

Tarjeta USB para

Lo que aquí mostramos, es una pequeña tarjeta USB con un gran número de aplicaciones, pudiendo ser programada por cualquier tipo de lenguaje, C, Java, Visual Basic 6. Nosotros solamente describimos algunas aplicaciones, pero sin ninguna duda, es posible cubrir todas vuestras pretensiones.

Se puede decir que hoy en día es prácticamente imposible que no haya un trabajo, en el cual no se utilice un ordenador.

Desde el artesano al estudiante o desde el más joven a los adultos, no hay nadie que hoy en día prescindiera del ordenador. Debido al gran número de peticiones que nos han hecho llegar nuestros lectores, sobre el desarrollo de aplicaciones con el PC, hemos creado una tarjeta USB con el intento de jubilar la vieja interface serie/paralelo multiuso **LX.1127** (publicada en la revista N. 118).

Con la creación de esta tarjeta **USB** en **SMD**, podremos tener tantos medidores como queramos, utilizar en una sola interface acoplada al ordenador.

NUESTRA TARJETA

Como ya hemos comentado, suministramos la tarjeta **KM1374K** (en la parte central de la Fig.1) con todos los componentes en **SMD**, lista para su funcionamiento. Esta tarjeta se creó gracias a las sugerencias que nos realizaban nuestros lectores, que no encontraban en tiendas estas pequeñas tarjetas con un sistema abierto a cualquier lenguaje de programación. Como podemos apreciar en la Fig.4, nuestra tarjeta, en concreto; tiene una superficie de pocos centímetros cuadrados y en ella hay montado un microcontrolador programado por **Microchip**, el **PIC18F4553 I/P**. Se trata de un microcontrolador dotado internamente con un puerto USB de configuración "**bridge**", es decir

con conversión **USB-RS232**, que hemos utilizado para materializar el interfaz multivalente.

Para esta aplicación hemos utilizado del microprocesador solo algunas partes (en realidad este componente tiene una capacidad muy superior), con el objetivo de crear **cuatro** instrumentos que nos permitan una enseñanza tanto didáctica como profesional.

Observando el dibujo de la Fig.6, la tarjeta se sirve del puerto **USB** (ver **JP5**) para comunicarse con el **PC**, y una entrada analógica-digital de 12 bit para digitalizar la señal procedente de cualquier sensor valido (ver en la Fig.6, el primer pin a la izquierda del conector **JP3**).

La tarjeta fue creada para que pudiese ser transformada en un conductivímetro profesional. Para ello debe estar completada por diferentes tensiones, que sirven también para alimentar el modulo de **0 a 500 $\mu\text{S/cm}$ (microsiemens/cm)**, que ya conocéis de un número anterior **el conductivímetro profesional LX. 1697/11** publicado en la revista **N.275**.

Por lo tanto en la tarjeta hay **5 Voltios** para la alimentación de los sensores externos, **9 Voltios** para la alimentación de la tarjeta **KM1697/2**, **3,3 Voltios** para la alimentación del micro, y todo ello a partir de un sencillo alimentador externo de **12 Voltios**.

Solamente en el caso de utilizar la tarjeta USB como conductivímetro, **no** debemos emplear el **diodo de silicio** como se indica en la Fig. 5, de esta manera, podremos utilizar los **5 Voltios** suministrados de la toma **USB** del ordenador para la alimentación de la tarjeta o del sensor, y no será necesario de acudir a ningún otro sistema de alimentación externa.

1.000 APLICACIONES

El diodo también realiza la función de **protector** por si se producen posibles retornos de tensión hacia el PC.

Además, en la tarjeta también hay previsto un diodo led que sirve para **indicarnos** si nuestra tarjeta ha sido reconocida por el PC.

Multiples Aplicaciones



Fig.1 En la foto se puede apreciar al centro la tarjeta USB KM1734K, y alrededor algunos de los accesorios que utilizaremos para nuestras aplicaciones.

Cada vez que introduzcamos **el USB**, el parpadeo de una luz nos indicará que todo funciona perfectamente.

Obviamente, el **diodo led** no es indispensable, pero sin embargo realiza una función muy práctica. Si apreciamos que los datos no llegan al PC, lo que debemos hacer es re-introducir el conector USB de nuevo, y si comenzará el parpadeo del LED que nos indica que todo va bien.

Dentro del kit hemos incluido algunos componentes necesarios para transformar el circuito en una "tarjeta universal":

Un **conector hembra** (ver **JP3** en la Fig.6) necesario para alojar al módulo **KM1697/1**, en el caso que deseemos utilizar la tarjeta como **conductímetro**.

Una **toma jack de 3mm estereo** (ver **JP6** en la Fig.6) para ser utilizada como entrada de los sensores en diferentes aplicaciones, de las cuales hablaremos más adelante.

Nota: Hemos utilizado un **jack de 3mm estereo** porque **dos** de los tres polos sirven para la alimentación, mientras que el otro para sirve recibir la señal a digitalizar.

Un diodo **led rojo** (ver. DL7 y R1 en la Fig.7) que se deberá conectar al conector **JP1** en la salida reservada a esta función.

Un diodo de silicio para "desviar" en su sentido más literal, la alimentación de los **5 Voltios** que vienen de la toma del **USB** del **PC** (en un punto concreto de la tarjeta - ver Fig. 5), y alimentar a todos los componentes sin tener que recurrir a una alimentación externa al PC, además, como indicamos anteriormente, de proteger al PC de retornos de tensión.

Nota: también está previsto un **módulo** para reemplazar la **temperatura ambiente KM1734 KT** (ver Fig. 7), la cual **suministramos solo a quien la solicite**.

REALIZACION PRACTICA

Cuando tengamos a nuestra disposición la tarjeta **KM1734K**, debemos decidir que instrumento prepararemos.

No habrá ningún problema si por algún motivo decidimos adquirir tantas tarjetas como instrumentos que se necesitan. Solo es necesario disponer de un soldador y un poco de estaño para realizar las pequeñas "uniones" necesarias para adaptar el hardware a la aplicación deseada.

Por tanto, como hemos especificado en la Fig.7, conectaremos la resistencia **R1** al terminal **+** (**A**) del diodo led y el otro terminal de la resistencia y el cátodo (**K**) del diodo led a los taladros indicados del conector **JP1**. Introduciremos el conector de **11** pines en los taladros de **JP3** y soldaremos sus terminales por el lado contrario.

Si empleamos la tarjeta únicamente para crear el conductímetro ya hemos concluido. Si decidimos emplearla para otras aplicaciones, soldaremos el diodo de silicio y continuaremos con el montaje de forma que podamos pasar con rapidez de un instrumento a otro. Primeramente, uniremos un conector de tres pines hembra y la **toma jack de 3mm estereo** que servirá de entrada para los sensores de las diferentes aplicaciones.

Siguiendo el ejemplo de la Fig.7, conectaremos el terminal **de masa** de la toma jack al pin de la derecha del conector, y la terminal de **5 Voltios** al pin de la izquierda, dejando libre el pin central. Después, conectaremos el terminal **S** en el primer orificio del conector **JP3** que corresponde a la **entrada analógica**.

Ahora ya estamos listos para decidir cual será el primer instrumento, eligiendo uno y el software que se muestra en el CD-ROM que adjuntamos. Aconsejamos introducir la tarjeta en una caja apropiada solo cuando tengamos decidido cual será la aplicación.

