores MOSFET con un valor máximo de conmutación de 10 A en Vcc
- Módulo RS-485
- Módulos opcionales para envío y recepción de datos mediante radiofrecuencia o mediante una red móvil GSM/GPRS (modelos P18-XX-RF y P18-XX-GS respectivamente)

La alimentación del autómata puede ser en tensión continua o alterna según el modelo.

A continuación se expone un cuadro con las características de cada modelo:

| Modelo    | Consumo nominal (W) | Consumo máximo (W) | Tensión de alimentación | Corriente de entrada máxima (A) |
|-----------|---------------------|--------------------|-------------------------|---------------------------------|
| P18-AC    | 0,125               | 1                  | 240 Vca                 | 0,004                           |
| P18-CC    | 0,125               | 1                  | 6-32 Vcc                | 0,166                           |
| P18-AC-RF | 0,355               | 1,23               | 240 Vca                 | 0,005                           |
| P18-CC-GS | 0,689               | 2,19               | 6-32 Vcc                | 0,365                           |
| P18-CC-RF | 0,355               | 1,23               | 6-32 Vcc                | 0,205                           |
| P18-AC-GS | 0,689               | 2,19               | 240 Vca                 | 0,009                           |

### 3 – Descripción de los pines

| Nº del autómata                  | Pines relacionados del microcontrolador | Funcionamiento  |
|----------------------------------|---|---|
| 1,2                              | RA0                                     | <u>Salida digital</u><br>Poniendo en alto el pin RA0 se activa la bobina del relé que cierra el circuito (normalmente abierto) entre los pines 1 y 2 lo que permite manejar cargas externas de hasta 10 amperios. |
| 3,4                              | RA1                                     | <u>Salida digital</u><br>Poniendo en alto el pin RA1 se activa la bobina del relé que cierra el circuito (normalmente abierto) entre los pines 1 y 2 lo que permite manejar cargas externas de hasta 10 amperios. |
| 5,6                              | RA2                                     | <u>Salida digital</u><br>Poniendo en alto el pin RA2 se activa la bobina del relé que cierra el circuito (normalmente abierto) entre los pines 1 y 2 lo que permite manejar cargas externas de hasta 10 amperios. |
| 7                                | RA3                                     | <u>Entrada/Salida digital (0-5 V)</u><br><u>Entrada analógica 3 (0-5 V)</u><br><u>Entrada analógica de referencia de voltaje para el convertor A/D (0-5 V)</u>  |
| 8                                | RA4                                     | <u>Entrada/Salida digital (0-5 V)</u><br><u>Entrada de reloj externo para el TIMER 0</u><br><u>Salida del comparador 1</u>  |
| 9, 10, 18, 23, 26,<br>29, 32, 35 | -                                       | <u>Salidas del neutro para conectar dispositivos externos (sensores, botones, etc.)</u>   |
| 11                               | RA5                                     | <u>Entrada/Salida digital (0-5 V)</u><br><u>Entrada analógica 4 (0-5 V)</u><br><u>Entrada de selección del esclavo SPI</u><br><u>Entrada de detección de voltaje alto/bajo</u><br><u>Salida del comparador 2</u>  |

(Continuación)

| Nº del autómata | Pines relacionados del microcontrolador | Funcionamiento   |
|-----------------|---|--|
| 12, 13          | RC1                                     | <u>Salida digital (0-5 V)</u><br>Poniendo en alto este pin se actúa sobre la puerta del MOSFET por lo que se comunican los puertos 12 y 13 (colector abierto con neutro en pin 13) para gestionar cargas externas de corriente continua de hasta 10 amperios.<br>Salida del comparador 2 y del módulo PWM 2            |
| 14, 15          | RC2                                     | <u>Salida digital (0-5 V)</u><br>Poniendo en alto este pin se actúa sobre la puerta del MOSFET por lo que se comunican los puertos 14 y 15 (colector abierto con neutro en pin 15) para gestionar cargas externas de corriente continua de hasta 10 amperios.<br>Salida del comparador 1 y del módulo PWM 1            |
| 16, 17          | -                                       | <u>Alimentación del dispositivo 16 (-) y 17 (+)</u><br>Los autómatas alimentados con tensión continua tienen el neutro conectado al neutro del autómata pero no así los alimentados con tensión alterna por lo que no debe usarse esta entrada como un neutro de conexión para cualquier uso salvo el de alimentación. |
| 19              | RB7                                     | <u>Entrada/Salida digital (0-5 V)</u><br>Puede configurarse para causar una interrupción en el programa por cambio de estado de entrada  |
| 20              | RB6                                     | <u>Entrada/Salida digital (0-5 V)</u><br>Puede configurarse para causar una interrupción en el programa por cambio de estado de entrada  |
| 21              | RB5                                     | <u>Entrada/Salida digital (0-5 V)</u><br>Puede configurarse para causar una interrupción en el programa por cambio de estado de entrada  |

