



Práctica

LabVIEW™

USB_CDC

DHT11 y ALS-PT19-315C

LabVIEW™

USB_CDC

Descripción

En esta práctica se implementa una comunicación USB configurada como CDC (Communications Devices Class) emulando un puerto serial UART para lectura de la luminosidad con el **X-NODE Ambient Light (XB009)**, así como la medición de temperatura y humedad con el **X-NODE Temp & Hum (XB003)**. Los datos serán solicitados a través de comandos y podrán ser visualizados mediante una GUI (Graphical User Interface) que se ejecuta mediante el software LabVIEW™.

Materiales

1 x Módulo **X-TRAINER**
1 x **X-NODE Ambient Light (XB009)**
1 x **X-NODE Temp & Hum (XB003)**
1 x **X-BOARD X-TRAINER**
1 x Cable micro USB.

Software

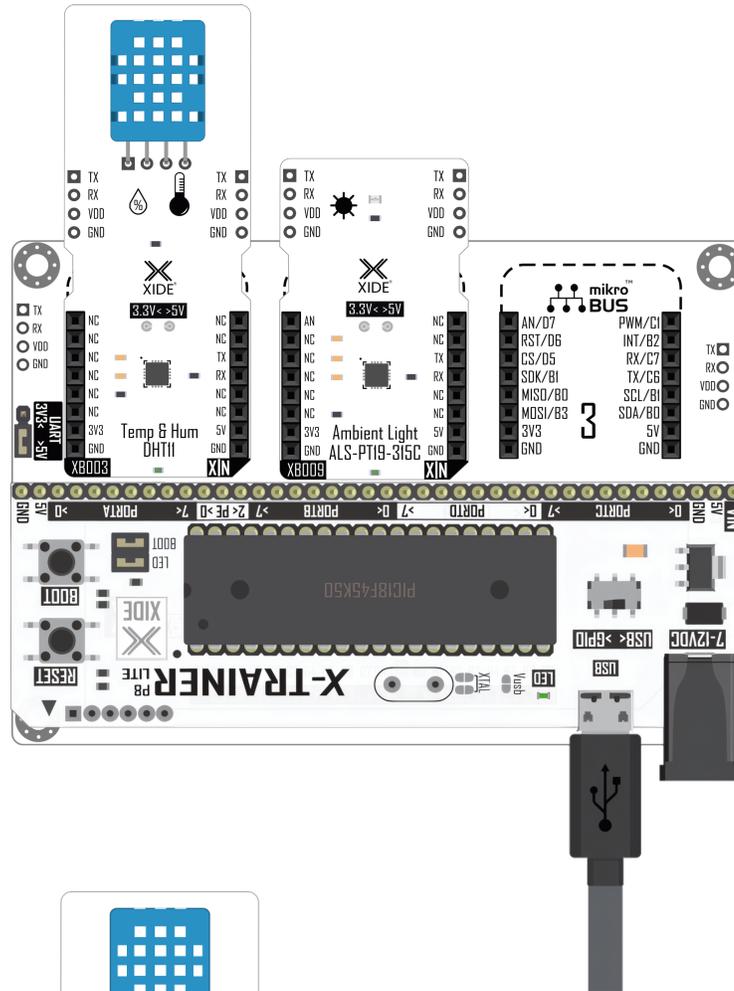
Windows 7 o posterior.
LabVIEW™ 2017 o posterior.
MICROSIDE v1.0