Recordamos que es importante decidir si utilizamos la tarjeta **LX.1734** (ver Fig.7) para crear un **conductímetro**, en el que hay que recurrir a una alimentación externa (el USB no tiene capacidad para suministrar tanta corriente), o un **gaussímetro**, o un **detector de rayos UVA/B**, o **termómetro con termopila**, que como consumen pocos miliAmperios, no tienen que alimentarse desde el exterior. Si elegimos una de estas tres soluciones, debemos conectar el **diodo** de silicio en la parte posterior de la tarjeta, como se ha diseñado en la Fig.5. Antes de pasar a la descripción de los programas de este proyecto, nos paramos en la descripción de otras operaciones preliminares, necesarias para su utilización. Estas ope-

...a propósito del puerto USB

El puerto **USB** (Universal Serial Bus) permite una comunicación serie estandar que para conectar diferentes periféricos a un ordenador.

La toma **USB** está proyectada para que un mayor número de periféricas se puedan conectar usando una sola interfaz estandarizada y un solo tipo de conector, y poderlos conectar/desconectar sin tener que reiniciar el ordenador.

Este tipo de conexión solo se da en ordenadores que puedan soportar un **USB** en su sistema operativo y en la parte electrónica.

Por ejemplo, quien tiene el sistema operativo DOS no puede utilizar el USB para comunicarse con los periféricos.

Solo con la **segunda edición de Windows 95**, y por supuesto con una electrónica adecuada se ha podido conseguir por primera vez.

Actualmente con los nuevos operadores con **WindowsXP, Vista, Linux, MAC OS**, etc, este tipo de conexión se ha generalizado a la hora de emplear los distintos tipos de periféricos: impresoras, ratones, escáneres, pen drives, cámaras de fotos, altavoces, y naturalmente tarjetas electrónicas como nuestras LX.1690 (osciloscopio/analizador de espectro), o el LX.1666 (convertidor de audio **USB RISS** para copiar vinilos en el PC o CD-ROM).

Para el funcionamiento del puerto **USB** en el ordenador se necesitan instalar los **drivers**.

Estos drivers son pequeños programas que se instalan en el PC, para indicar a éste como reconocer emplear el aparato dotado de USB.

En nuestro caso, los drivers que suministramos son compatibles con **Microsoft**, o sea **WindowsXP y Vista**. Otros sistemas operativos necesitan sistemas específicos que los suministra directamente la casa constructora del periférico.

En el caso de periféricos tradicionales como impresoras, cámaras de fotos o pen drivers, los drivers se encuentran en el sistema operativo en el 99% de los casos (nos referimos naturalmente a sistemas operativos de última generación como WindowsXP, Vista, Linux, MAC OS, etc.).

Si compramos un ratón y lo insertamos en su correspondiente toma USB, el ordenador lo recono-

cerá en poco segundos, y podremos utilizarlo sin tener que apagar y encender como se hacia antes.

En los USB hay dos tipos de conectores para enlazar los dispositivos, el **conector A** y el **conector B**. En los últimos años algunos productores han variado los diferentes tipos de conectores en sus dispositivos, creando conectores más pequeños de los estándar.

En realidad, estos dispositivos respetan a los estándar a todos los efectos, la única diferencia es el cambio físico del conector (se denominan **micro USB** o **mini USB** de tipo **A** o **B**).

Desde el punto de vista técnico la toma se forma por **4 pines** divididos, según forma y tamaño, en dos terminales serie **D+** y **D-** que sirven para el intercambio de datos, y en dos terminales donde uno suministra **5 Voltios** del ordenador y en el otro representa la **masa** general.

Estos 5 Voltios que dispone el USB son indispensables para alimentar al periférico conectado, que hace independiente al puerto de cualquier alimentación externa.

Naturalmente, no se podrán conectar los que requieran grandes consumos, ya que el límite está establecido en los **500 mA**.

En el caso de que los periféricos USB necesiten una corriente superior al límite, se debe introducir una alimentación externa adecuada.

La trasmisión de datos en el viejo serie **RS232** se realizaba sobre dos líneas específicas, es decir

RS-232 (EIA/TIA-232) serie asíncrono, administra un máximo de 3 unidades a una distancia de 20-40 metros con una velocidad de 20 kilobytes (115 kilobytes con algún hardware especial).
Administra: MODEM, ratón, instrumentación.

USB serie asíncrono, gestiona un máximo de 127 unidades a una distancia de 5 metros (mayor de 30 metros con 5 hubs) a la velocidad de 1.5M, 12M, 480M.

Gestiona: ratón, teclado, unidad de almacenamiento, audio, impresora, y diferentes periféricos tanto estándar como especiales.

una **TX** (trasmite el dato) y la otra **RX** (recibe el dato) (**pin 2 = RX, pin 3 = TX, pin 5 = masa**).

En la transmisión **USB** los datos que circulan por los terminales **D+** y **D-** pueden ser de tipo **RX** o **TX**, el software se ocupa de diferenciar entre el uno y el otro. Nosotros no debemos preocuparnos por esto, porque esta función la desarrollan los **drivers** asociados al periférico.

Por tanto, por medio de un simple esquema gráfico, las diferencias se aprecian entre una transmisión **RS232** y una transmisión **USB**.

RS232

Se utiliza la forma de transmisión de datos binarios llamada **NRZ-L**, acrónimo Bipolar de Non Return to Zero-Level, donde:

“**1**” representa el nivel de tensión físico normalmente negativo entre **-5** y **-12 Voltios**;

“**0**” representa el nivel de tensión físico, normalmente positivo, entre **+5** y **+12 Voltios** (ver Fig.2).

Se dice que la señal oscila del positivo al negativo respecto al estado precedente del clock bit. Medida del clock, la señal **0** indica una salida **positiva**, y la señal **1** una salida **negativa**.

USB

Se utiliza la transmisión de datos binarios **NRZI**, o lo que es igual Non Return to Zero Inverted, con

un principio similar, pero en este caso no es el nivel de tensión, ya que es solo la **transición** entre el más y el menos que distingue si un dato es **1** o **0**.

Por tanto, si hay un cambio el estado lógico será **1**, mientras que si el nivel queda constante (ya sea positivo o negativo) el estado lógico será **0** (ver Fig.3).

Entonces, los convertidores **RS232-USB** a nivel de hardware no son muy complejos, ya que existiendo una pequeña diferencia a nivel electrónico se compensa con un gran trabajo a nivel de software.

Respecto a los **USB**, a menudo escuchamos hablar de diferentes **tipos de uso**.

Esto se debe a que en el protocolo de transmisión del USB no solo vienen los datos de uso, sino que también se incluyen los datos binarios que reconoce el ordenador según el tipo de periférico.

Pongamos un ejemplo.

Si conectamos el ratón al ordenador, este será reconocido como “human interface”, es decir interfaz humana.

Por todo ello, se crean diferentes tipos de clases de usos específicos y de control: la serie de audio, la serie de almacenamiento masivo, etc.

De esta manera, el ordenador “sabe” rápidamente a que tipo de interfaz corresponde el “enchufe” del USB. En nuestro caso, se dice que el PC entra en fase de “**reconocimiento**”, es decir, el periférico USB se reconoce, y entra en la lista de accesorios varios del panel de control, si estamos trabajando con Windows (**Nota:** Inicio – panel de control – sistema – hardware).

| CONECTOR USB | | | | | | |
|---------------------------------|--------------|------|----------|------|-----------|------|
| P I E D I N I | A STANDARD B | | A MINI B | | A MICRO B | |
| | | | | | | |
| 1 | VBUS | VBUS | VBUS | VBUS | VBUS | VBUS |
| 2 | D- | D- | D- | D- | D- | D- |
| 3 | D+ | D+ | D+ | D+ | D+ | D+ |
| 4 | GND | GND | GND | n.c. | GND | n.c. |
| 5 | === | === | GND | GND | GND | GND |



Fig.2 El protocolo RS232 se caracteriza por un inicio y un final (bit start y bit stop). En el gráfico mostramos como se presenta un protocolo RS232. Cada cifra binaria del "texto" (8 bit) se obtiene sencillamente de la polaridad de la señal. Cuando el signo de la señal sea negativo, tendremos un 1, cuando sea positivo, tendremos un 0.