(Continuación)

| Nº del autómata | Pines relacionados del microcontrolador | Funcionamiento  |
|-----------------|---|---|
| 22              | RB4                                     | <u>Entrada/Salida digital (0-5 V)</u><br>Puede configurarse para causar una interrupción en el programa por cambio de estado de entrada<br>Entrada analógica 11 (0-5 V)   |
| 24              | RB3                                     | <u>Entrada/Salida digital (0-5 V)</u><br>Puede configurarse para causar una interrupción en el programa por cambio de estado de entrada<br>Entrada analógica 9 (0-5 V)<br>Configurable como entrada de captura 2<br>Configurable como salida de comparación 2 y PWM 2 |
| 25              | RB2                                     | <u>Entrada/Salida digital (0-5 V)</u><br><u>Interrupción externa 2</u><br><u>Entrada analógica 8 (0-5 V)</u>  |
| 27              | RB1                                     | <u>Entrada/Salida digital (0-5 V)</u><br><u>Interrupción externa 1</u><br><u>Entrada analógica 10 (0-5 V)</u><br><u>Entrada/salida de reloj síncrono para comunicaciones SPI e I2C</u>  |
| 28              | RB0                                     | <u>Entrada/Salida digital (0-5 V)</u><br><u>Interrupción externa 0</u><br><u>Entrada analógica 12 (0-5 V)</u><br><u>Entrada de datos modo SPI</u><br><u>Entrada/salida de datos modo I2C</u>  |
| 30, 31          | RC0, RC6, RC7                           | <u>Entrada/salida de datos del modulo para comunicaciones RS-485</u><br>30 (B) y 31 (A)<br>El pin RC0 en estado alto activa el módulo   |
| 33, 34          | RC6, RC7                                | <u>Entrada/salida de datos para las comunicaciones RS-232</u><br>33 (Tx) y 34 (Rx)<br>Se puede conectar directamente al puerto serie de un PC configurando la salida de datos como invertida  |

(Continuación)

| Nº del autómata | Pines relacionados del microcontrolador | Funcionamiento  |
|-----------------|---|---|
| -               | RC4, RC5                                | <u>Entrada/salida de datos USB</u><br>RC4 (D-) y RC5 (D+)   |
| -               | OSC1, OSC2                              | <u>Entrada del reloj del sistema (no usar)</u>  |
| -               | RE3                                     | En el arranque con el interruptor en modo programación se pone en alto este pin para indicar al microcontrolador que entre en modo programación<br>En funcionamiento normal sirve para monitorizar la conexión/desconexión del autómata a un puerto USB para envío/recepción de datos |

Actuando sobre los pines del microcontrolador mediante el programa se activan las salidas del autómata asociadas según el esquema de la figura 1 y el cuadro anterior.

## **4 – Lenguajes de programación**

El autómata puede ser programado en tres lenguajes de programación:

### **El lenguaje BASIC:**

#### **Ventajas:**

- Es un lenguaje muy simple y con instrucciones fácilmente legibles, incluso por no expertos.

#### **Desventajas:**

- Nunca se va a tener el control total del programa en cuanto a tiempos de ejecución y control de registros bit a bit.
- Es muy complicado el manejo de interrupciones simultáneas en este lenguaje. Tiene limitaciones cuando genera el archivo .hex, es decir no optimiza el tamaño de memoria de programa del PIC.
- La mayoría de compiladores para este lenguaje pueden utilizarse únicamente bajo ambiente Windows.

Existen varias casas que producen compiladores para este lenguaje, entre ellas se pueden mencionar a:

[www.letbasic.com](http://www.letbasic.com)

[www.melabs.com](http://www.melabs.com)

[www.basicmicro.com](http://www.basicmicro.com)

## Lenguaje C

### Ventajas:

- Es un lenguaje de alto nivel más cercano a la máquina.
- Permite construir rutinas matemáticas fácilmente.
- Puede ser de ayuda al combinarlo con Ensamblador sobre todo en la gama alta.
- Se pueden crear macros con este lenguaje, para después simplificar el código en diferentes desarrollos.
- Es aceptado por la empresa fabricante Microchip, incluso ellos tienen algunos compiladores C.