I. Procedimiento

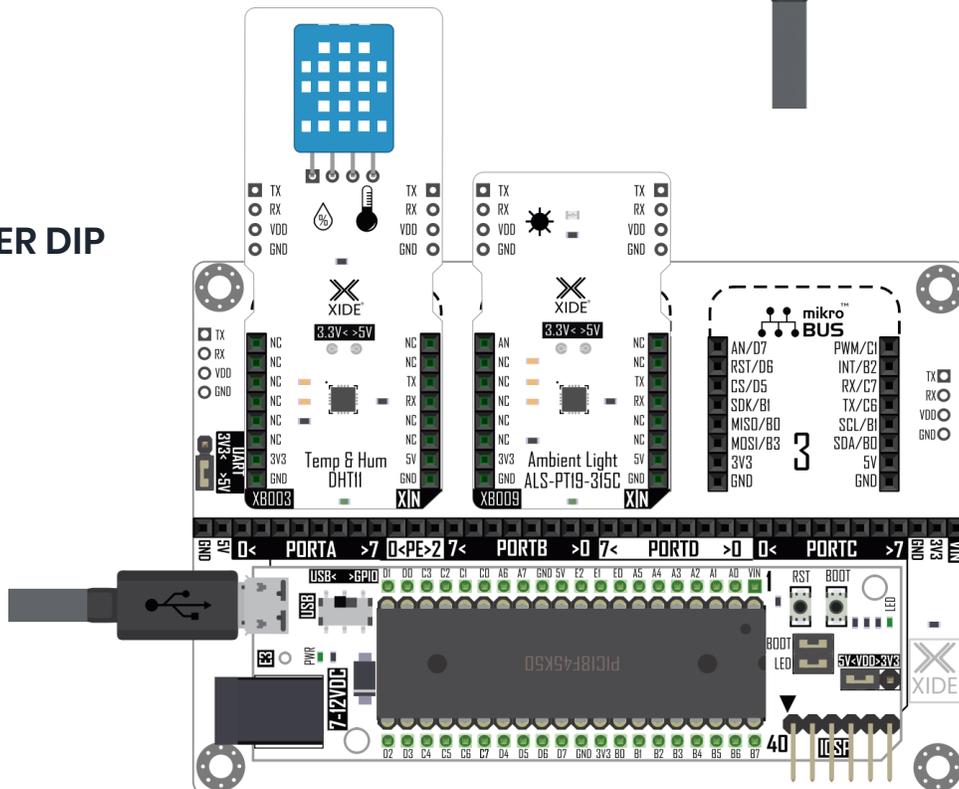
1. Realiza las conexiones que se muestran en la imagen del apartado “Diagrama Esquemático” y conéctalo a la computadora. **Nota:** Las conexiones mostradas en los diagramas esquemáticos son las mismas ya sea que la tarjeta X-TRAINER posea el PIC18F4550 o el PIC18F45K50.
2. En la carpeta incluida, puedes encontrar el archivo del código fuente necesario, dentro se encuentran las librerías para la compilación e incluye los drivers para la detección del puerto COM. También puedes visualizarlo en el apartado “Código” al final de esta práctica.
3. Ejecuta PIC C Compiler y compila el programa. En el siguiente [link](#) encontrarás un manual para compilar en PIC C Compiler.
4. Abre el Software MICROSIDE v1.0 y programa el código en el módulo **X-TRAINER**. En el siguiente [link](#) encontrarás el manual de usuario para el Software MICROSIDE v1.0.
5. Al terminar de programar, el módulo se reiniciará y el programa de esta práctica se ejecutará.
6. Entra al software LabVIEW™, abre el archivo llamado CDC-UART.vi que se encuentra en la carpeta de este ejemplo y configura el puerto COM que proporciona el **X-TRAINER** con los siguientes parámetros. Puedes guiarte con el apartado “Interfaz de LabVIEW™” que se encuentra más adelante.
7. Una vez configurado el puerto en LabVIEW™ ejecuta el programa y comenzarás a visualizar los valores de cada variable que proporcionan los módulos **X-NODE Ambient Light y Temp & Hum** a través de los comandos que se envían. Puedes encontrar estos comandos en los manuales de usuario de cada **X-NODE**.