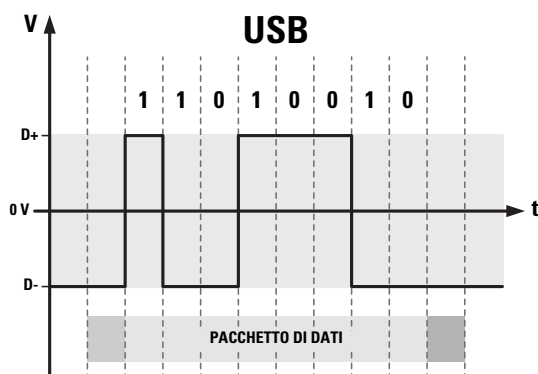
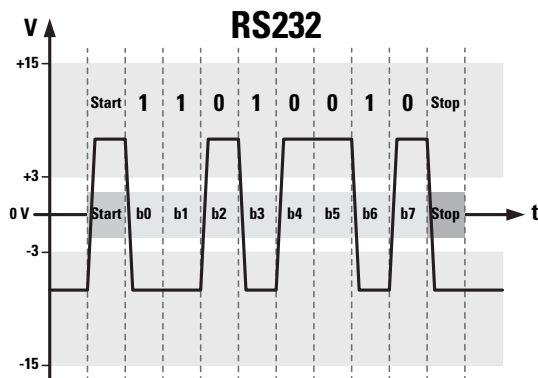


Fig.3 El protocolo USB es muy complejo. Lo que se puede ver es solo una pequeña parte del protocolo general, que se sucede en diferentes niveles. El sistema USB, a diferencia del RS232, transfiere la información 1 todas las veces que la señal pasa de + a -. Por el contrario, cuando queda constante tanto en la zona positiva como en la negativa la información es siempre 0. Por lo tanto se trata de un sistema mucho más veloz y con menor número de errores.

Nuestro periférico se reconoce como tarjeta de I/O en el microchip, porque la gestión y los drivers que utiliza el microprocesador los suministra el MICROCHIP.

Las conexiones USB se dividen en tres grandes grupos:

Un grupo lo constituyen los periféricos de **almacenamiento masivo**, incluidos los "pen drive", que sirven para memorizar datos, que aparecen una vez se ha conectado el USB, se reconocen como memoria masiva añadiendo al PC una unidad: en realidad es como si tuviésemos otro disco duro:

A/B = disquete

C = disco duro

D = lectora / grabadora de CD-Rom / DVD

E = Pen drive

Esta última unidad es igual en todo al disquete que se utilizaba para transportar datos de una unidad a otra.

Otra conexión **USB** es esa que se ve como un **puerto serie de comunicación**, en donde la gestión de datos se desarrolla de la misma manera que el puerto serie con la única diferencia que no debemos enfocar los datos de comunicación como la tasa de baudios, el número de bit es la paridad. Únicamente se necesita saber que puerto serie se acopla al USB que estamos utilizando. En la paridad de entrada, al USB siempre le corresponde el mismo puerto serie. Por ejemplo, en el PC en el cual estoy escribiendo este artículo es el COM6, en el PC en el cual he desarrollado los diferentes software de muestra, es el COM5. En otros ordenadores, como por ejemplo, los que se encuentran en el laboratorio es el COM4. La numeración depende solamente del primer puerto serie libre tras la primera inserción del USB.

El tercer grupo lo constituyen aquellos que utilizan el **USB** como impresora, ratón, etc, denominándose "**human interface**", que transmiten datos según los protocolos creados por la casa productora de periférico.

raciones consisten en la **instalación** de los **drivers** y en la selección de los **puertos de comunicación**.

INSTALACIÓN del DRIVER y selección del PUERTO de COMUNICACIÓN

El programa viene con un **Driver** que se puede instalar tanto con **WindowsXP** como con Vista. Los ejemplos del software de gestión y las fuentes se han realizado en **Visual Basic 6**, que no está directamente compatibilizado con **Vista**.

Para que funcione con Vista, los más experimentados podrán descargar gratuitamente **Visual Basic 2008** que es compatible con Vista en la Web de Microsoft, y cambiar nuestros programas de VB6 a **VB2008**.

Recordamos que su procedimiento es complejo, y por este motivo solo se lo recomendamos a los más versados en la materia.

Para realizar el trabajo con éxito debemos disponer de un **PC** que funcione con el sistema operativo **Windows XP**, y tener el **VB6**.

Para que la tarjeta USB sea compatible con el **software** es indispensable instalar los **drivers**. Una vez insertado el **CD-Rom 1734**, el ordenador al conectar por primera vez el **USB** en el puerto de vuestro PC, aparecerá la ventana izquierda de la Fig.9

Por tanto, haremos click en inicio y aparecerá otra ventana donde se solicitan los Drivers CCS. Pulsaremos entonces sobre el botón **select**, y posteriormente sobre la carpeta del **Driver CCS** del CD-Rom y en **aceptar**.

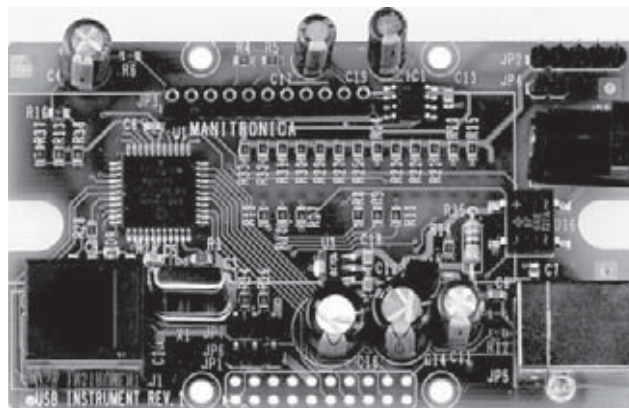


Fig.4 Foto de la interface USB denominada KM1734K, que suministramos ya montada con componentes en SMD y probada.

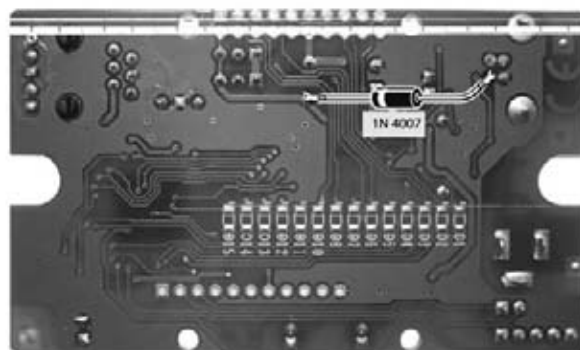


Fig.5 Tenemos aquí representada la tarjeta KM1734 vista desde su lado posterior, es decir del lado donde debemos soldar el diodo de silicio 1N4007 necesario para transferir la alimentación de 5 Voltios procedente de la toma USB del PC a todos los componentes. El diodo no se deberá montar en el caso del conductivímetro, ya que para este se necesita una alimentación externa de 12 Voltios.