### Desventajas:

- Los programas al compilarlos pueden resultar un poco extensos y pesados por ello debe tenerse en cuenta la capacidad de memoria de programa del PIC a utilizar.
- Con este lenguaje tampoco se puede controlar del todo los tiempos y los registros bit a bit.

Para este lenguaje existen también varias casas que producen software y compiladores para este lenguaje entre ellas las más importantes son:

[www.microchip.com](http://www.microchip.com): compiladores C18 para la gama alta y C30 para los dsPIC, ambos se pueden incluir en el entorno gratuito MPLAB IDE, y permite a los desarrolladores, llevar a cabo su aplicación en un lenguaje de alto nivel. No son del todo gratuitos.

[www.ccsinfo.com](http://www.ccsinfo.com): sus herramientas más conocidas, son la serie de compiladores para las distintas familias de microcontroladores Microchip PIC® MCU: PCW IDE, PCWH IDE y PCWHD IDE. Ofrecen la posibilidad de que el usuario pueda elegir el compilador concreto para la familia que va a utilizar, y compilar en modo línea de comandos, tanto para Microsoft Windows, como para Linux. Para los usuarios de Microsoft Windows, también existe la posibilidad de utilizar su potente entorno de desarrollo PCWHD IDE, que incluye además de los compiladores para todas las familias de PICs y dsPICs, entre otras cosas, un editor de código con reconocimiento de comandos, un depurador ICD, y un Wizard que en base a lo que el usuario le indique, generará la mayor parte del código para inicializar el microcontrolador. Además incluyen una gran cantidad de librerías muy útiles para asociar con la rutina que estamos desarrollando. Este compilador también puede integrarse con MPLAB IDE. No es gratuito.

[www.htsoft.com](http://www.htsoft.com) (HI-TECH): Sus herramientas más conocidas son PICC y PICC18. No son gratuitos.

## Lenguaje Ensamblador

### Ventajas:

- Es el lenguaje de bajo nivel natural de la línea PIC tanto para gama baja, media o alta.
- Con él se tiene un aprovechamiento eficiente de los recursos del PIC.
- Se pueden crear macros con este lenguaje, para después simplificar el código en diferentes desarrollos.
- Con él se pueden controlar los tiempos y los registros bit a bit.
- Excelente para manejar interrupciones simultáneas.
- Cuando se genera el archivo .hex éste es completamente optimizado.

### Desventajas:

- Si no se tiene experiencia en programación puede tardarse el desarrollo de alguna rutina en comparación con los otros lenguajes.

El compilador para este lenguaje es el MPLAB IDE el cual se consigue totalmente gratuito en la página de Microchip y constantemente están sacando nuevas versiones, debe aclararse que este solo funciona bajo ambiente Windows. Para Linux existe un versión alternativa llamada: PIKLAB la cual presenta simuladores, editores, ensambladores y soporta una gran cantidad de compiladores C para PIC.

### NOTA

Todos los archivos y ejemplos incluidos con este autómata están programados bajo el entorno MPLAB IDE propietario de la propia MICROCHIP y gratuito. No obstante cada usuario puede escoger el entorno que mejor se adapte a sus necesidades.

La mayoría de lenguajes se integran perfectamente en el entorno MPLAB e incluso permiten la posibilidad de realizar una programación híbrida definiendo la programación de las tareas más comunes en un lenguaje de alto nivel (C o BASIC) y a la vez insertando código en ENSAMBLADOR para aquellas tareas de bajo nivel y alta precisión como son el manejo de los “timers”, “flags”, etc. siendo esta la forma de trabajo más equilibrada entre tiempo de programación y control del PIC.

