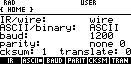
**CONFIGURACIÓN PARA HP**

Observa la siguiente pantalla. Tienes que tener la misma configuración para lograr la comunicación con el PIC16F84.



|  |  |
| --- | --- |
| **PORT:** | Selecciona el puerto de comunicaciones. |
| **FMT:** | Seleccione el formato de transferencias, o bien ASCII o bien Binary. |
| **BAUD:** | Introdusca la velocidad de transferencia. La definición debe coincidir con la del ordenador. |
| **PARITY:** | Introduzca la paridad de transferencia. La definición debe coincidir con la del ordenador. |
| **CHK:** | Introduzca uno de los tres protocolos de detección de errores. |

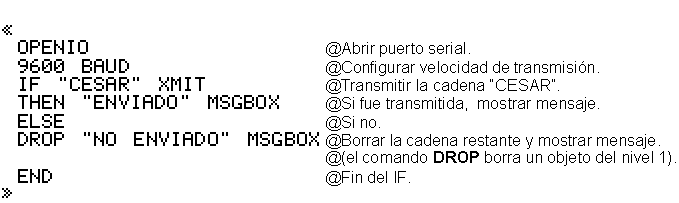
**Comandos:**  
Los comandos para la transmisión serial, se muestran en la siguiente figura:



|  |  |
| --- | --- |
| **XMIT:** | Envía la cadena del nivel 1 sin protocolo Kermit. Una vez enviada toda la cadena, se devuelve 1 al nivel 1. |
| **SRECV:** | Recibe los caracteres especificados en el nivel 1. |
| **STIME:** | Ajusta la espera de transmisión/recepción serie al número de segundos especificados en el nivel 1. |
| **SBRK:** | Envía una señal BREAK serie. |
| **BUFLEN:** | Devuelve el número de caracteres de la memoria intermedia de entrada al nivel 2, y el estado de errores al nivel 1. |

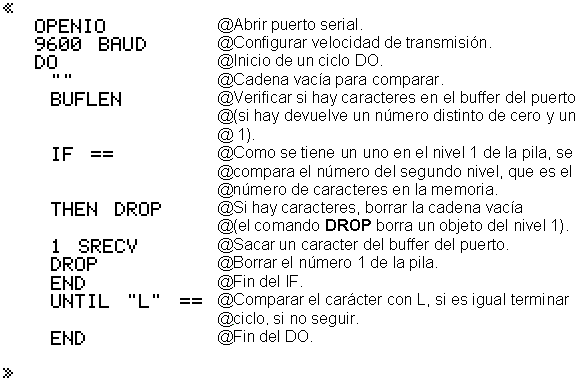
**DETALLE DE COMANDOS PARA COMUNICACIÓN SERIAL Y EJEMPLOS MUY UTILES.**

**Comando** **OPENIO**. Abre el puerto serial. Este comando es útil cuando se quiere iniciar una comunicación serial dentro de un programa. El comando BAUD se puede utilizar para configurar la velocidad de transmisión dentro de un programa .

**Comando** **XMIT**. Envía una cadena del nivel 1 de la pila. Una vez enviada toda la cadena, se devuelve 1 al nivel 1. Si no se lograra transmitirse toda la cadena, se devuelve 0 al nivel 1 y se devuelve la parte no enviada de la cadena de entrada al nivel 2. A continuación se presenta un programa de ejemplo:  
  
  
  
El programa transmite una cadena de caracteres a una velocidad de 9600 bauds, el comando XMIT es utilizado como cláusula-prueba, si se transmite la cadena completa se muestra un mensaje en un recuadro, pero si no se transmite la cadena completa, se borra la parte que no se envío y se muestra un mensaje en un recuadro. De esta forma se puede controlar la transmisión de datos por el puerto serial.

**Comando** **BUFLEN**. Devuelve el número de caracteres de la memoria intermedia de entrada al nivel 2 de la pila, y el estado de errores al nivel 1 (1=ningún error de comunicación ni de desbordamiento del UART, o 0=error de comunicación o de desbordamiento del UART). Si devuelve 0 al nivel 1 de la pila, el número de caracteres devuelto al nivel 2 representa la parte de los datos recibidos antes del error.

**Comando** **SRECV**. Recibe los caracteres especificados en el nivel 1 de la pila. En una transferencia correcta, se devuelven los caracteres al nivel 2 de la pila como cadena, y se devuelve 1 al nivel 1. En una transferencia incorrecta, se devuelve una cadena vacía o incompleta al nivel 2 de la pila y se devuelve 0 al nivel 1.

A continuación se muestra un programa de ejemplo:  
  
  
El programa primero inicializa y configura el puerto serial, después entra a un bucle DO dentro del cual se verifica el número de caracteres recibidos y si ese número es igual que uno se recibe un caracter y se compara con el carácter “L” y si no es igual se vuelve a verificar si hay datos en la memoria del puerto.

Observe que se tiene un IF...THEN...END dentro de un DO...UNTIL...END. el IF junto con los comandos, forman la cláusula-bucle y la comparación con “L” la cláusula-prueba.

Este programa permanece en un ciclo infinito hasta que se reciba un caracter “L”.