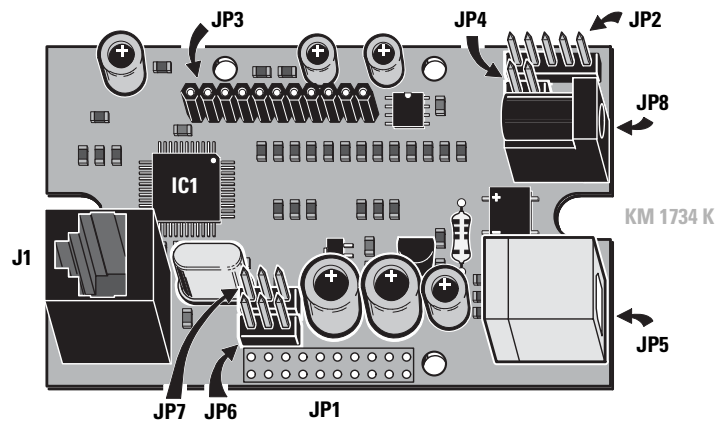


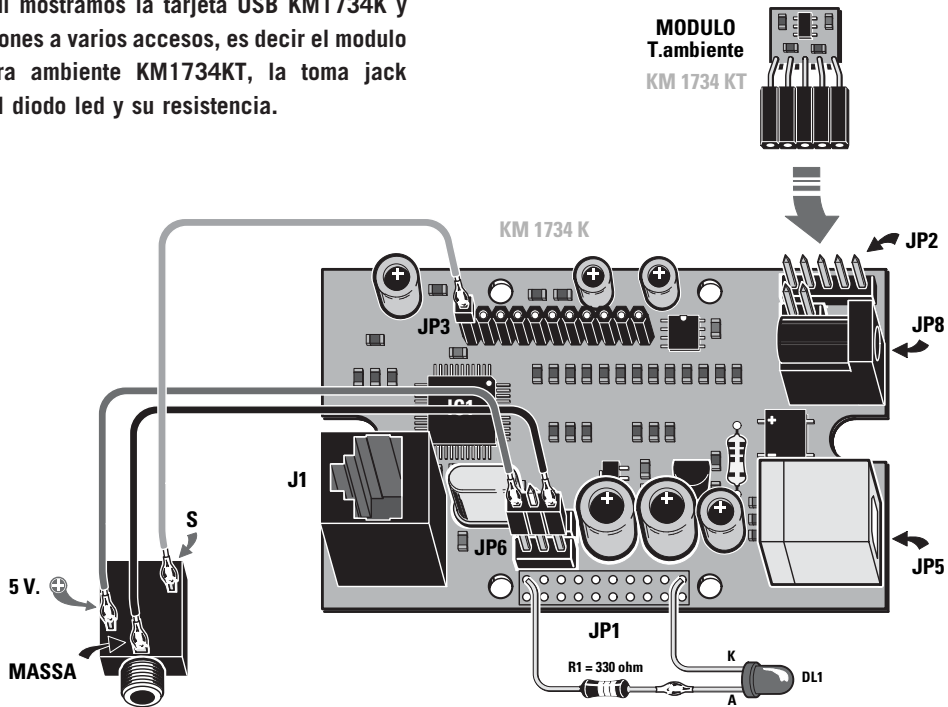
Fig.6 Esquema práctico del montaje de la tarjeta USB KM1734K.

LEYENDA

JP4 = con. por célula conducti
 J1 = entrada programación PIC
 JP5 = conector USB
 JP1 = Conexión expansión I/O

JP6 = conexión expan. PWM
 JP2 = conexión por modulo T° ambiente
 JP7 = conexión expan. PWM
 JP3 = conexión por modulo conductímetro
 JP8 = entrada aliment. 12 V

Fig. 7 Aquí mostramos la tarjeta USB KM1734K y las conexiones a varios accesos, es decir el modulo temperatura ambiente KM1734KT, la toma jack hembra, el diodo led y su resistencia.



| MSB | 12 bit | | | | | | | | | | LSB |
|----------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| | | | | | | | | | | | |
| 2^{11} | 2^{10} | 2^9 | 2^8 | 2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| 2.048 | 1.024 | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Así pues, aparecerá la ventana de la derecha de la Fig.9, pinchando en **finalizar** para concluir. Reinsertaremos de nuevo el interface. Si la luz del led **parpadea** indicará que la operación ha sido terminada con éxito.

Una vez llegados a este punto es necesario verificar a que puerto **COM** corresponde la tarjeta **USB**. Para llevarlo a cabo, seguiremos el ejemplo indicado en la Fig.10, en la sección dedicada a la instalación del programa.

La **instalación** de los **drivers** y la **selección** del puerto COM no se deberán repetir más veces una vez ejecutados, y sirve para todas las aplicaciones proponemos en la segunda parte de este artículo.

Entre el VOLTIO y el BINARIO

Todos los sensores que utilicemos para nuestras aplicaciones (ver siguientes páginas) deben entregar a su salida una tensión proporcional. La máxima tensión que podemos aplicar a nuestro circuito es de **+5V** (aunque utilizando un divisor de tensión es posible aplicar una tensión mayor de **+5V**).

La tensión aplicada en la entrada, que será proporcional a la "magnitud física" y que mediremos (de la misma manera que la temperatura se mide con un sensor de temperatura), se aplicará al **convertidor A/D** (analógico/digital) con una resolución de **12 bit**.

Recordamos que se entiende por 12 bit.

12 bit indica el número de cifras binarias que corresponden a la tensión de entrada después de haberse ejecutado la conversión **A/D**: en la práctica, el **convertidor A/D** transformará la tensión aplicada en la entrada en un número binario de **12 cifras**.

Como el número más alto que se puede conseguir con **12 cifras binarias** es **4.096** ($2^{12} = 4.096$), hace suponer que la diferencia de tensión más pequeña que se pueda apreciar será igual a:

$$+5V : 4.096 = 0,00122 \text{ V igual a } 1,220 \text{ mV}$$

Para comprender mejor este concepto observaremos la tabla adjunta.

En la primera fila horizontal hemos descrito el ejemplo de un **dato** en salida del **convertidor A/D**, en la segunda fila el valor **binario** correspondiente, y en la tercera los valores decimales equivalentes. Si ahora sumamos entre ellos los valores **decimales equivalentes**, que se encuentren en algún valor binario correspondiente a **1**, en la salida del **convertidor A/D** obtendremos:

$$1+2+4+16+64+256+1024 = 1.367$$

que se corresponde con el **decimal** del número binario estimado.

Para saber a que valor de tensión de entrada corresponde, es suficiente con **multiplicar** el valor de tensión más pequeño apreciable para este número (que sabemos que es **1220 mV**):

$$1,220 \text{ mV} \times 1.367 = 1.667,74 \text{ mV}$$

correspondiente a **1.667,74 V**.

PRECIO de REALIZACIÓN

LX.1734: El kit LX1734 junto la tarjeta **KM.1734k** montada en **SMD**, y el **CD-Rom CDR1734** conteniendo las **fuentes**, los **drivers** y las **aplicaciones** de instalación, excluyen el **modulo de temperatura KM1734KT** 135,00 €

KM.1734KT: módulo de **temperatura ambiente KM1734KT**: 27,00 €

KM.1734M: Tarjeta **LX.1734** equipada con todos los circuitos: **4 kits programados**, es decir **conductímetro LX1734/1**, **Gaussímetro LX.1734/2**, **detector UVA/UVB LX.1734/3**, **Termópila LX.1734/4** + modulo **KM1734KT**: 270,00 €

ESTOS PRECIOS NO INCLUYEN I.V.A.