II. Diagrama esquemático

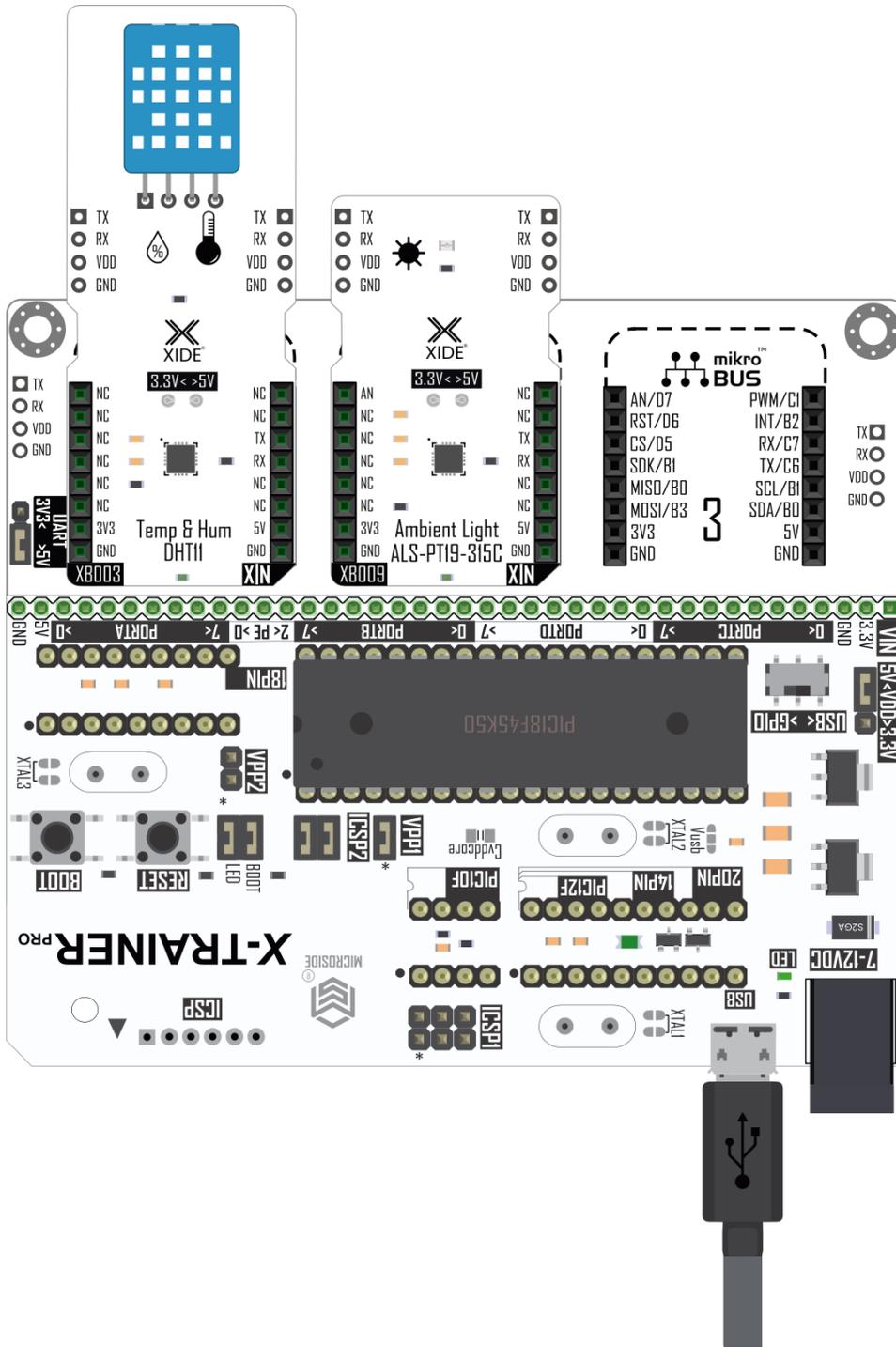
X-TRAINER LITE



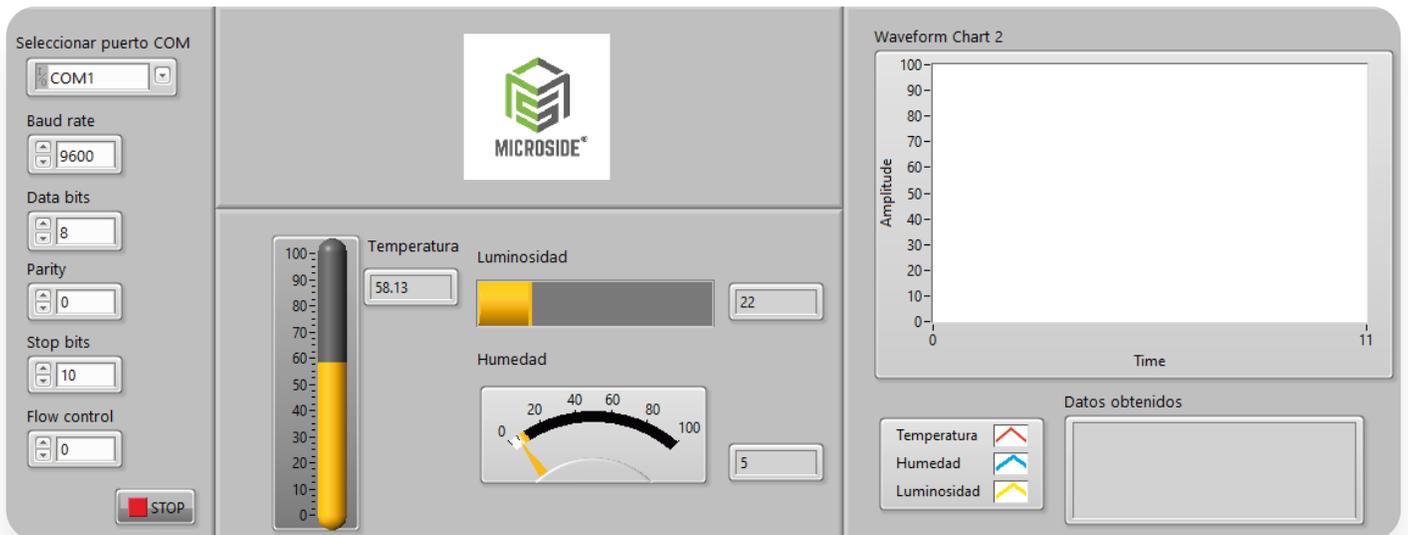
X-TRAINER DIP



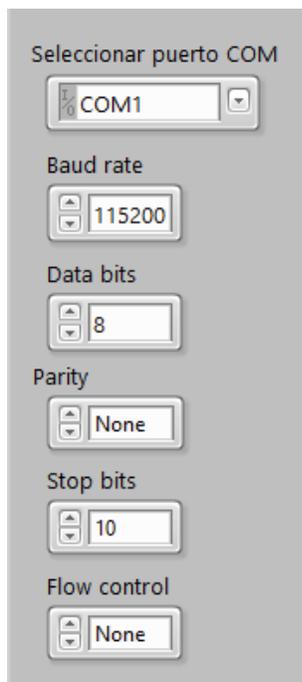
X-TRAINER PRO



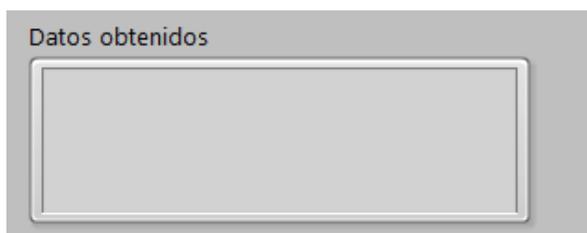
III. Interfaz en LabVIEW™



Configuración



- Selecciona el puerto COM que se asigna al momento de conectar el módulo por el puerto USB a la computadora, para identificar el puerto correcto puedes visualizarlo en el administrador de dispositivos de tu ordenador.
- Velocidad de comunicación: 115200 bps.
- Paridad: Ninguna
- Bits de datos: 8
- Bits de paro: 1

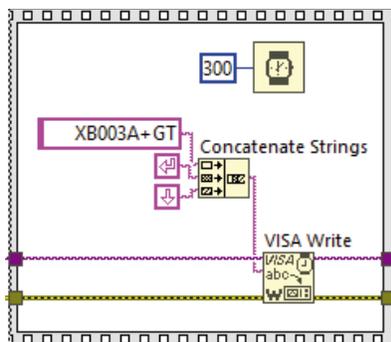