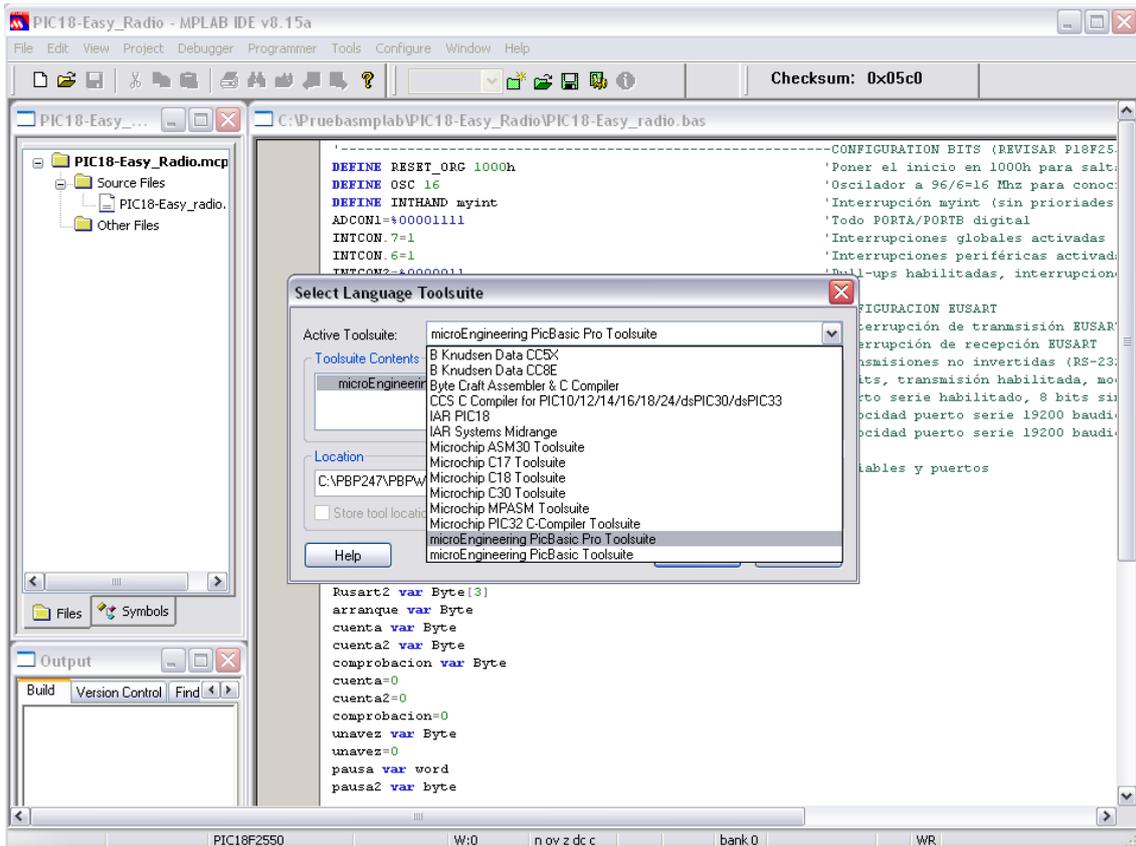


Figura 2 – Entorno de programación MPLAB IDE con ventana abierta de selección de compilador

Una vez diseñado y compilado nuestro programa en el ordenador tendremos un archivo “.hex” que debe ser introducido en el microcontrolador.

Para esto el microcontrolador viene con un BOOTLOADER embebido que permite la autoprogramación sin ningún hardware de apoyo adicional.

Este BOOTLOADER está localizado en el rango de memoria de programa que va desde la posición 0 hasta la 0FFF en base hexadecimal.

## MUY IMPORTANTE

Al principio de cualquier programa (sea el lenguaje y compilador que sea) debe especificarse que el programa empiece en la dirección 1000 (en hexadecimal) para saltarse el código del Bootloader.

Por seguridad, el Bootloader lleva programada una rutina que impide la escritura en este rango de memoria, no obstante debe prestarse especial cuidado a este hecho.

Ejemplo en MPLAB y lenguaje PICBASIC PRO:

```
DEFINE RESET_ORG 1000h
```

Si no se van a modificar los bits de configuración por defecto del PIC (no recomendable ya que podrían inutilizar el PIC) también debe indicarse a los lenguajes de alto nivel la velocidad del microcontrolador (por defecto 16 MHz) para que conviertan adecuadamente sus rutinas propias al código máquina.