INSTALACION DEL PROGRAMA

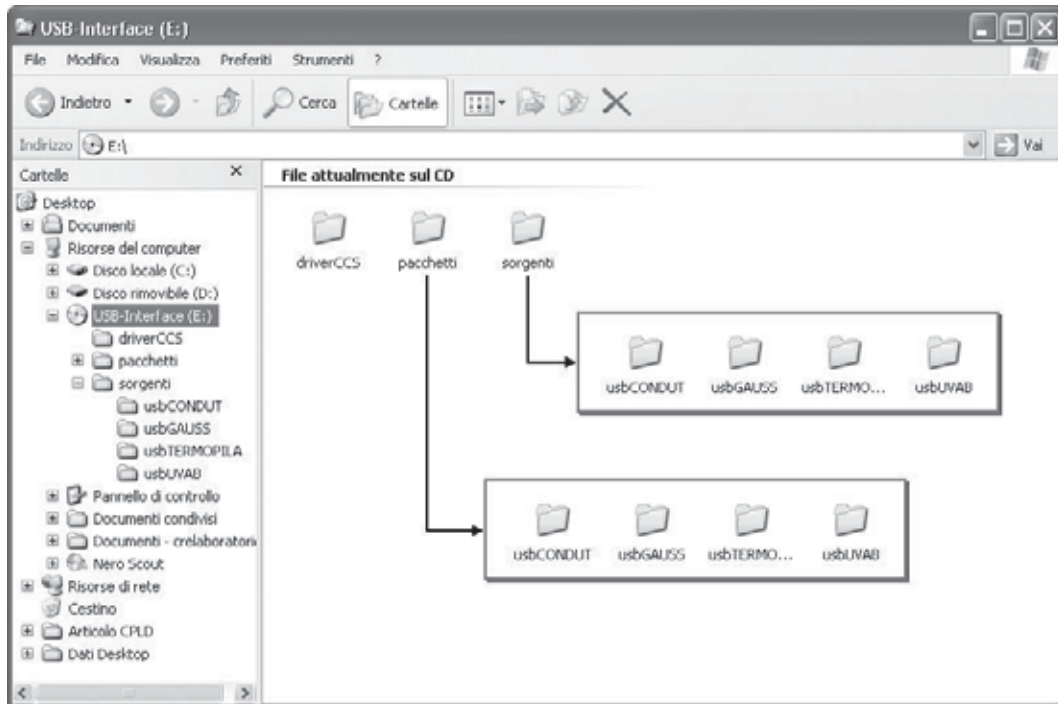


Fig.8 Insertad en el lector de DVD o CD-Rom, el CD-Rom CDR1724 que suministramos junto a la interfaz USB LX 1734. Hemos dividido el CR-Rom en tres carpetas principales. La carpeta de los drivers CCS: donde están los drivers que hacen funcionar el USB en el ordenador con el sistema operativo windows XP. La carpeta de fuentes: donde están todas las fuentes de varios programas de software. La carpeta de paquetes: donde se encuentran todos los programas preparados para ser instalados en el ordenador. Para cerrar la instalación del programa hacer click donde indica “septup.exe” dentro de la carpeta elegida.

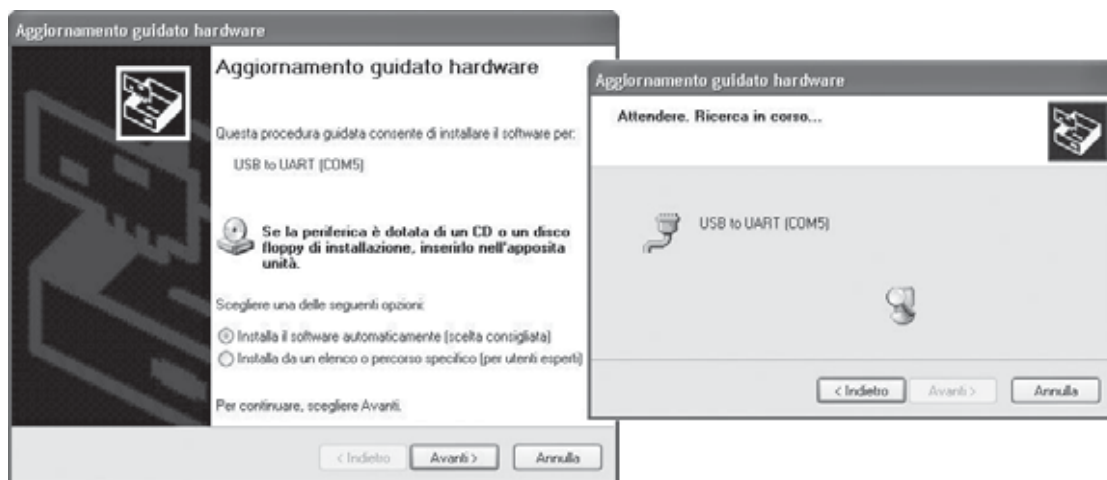


Fig.9 Introduciremos el cable USB conectado a la tarjeta KM1734K en un toma USB del PC. El PC reconocerá automáticamente un periférico nuevo, pidiendo instalar los drivers. A continuación clickad donde dice Adelante, Select y en la carpeta de los drivers CCS en Aceptar. Comenzará una búsqueda automática que terminará con un mensaje de finalización de la instalación, comunicándoos que la operación a finalizado con éxito.

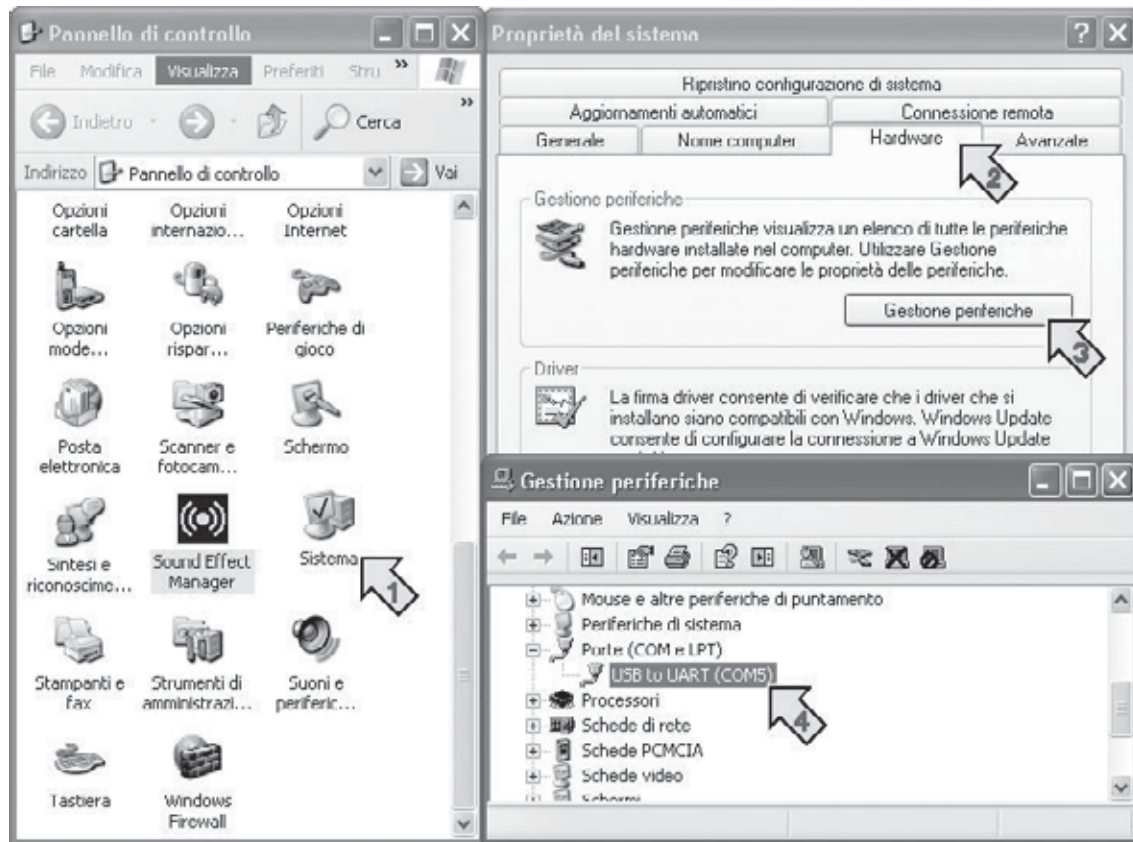


Fig.10 Ahora que la tarjeta USB está conectada, comprobada a que puerto COM corresponde vuestra tarjeta, pinchando sobre los siguientes iconos “ver” del menú, “sistema”, “hardware”, “gestión periférica”, “USB to UART”. En nuestro caso el puerto es la COM5.