- Una vez ejecutado el programa la respuesta de los X-NODE será mostrada en el recuadro de “Datos obtenidos” así como en los indicadores correspondientes.

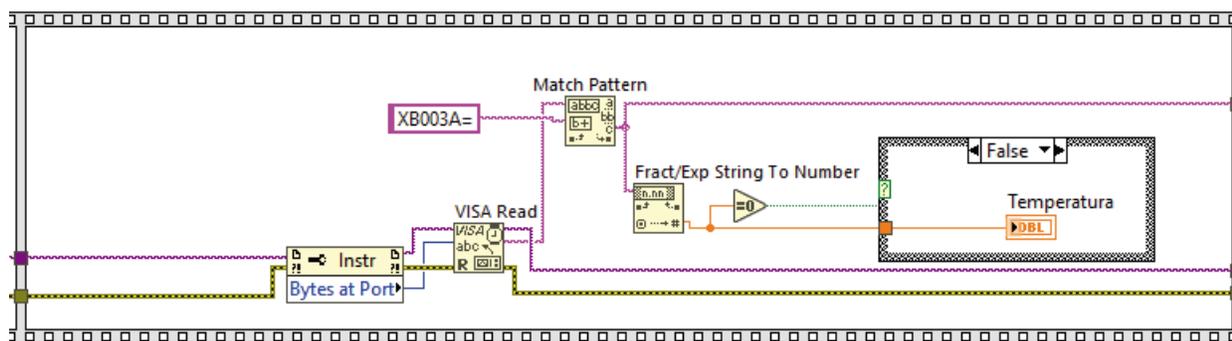
IV. Diagrama de bloques

Para poder establecer comunicación y obtener las variables deseadas, se deben introducir los comandos correspondientes de cada **X-NODE**.

1. En el bloque de “Concatenate Strings” que está conectado al bloque de escritura “VISA Write” se coloca el comando para solicitar datos al **X-NODE Temp & Hum** como lo indica su manual de usuario, para el caso de la temperatura el comando es “**XB003A+GT**”.