Ejemplo en MPLAB y lenguaje PICBASIC PRO:

```
DEFINE OSC 16
```

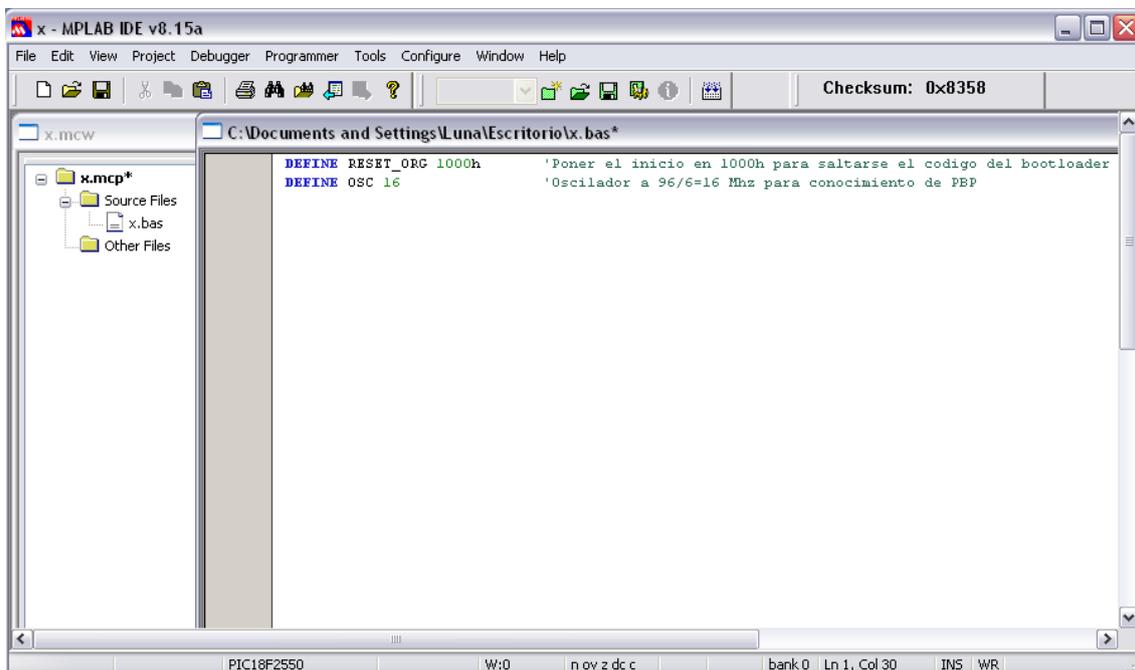


Figura 3 – Plantilla de inicio de un proyecto en MPLAB IDE con lenguaje PICBASIC PRO

## 5 – Programación del autómata

Una vez compilado el programa obtendremos un archivo “.hex” el cual debe ser introducido en el autómata.

Para esto se proporciona el programa CSR P18 Bootloader:

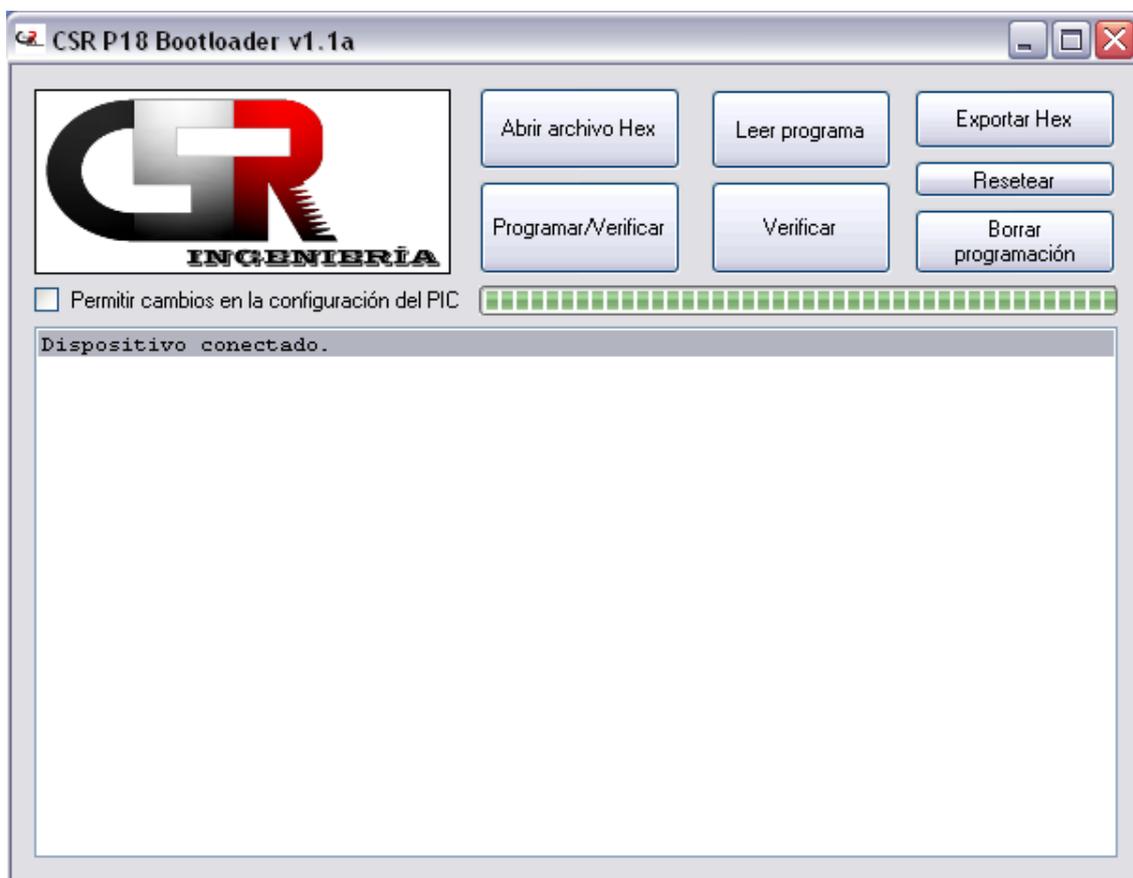


Figura 4 – Interfaz programa P18 Bootloader

Este sencillo programa se encarga de programar el autómata desde el ordenador mediante una interfaz muy simple.

Para la programación del autómata debe deslizarse el interruptor general a la posición de “PROG” (programación) y conectarlo a un ordenador mediante un cable USB. El ordenador detectará el autómata como un dispositivo HID (Human Interface Device) por lo que no necesita drivers adicionales.

Una vez detectado el autómata en el programa se selecciona el archivo “.hex” compilado y se pulsa el botón programar/verificar.

Si todo ha salido satisfactoriamente solo es necesario deslizar el interruptor a la posición de “ON” y el autómata empezará a ejecutar el programa.

## 6 – Comunicaciones inalámbricas

Además de las comunicaciones disponibles en el PIC (EUSART, I2C, SPI, LIN) el autómata incorpora unos módulos extra para comunicaciones:

### RS-485 (de serie en toda la gama)

Este modo de comunicación diferencial “Half-Duplex” permite enviar y recibir datos a distancias de más de 1 Km.

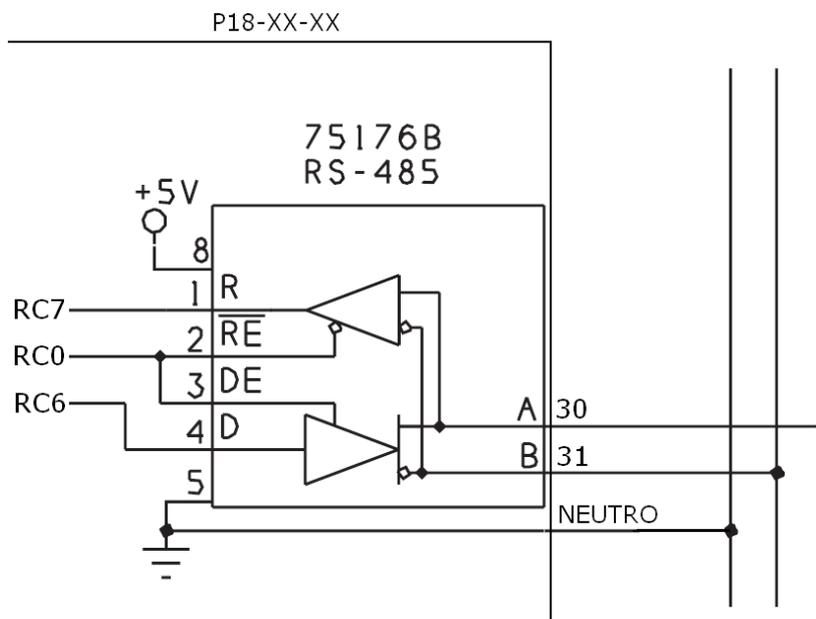


Figura 5 – Esquema de comunicación RS-485

Este sistema de interconexión puede soportar hasta 32 estaciones por si solo. Con ayuda de hardware especial puede llegar a las 256.

El integrado que realiza esta función es un S75176B, el cual se activa poniendo en alto el pin RC0 del autómata. Los datos se envían mediante EUSART de forma similar al interfaz RS-232 ya que ambos medios están conectados en paralelo (un dato enviado por RS-485 se presentará también en RS-232 por lo que una cuidadosa programación debe filtrar los datos según destinatario si se van a usar ambos medios de forma simultánea).