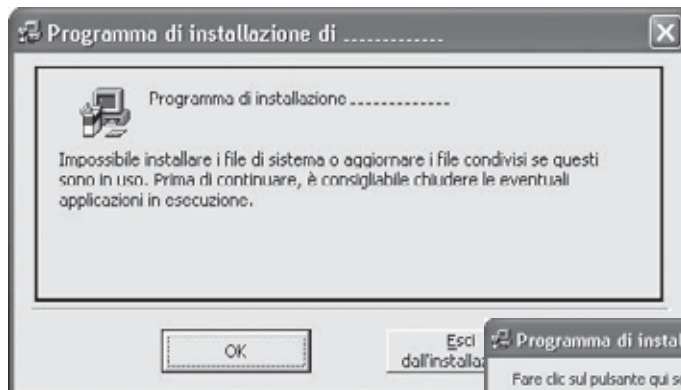
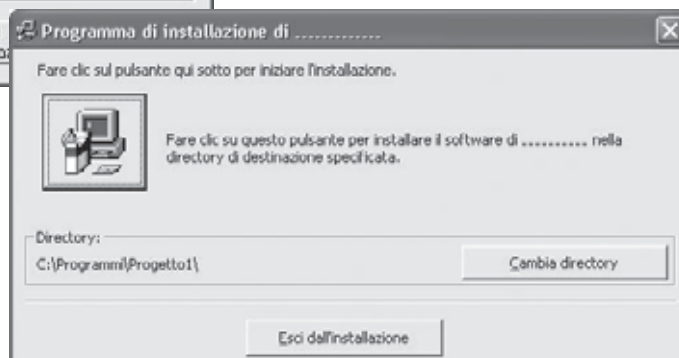


Fig.12 Clickad en el icono del PC para reiniciar la instalación. Sino tenéis preferencia no hagáis nada, y sino pinchad sobre el icono “cambia directory”.

Fig.11 Para instalar el programa demo tenéis que entrar en la carpeta de “paquetes” del CD-Rom y después en la carpeta de aplicaciones, clickando después en “setup. exe”.



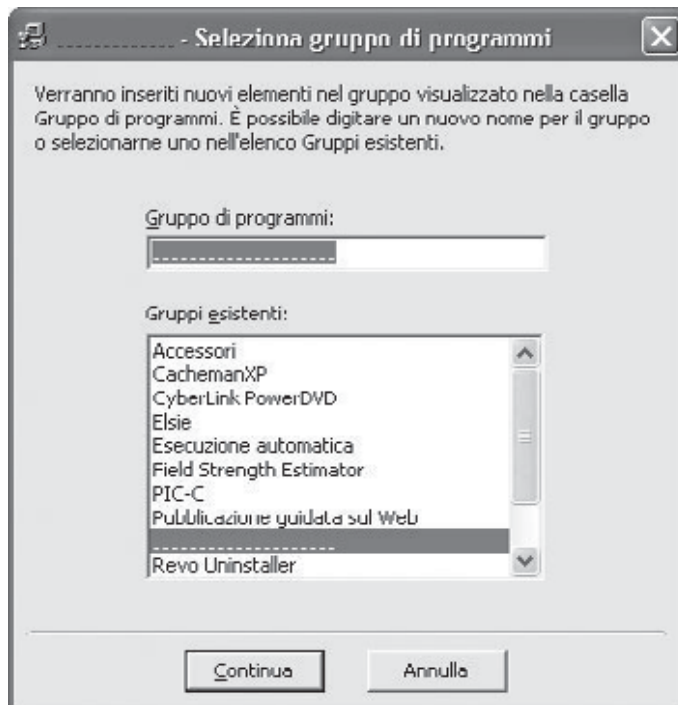


Fig.13 Os aparecerá esta ventana. Para seguir adelante solo tendréis que hacer click en continuar.

Fig.14 Cualquier aplicación que instaléis se cerrará con esta ventana. Pinchad en OK.

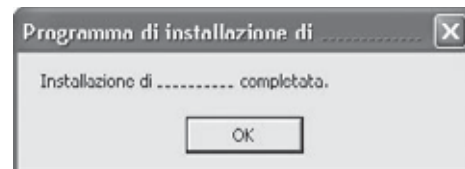


Fig.15 Para volver a los programas instalados, clickad en inicio, y luego abrir todas las aplicaciones clickando en el nombre del programa.

TARJETA USB PARA EL CONDUCTIVIMETRO

La primera aplicación que os queremos proponer es la de un 6, o sea un instrumento que nos sirve para medir la **calidad** del agua utilizando cualquier ordenador.

Para esta aplicación hemos utilizado el **módulo de conductividad** siglado **KM1697/1** en **SMD** que nos sirve para medir de **0 a 500 micro-Siemens** y para el acoplamiento de la célula de medida **SE1.K5**.

Gracias al conductivímetro nos es posible comprobar la “dureza” del agua que sale de nuestras tuberías, o lo que es igual, la cantidad de sal que hay en ella y su grado de mineralización, ayudándonos a comprender si puede ser consumida según los datos revelados.

Este dato también es interesante para quien trabaja en la cría de peces: de hecho, para que los peces de agua dulce puedan sobrevivir su agua debe estar muy dulce, mientras que los peces de agua salada deben vivir en un agua muy dura, o lo que es lo mismo “muy conductiva”.

Como hemos ya explicado, el **puerto USB** suministra además de los terminales de comunicación **D+** y **D-**, también una alimentación de 5 Voltios, que como mucho puede garantizar una corriente de **500 mA insuficiente** para conducir un interfaz útil para esta aplicación.

Por ello, necesitamos alimentaciones de **+5 a -5V**, para poder alimentar el **módulo de conductividad KM1697/1**.

Por tanto, es necesario acudir a una alimentación externa. Nos puede servir un alimentador de tipo comercial, o incluso uno de nuestros kit como el **LX.92**,



aunque también nos serviría sacar la tensión de **12 Voltios** de un PC que no sea un portátil.

Si pensáis utilizar el conector de alimentación de un PC, os recordamos que debéis utilizar los cables, rojo y negro, del conector que alimenta el disco duro, el lector de DVD o CD-Rom u otros periféricos.

REALIZACION PRACTICA

Para la realización práctica del circuito **LX.1734/1** podéis ver el dibujo explicativo que hemos diseñado en las Fig. 16 – 17.

Os sugerimos por comenzar acoplando en el conector **JP3** al módulo de conductividad siglado **KM1697/1**. Entonces, podréis pasar a insertar en el conector **JP4** el cable de conexión con la **célula de medida SE1.K5**.

Si disponéis ya de un alimentador de **12 Voltios**, introducid el enchufe en la toma **JP8**.

Podéis equipar el circuito con dos accesorios más, es decir, el módulo que nos advierte de la **temperatura ambiente** y el **diodo led + la resistencia** para comprobar visualmente la conexión.

