

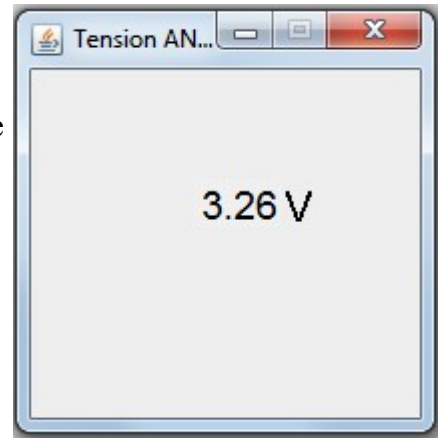
PIC Manejando un Socket en Java

Vamos a ver la creación de pequeño Socket en Java que recibe los datos de nuestro PIC. La aplicación es muy simple dejando al alumno la posibilidad de reformarla según crea conveniente de acuerdo a sus necesidades.

Como podemos ver se trata de una simple ventana Java que recibe el voltaje del canal AN0, encontrará dentro del archivo

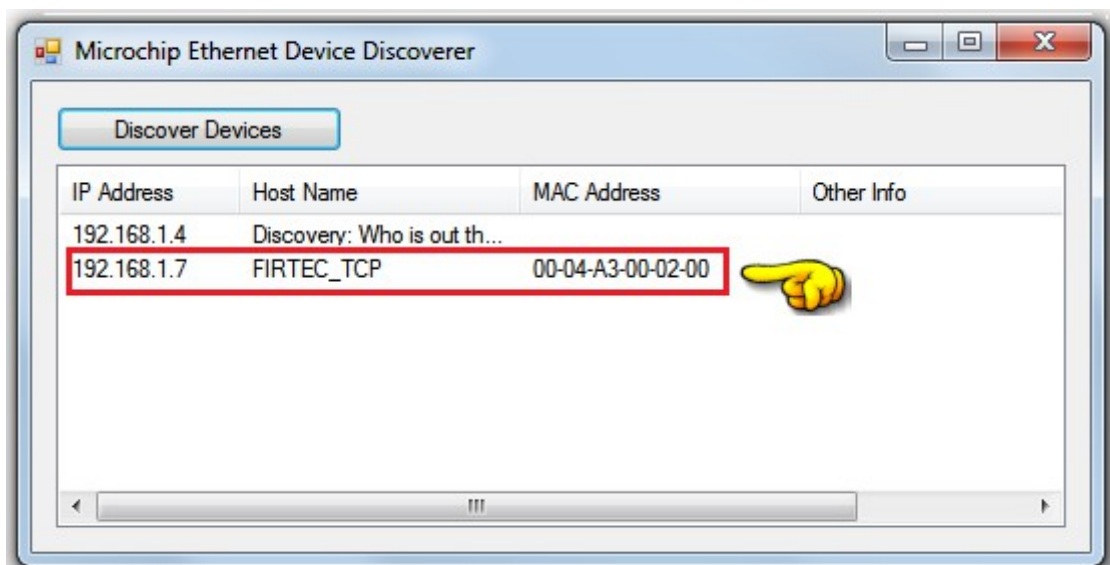
Proyecto_Java.zip todo lo necesario para compilar el proyecto directamente en el MPLAB, también encontrará las fuentes en Java para *NetBeans*.

El voltaje se mostrará tanto en la ventana Java como en el LCD.



En el proyecto para el PIC se han alterado los archivos **main.c**, **config.h** y **ftpd.c** para que el funcionamiento sea correcto.

Para compilar el código usted decidirá si usa **DHCP** o no dependiendo si se conecta por medio de un router, para saber exactamente cual es la dirección IP que su router le asignó es muy útil el uso de la herramienta **MCHPDiscover.exe** que nos revela la IP que tenemos asignada.



Observe también que podemos ver la dirección MAC oportunamente configurada en **config.h**.

```
#define DEFAULT_MAC_ADDRESS { 0x00, 0x04, 0xA3, 0x00, 0x02, 0x00 }
```

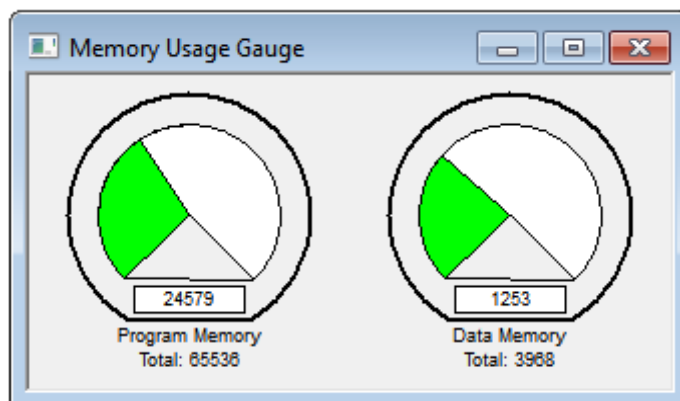
Esta forma de trabajo es muy interesante cuando no necesitamos de un servidor web sino solo transmitir datos mediante Sockets incluso a sitios remotos o programas remotos, el manejo de Sockets es practica común en la mayoría de los lenguajes de programación como *Visual Basic*, *Delphi*, *Java*, *C++*, etc, pudiendo usar cualquiera ya que el manejo del Socket es transparente al PIC, este solo envía datos a la dirección IP por el puerto definido el tratamiento de los datos puede ser cualquiera, completar formularios en un sitio web remoto, trabajar con una base de datos o simplemente monitoreo continuo de variables remotas.

Si fuera necesario podemos abrir mas puertos o secuenciar el envío de datos por un único puerto estableciendo un protocolo para la interpretación de estos datos.

Si miramos dentro del código java de la ventana principal podemos ver lo siguiente:

```
try{  
    s=new Socket("192.168.1.7",21);  
    System.out.println("Socket IP: "+(s.getInetAddress()).toString());  
  
    out=new PrintWriter(s.getOutputStream());  
    in=s.getInputStream();  
    .
```

El Socket abre una conexión en la dirección asignada por el puerto 21 que es el mismo que se usaba para el FTP es por eso que hemos reformado el archivo *ftpd.c* y re-utilizar el puerto 21.



El recuento de memoria ha bajado considerablemente pudiendo colocar este proyecto en cualquier micro 18Fxx con el requerimiento adecuado de memoria y velocidad. En cuanto a la electrónica empleada es la misma que en los trabajos anteriores sin embargo ya no necesitamos disponer de memoria exterior, al menos para contener el sitio web.



Todos los derechos reservados
© by Firtec Capacitación