RF (disponible en los modelos P18-XX-RF)

El módulo de radiofrecuencia opcional permite enviar y recibir datos de forma inalámbrica.

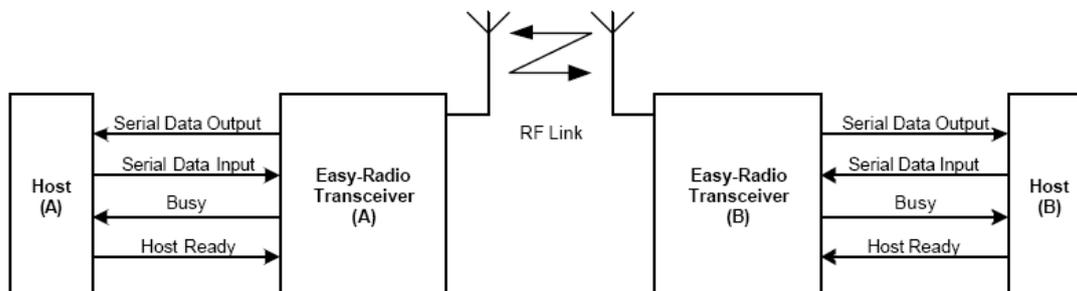


Figura 6 – Esquema de comunicación entre módulos

Este módulo es de la marca LPRS modelo ER900TRS. Puede consultarse la hoja de datos del mismo en la página del fabricante.

Para el envío/recepción de datos se usa la comunicación EUSART del PIC.

La configuración por defecto del modulo es:

- Velocidad de datos: 9600 baudios
- Frecuencia: 868 MHz (banda libre en Europa)

**NOTA**

Es posible cambiar la configuración del modulo en varios aspectos como la frecuencia de funcionamiento o la velocidad de los datos pero solo debe hacerse si se conoce a fondo el funcionamiento del mismo.

El autómata viene con una antena incorporada de 868 MHz y 1 dbi de ganancia. Si se desean cubrir distancias grandes pueden usarse antenas Yagi o similares con el conector SMA disponible en la carcasa.

GSM/GPRS (disponible en los modelos P18-XX-GS)

Este módulo GSM/GPRS cuatribanda permite acceder a una red de telefonía móvil para enviar y recibir datos.



Figura 7 – Módulo GPRS integrado

El módulo integrado es un MULTITECH MTSMC-G2-V-ED. Puede consultarse la hoja de datos del mismo en la documentación adjunta.

Para el envío/recepción de datos se usa la comunicación EUSART del PIC mediante el envío de comandos que empiezan por “AT”. Se puede encontrar una lista completa de estos comandos en la documentación adjunta.

El módulo tiene autodetección de velocidad de envío de datos desde el microcontrolador por lo que no es necesaria la configuración del mismo a una determinada velocidad.

Además del envío/recepción de llamadas y mensajes el módulo está capacitado para usar los siguientes protocolos de internet FTP, HTTP, POP3, SMTP, TCP (cliente y servidor) y UDP.

NOTA

El módulo requiere de una tarjeta SIM válida para acceder a la red móvil de la empresa suministradora de la misma.

Debe tenerse en cuenta la tarifa asociada a la tarjeta SIM por si resulta más interesante usar la red GSM (llamadas, mensajes) o GPRS (datos)

## **7 - INFORME DE CONFIDENCIALIDAD**

Lea atentamente las instrucciones y advertencias contenidas en el presente manual puesto que otorgan importantes indicaciones que preservan la seguridad del dispositivo, su uso y su mantenimiento. Cualquiera que instale, opere o realice el mantenimiento de este equipo debe tenerlas en cuenta. Ignorar estas instrucciones puede invalidar la garantía.

El presente Dossier técnico ha sido confeccionado para la empresa CSR ingeniería. Cada parte del Dossier ha sido creado al igual que el sistema de control para poder cumplir las necesidades de producción y calidad de CSR ingeniería.

Los nombres de empresas y productos mencionados en este manual son marcas registradas o marcas de las empresas respectivas.

La empresa CSR ingeniería no se hace responsable de:

-Cualquier daño material o físico al realizarse operaciones de mantenimiento o ampliación por parte de cualquier empresa no autorizada para realizar dichas operaciones.

-Ningún daño material ni físico si las medidas de seguridad descritas a lo largo de este manual no fuesen respetadas.