En las Fig. 16-17 hemos ilustrado las diferentes modalidades de conexión de los accesorios del circuito impreso.

COMO USAR EL CONDUCTIVIMETRO

Después de conectar la célula de **medida SE1.K5** en el conector **JP4**, debéis acometer

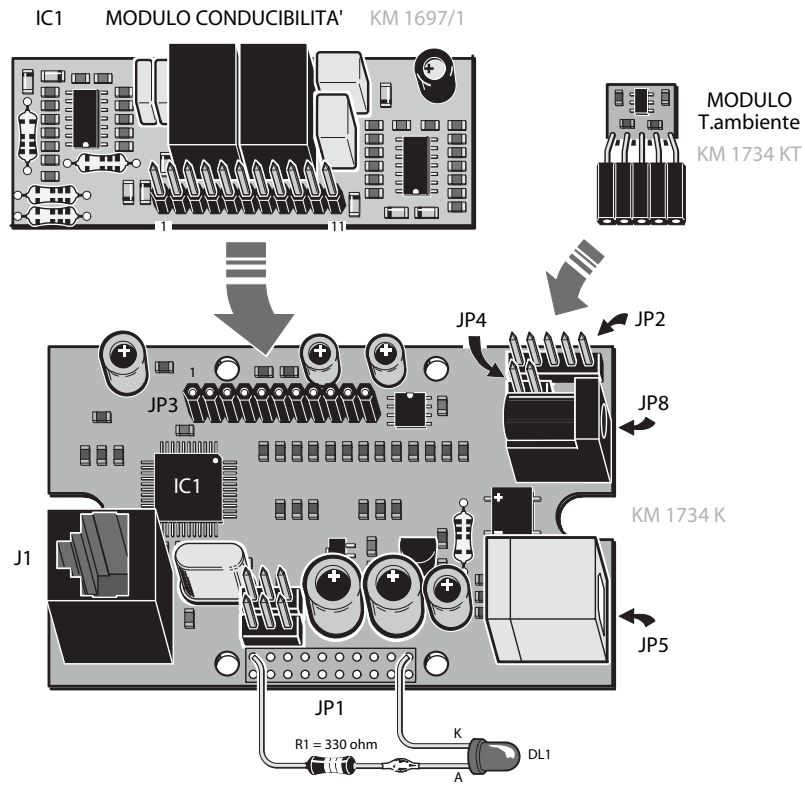


Fig. 16 Como podéis ver, para realizar esta aplicación debéis conectar el módulo de conductividad KM1607/1 en la tarjeta USB KM1764K.

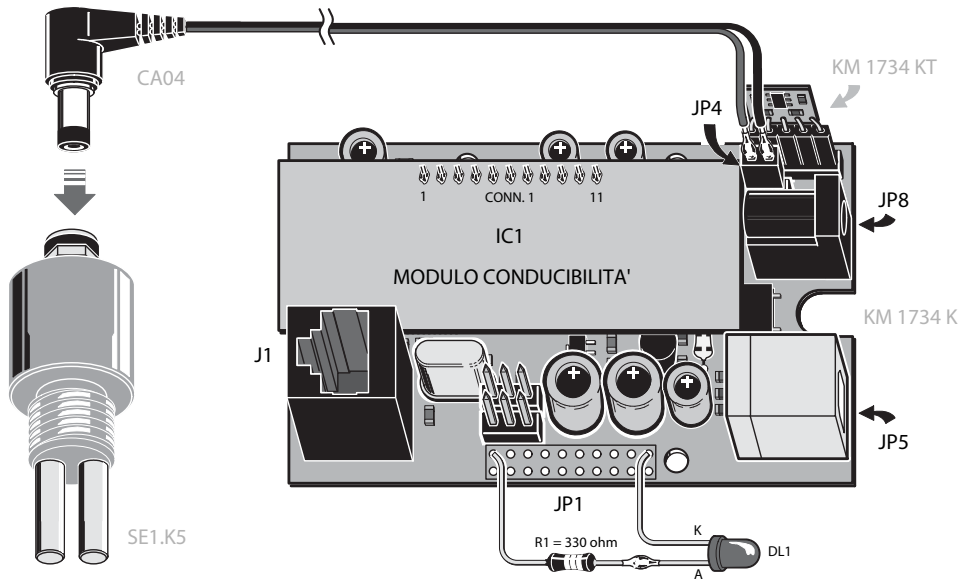


Fig.17 En este dibujo hemos explicado la conexión entre la tarjeta USB y la célula de medida SE1.K5 que encontraréis junto al kit.

dos calibrados: una en vacío, o sea, una **sin** conectar la célula dentro del agua, y otra inmersa dentro para revelar su conductividad.

El programa ya prevé un proceso de **calibrado**, que servirá para que el ordenador adquiera las dos medidas, que más adelante explicaremos cuando hablemos del software.

¿Por qué estas dos medidas? Porque la función matemática que nos suministra la correspondencia entre los **voltios**, que leemos con la célula, y los microSiemens, necesitan la **integración** de estos valores.

El **Siemens** es la medida **inversa** de la **impedancia Z** (en ohm) y se llama **“admitancia”**. Su gran tamaño nos sirve ya que de frente nos encontramos ante un parámetro, la conductividad, que es literalmente lo contrario de la resistencia eléctrica.

Teniendo dos valores, uno “en vacío” y otro “en el agua”, y sabiendo que este valor es lineal, podemos aplicar la tan conocida **formula** de la **recta**.

Formula:

$$Y2 - Y1 = m \times (X2 - X1)$$

Donde:

X e **Y** son las coordenadas de un plano imaginario y **m** es el **coeficiente angular Tg** (ángulo), que nos dice cual es la inclinación de la recta respecto a las ordenadas y abscisas.

COMO SE PROGRAMA

Ahora nos introduciremos en la programación.

Si tenéis el **Visual Basic 6** es la hora de utilizarlo: Clickad en el **archivo** y entrad en la carpeta de **fuentes** del CD-Rom, entonces, una vez dentro de la carpeta **usbCONDUT** (ver fig.8) debéis seleccionar el archivo con el nombre proyecto **conductor.vbp**.

Este archivo, además de la creación del programa, se ocupa de unir los diferentes módulos relativos a los **comandos**, a la **visualización gráfica** y a todas las **características** que distinguen al programa.

En la **form principal** denominada **Conductor.frm**, además otros tipos de trabajo como la denominada **Load_Forn** (que sirve para que el programa pueda comenzar con todo el trabajo), podéis ver el siguiente código:

Private Sub Form-Load()

‘escribe el nombre un archivo sobre el disco con la fecha como título

title = App.Path & “1” & “dati-” + Date\$ + “.txt”

Open title For Output As #2

‘activa el calendario

Calendar1.Today

Calendar1.Visible = False

‘carga los contenidos de los file.txt

AlfaNum1.Inizializza 2

AlfaNum1.Escribe “Nueva Electronica”

calibrado.quieneseran

Module1.quieneson

‘parad el contador para su lectura periódica en el puerto USB”