2. Una vez solicitada la información, el **X-NODE** responderá de la siguiente forma “XB003A=<0-100>”, al recibir el mensaje es necesario filtrar la información necesaria para el sistema, la función “Match Pattern” nos permite desechar la primera parte del mensaje, dejando el valor de la variable solicitada, es importante tomar en cuenta que la información que se recibe es de tipo string por lo que el bloque anidado “String To Number Function” se encarga de convertir la información a un valor numérico.



Se repite esta configuración para las variables de humedad y luminosidad con los comandos correspondientes de cada **X-NODE**.

V. Código USB_CDC

PIC18F4550

```
/*
AUTOR: MICROSIDE TECHNOLOGY S.A. DE C.V.
FECHA: NOVIEMBRE DE 2020
*/
/*
-----
En esta práctica se implementa una comunicación USB configurada como CDC
(Communications Devices Class) emulando un puerto serial UART para lectura
de la luminosidad con el X-NODE Ambient Light (XB009), así como la medición
de temperatura y humedad con el X-NODE Temp & Hum (XB003). Los datos serán
solicitados a través de comandos y podrán ser visualizados mediante una GUI
(Graphical User Interface) que se ejecuta mediante el software LabVIEW™.
-----
*/
#include <18F4550.h> //Incluye el microcontrolador con el que se va a trabajar
#include <delay.h> //Tipo de oscilador y frecuencia dependiendo del microcontrolador
#include <rs232.h> //Comunicación serial a 115200bps, 1 bit de paro, sin
paridad
#include <zero_ram.h>
#include <build.h> //Asignación de los vectores de reset e interrupción
#include <org.h> //Reserva espacio en la memoria para la versión con bootloader
//-----

#include <usb_cdc.h> //LIBRERIA USB
#include <usb_desc_cdc.h> //LIBRERIA USB

char c,u; //Crea variables tipo char para guardar los datos

void main() {

usb_cdc_init(); // Configuramos al puerto virtual
usb_init(); //inicializamos el USB

while (!usb_cdc_connected()) {} // espera a detectar una transmision de la PC
while (true)
{
usb_task();
if(usb_enumerated()) // Espera a que el dispositivo sea enumerado
{
if (usb_cdc_kbhit())
{
c = usb_cdc_getc(); //Espera a recibir un datos por el puerto USB
putc(c,UART); //Envía la informacion por UART
}

if (kbhit(UART))
{
u=getc(UART); //Recibe informacion por UART
usb_cdc_putc(u); //Envia informacion hacia la computadora por puerto emulado USB
}
}
}
}
}
```

PIC18F45K50

```
/*
AUTOR: MICROSIDE TECHNOLOGY S.A. DE C.V.
FECHA: NOVIEMBRE DE 2020
*/

/*
-----
En esta práctica se implementa una comunicación USB configurada como CDC
(Communications Devices Class) emulando un puerto serial UART para lectura
de la luminosidad con el X-NODE Ambient Light (XB009), así como la medición
de temperatura y humedad con el X-NODE Temp & Hum (XB003). Los datos serán
solicitados a través de comandos y podrán ser visualizados mediante una GUI
(Graphical User Interface) que se ejecuta mediante el software LabVIEW™.
-----
*/

#include <18F45K50.h> //Incluye el microcontrolador con el que se va a trabajar
#include <delay.h> // Tipo de oscilador y frecuencia dependiendo del microcontrolador
#include <rs232.h> //Comunicación serial a 115200bps, 1 bit de paro, sin
paridad
#include <zero_ram.h>
#include <reset.h> // Asigna los vectores de reset e interrupción para la versión con bootloader
#include <org.h> // Reserva espacio en memoria para el bootloader

//-----

#include <usb_cdc.h> //Librería USB
#include <usb_desc_cdc.h> //Librería USB

char c,u; //Crea variables tipo char para guardar los datos

void main() {

usb_cdc_init(); // Configuramos al puerto virtual
usb_init(); //inicializamos el USB

while (!usb_cdc_connected()) {} // espera a detectar una transmisión de la PC
while (true)
{
usb_task();
if(usb_enumerated()) // Espera a que el dispositivo sea enumerado
{
if (usb_cdc_kbhit())
{
c = usb_cdc_getc(); //Espera a recibir un datos por el puerto USB
putc(c,UART); //Envía la información por UART
}

if (kbhit(UART))
{
u=getc(UART); //Recibe información por UART
usb_cdc_putc(u); //Envía información hacia la computadora por puerto emulado USB
}
}
}
}
}
```



XIDE®

www.xide.pro