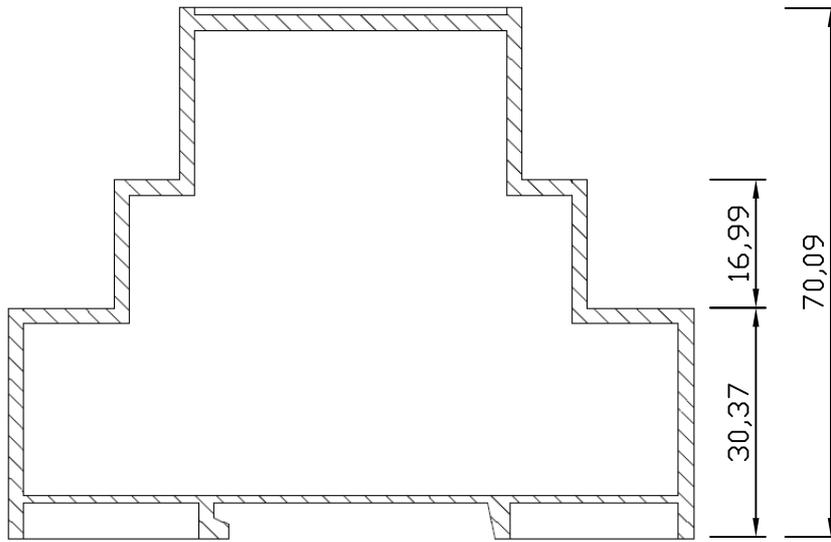
Ninguna parte de esta documentación puede ser reproducida, almacenada o transmitida, de ninguna forma ni por ningún medio electrónico, mecánico, mediante fotocopia, grabación o de cualquier otro tipo, sin el previo permiso por escrito de la empresa CSR ingeniería.



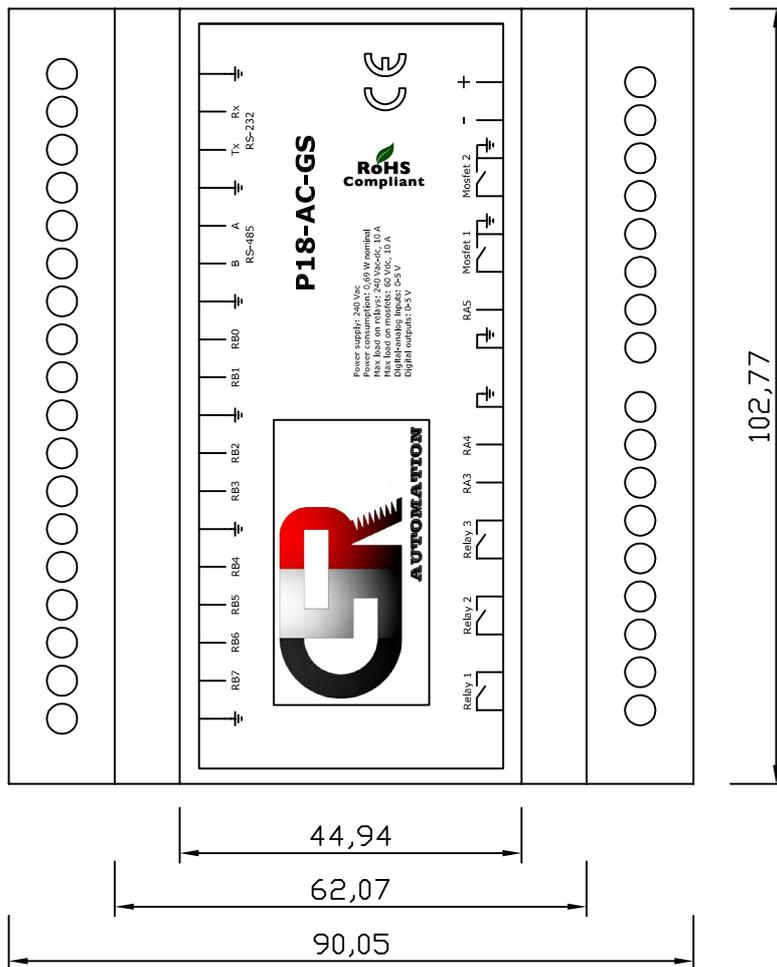
**www.CSR-ingenieria.com**

Av. Paradaseca, Nº10 – Tfno. y Fax: 987 54 04 49  
24500 Villafranca del Bierzo (León)  
Correo electrónico: [info@csr-ingenieria.com](mailto:info@csr-ingenieria.com)

### Corte de sección en Vista frontal



### Vista superior



Dimensiones (E= 1:1)