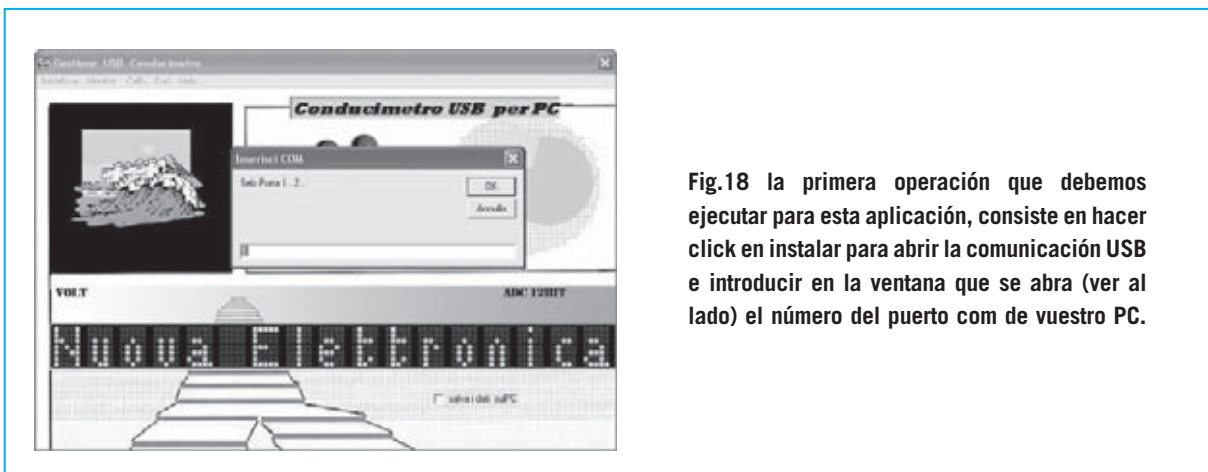


Fig.18 la primera operación que debemos ejecutar para esta aplicación, consiste en hacer click en instalar para abrir la comunicación USB e introducir en la ventana que se abra (ver al lado) el número del puerto com de vuestro PC.

Fig.19 en el monitor se abrirán los datos poco a poco. Después proceded al calibrado de la célula de medida.

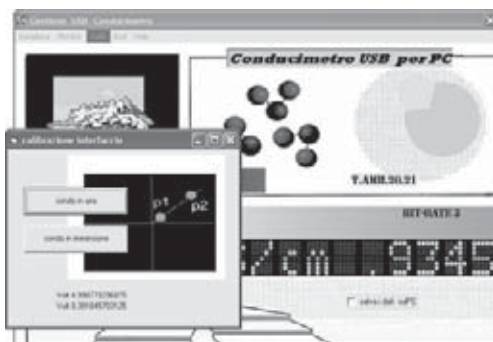


Fig.20 clickad sobre Calib y teniendo la sonda en el aire clickad en el botón, luego sumergirla en el agua y pulsad el botón.

Fig.21 clickando con el monitor podréis visualizar en un gráfico los datos que revela la sonda.

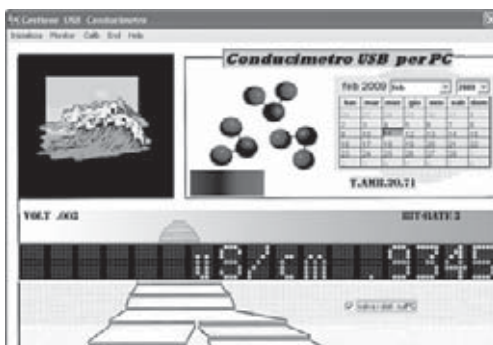
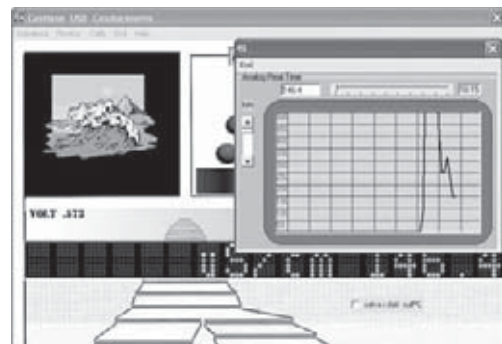


Fig.22 para guardar los datos obtenidos, clickad en el icono "guardar como" que hay en la ventana abajo a la izquierda.


```

txtReceive.Visible = False
'leed el puerto cada 1 millisecondo
tmrReceive.Interval = 1
tmrReceive.Enabled = False
'prepara el file guardar como"
Print #2, "Mes"&"Día"&"Año"&"Hora"&"
microSiemens "&"temperatura"
End Sub

```

Otra parte interesante del software es la relativa a la **comunicación** con el **puerto USB**, que se encuentra dentro de contador, desarrollado con las **API** de **Windows**.

```

Private Sub tmrReceive_Timer()
fSuccess = ReadFile(hCom, Buffer(1), Buffer-
Len, ReceivedBytes, 0)
If (fSuccess <> 0) Then
If ReceivedBytes <> 0 Then
For i = 1 To ReceivedBytes
txtReceive = txtReceive & Chr(Buffer(i))
' si llega el código 13 empieza a leer el dato
If Asc(Chr(Buffer(i))) = 13 Then
' busco la palabra TEMP=Temperatura
a = InStr(txtReceive, "TEMP=")
If a = 0 Then
GoTo fuori1
End If
title = Mid$(txtReceive, a + 5, 2) & "." &
Mid$(txtReceive, a + 7, 2)
If Len(title) > 3 Then
temperatura = Val(title)
Label1.Caption = "T.amb." & title
End If
fuori1:
' como arriba. Del puerto a a 12 bit el valor del sensor
a = InStr(txtReceive, "ADRA0=")
If a = 0 Then
GoTo fuori2
End If
title = Mid$(txtReceive, a + 6, 4)
señalAD0 = Val(title)
Label2.Caption = "bit-rate " & title
Module1.instrumentoANA
Module1.conductimetro (temperatura)
Monitor1.grafd1
fuori2:
End If ...

```

La gestión de la señal analógica en la entrada es realizada en el **módulo 1**:

```

volt = (5 * segnaleAD0) / 4096
'no tiene en cuenta la temperatura
microsim = 100 * volt / Val(variab(2))
'tiene en cuenta la temperatura
'microsim = (volt * (100 + (2 * temperatura))) /
100
Form1.Label5.Caption = "Volt" &
Left$(Str$(volt), 5)
'visualiza el dato en microSiemens/cm
'Form1.AlfNum1.Inicializa 2
Form1.AlfNum1.Scrivi "uS/cm" &
Left$(Str$(microsim), 6)

```

Veamos como se presenta el programa en general.

Haz click en "**inicializa**" para abrir la comunicación **USB** y escribe el número del **puerto** que habéis determinado en los parámetros del sistema (ver fig.18) en la form (en el ejemplo com5), luego pulsad **OK**.

La interfaz comenzará a comunicar cuando veamos que la luz led parpadea, y que en el monitor empiezan a verse los datos ver (Fig.19).

La primera operación que debemos realizar es calibrar la **célula** de **medida SE1.K5**.

Pinchad en **Calib** (ver. Fig 20) y, teniendo la sonda en el aire, pinchad en el botón "sonda en el aire", luego meterla en el agua y pulsad en el botón "sonda en inmersión".

Cerrad todas las aplicaciones haciendo click en el X roja de la derecha.

Pinchad en "**Monitor**" y veréis los mismos datos en forma de gráfico (ver fig.21), muy interesante para realizar un análisis visto desde el tiempo.

Si queréis guardar los datos, ir a "**guardar como**" y al segundo todos los datos quedarán memorizados en un archivo bajo el nombre **dati02-12-2009.txt** (mes, día, año) en la siguiente secuencia: mes, día, año, hora, minutos, segundos, microsiemens y temperatura.

Obviamente, la fecha se pondrá al día automáticamente cada vez que se ejecute el programa, cogiendo como referencia la fecha del ordenador.

Lamentablemente, debido a que el módulo de conductividad **KM.1697/1** de la tarjeta **USB** absorbe tanto, debemos estar constantemente alimentando al interfaz con un alimentador externo.