

**Project Name:** USBUART\_Example\_Project  
**Programming Language:** C  
**Associated Part Families:** PSoC 3, PSoC 5  
**Software Version:** PSoC Creator™ Beta 4.1  
**Related Hardware:** CY8CKIT-001  
**Prerequisites:** USBUART\_Lib  
**Author:** Dan Sweet

## Project Objective

This example project demonstrates the use of a custom developed USBUART component with PSoC<sup>®</sup> 3 / PSoC 5.

## Overview

The USBUART component uses a USB interface to emulate a COM port. This custom component is based on a standard USB component, modified to enumerate and act as a CDC USB device. High level communication functions are available on the PSoC device side and the PC communicates through a standard terminal program.

This example project demonstrates a PSoC project using a USBUART component to echo any data sent to it from a PC terminal.

Table 1. Component List

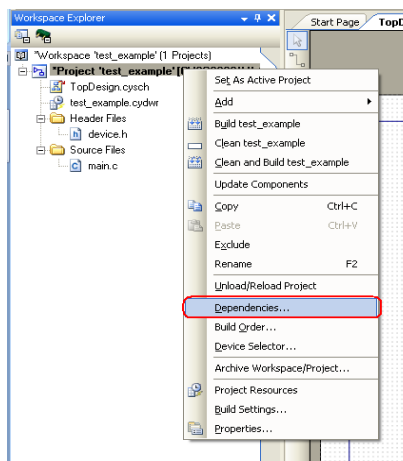
Instance Name	Component Name	Version	Component Category	Comments
USBUART_1	USBUART	1.0	Custom	VID, PID, Device Release, Mfr String and Product String are set

The USBUART component is the only component used in this example. To use this component in a project, a dependency must be set pointing to the library project containing the USBUART component. The instructions for adding this component to your own project are mentioned in the following section.

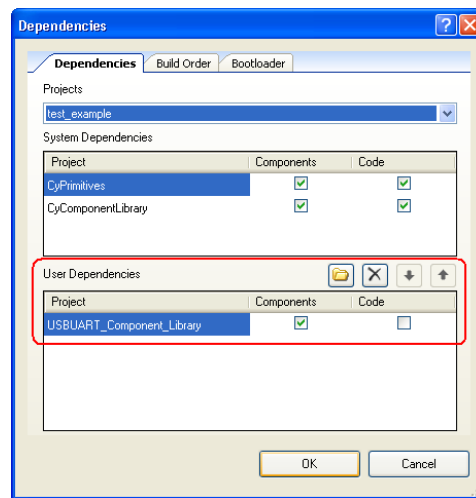
## Importing a Custom Component in a Project

Follow these steps to add the USBUART component to a new project:

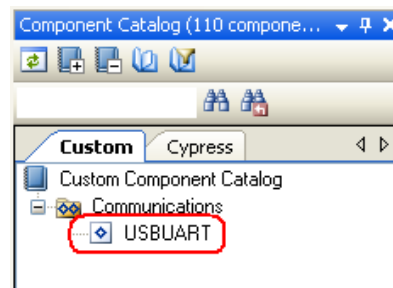
- Right click on the project name in the **Workspace Explorer** and select **Dependencies**.



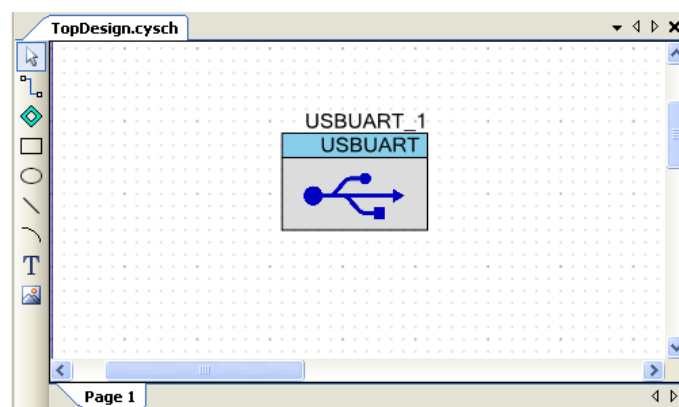
- Add a new User Dependency. Navigate to and select the *USBUSART\_Component\_Library.cypri* file provided with the example project.
- Ensure **Components** is checked and the **Code** check box is unchecked.



- Click **OK** and the USBUSART component now appears in a new tab in the **Component Catalog**. It is now ready to be placed on your schematic.

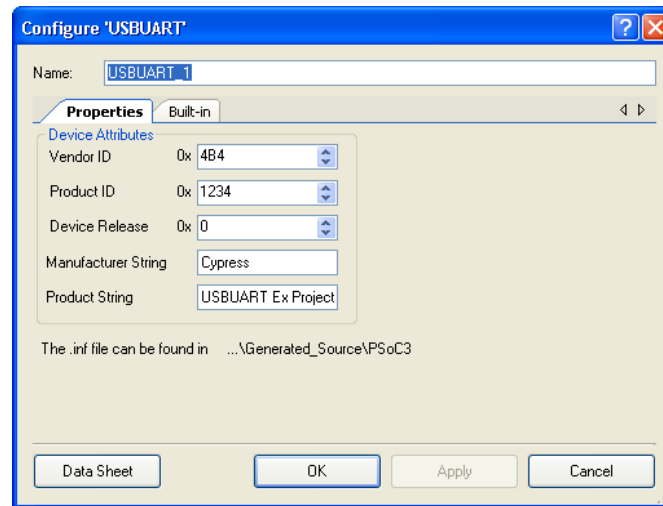


## Top Design



## Component Configuration

### USBUART

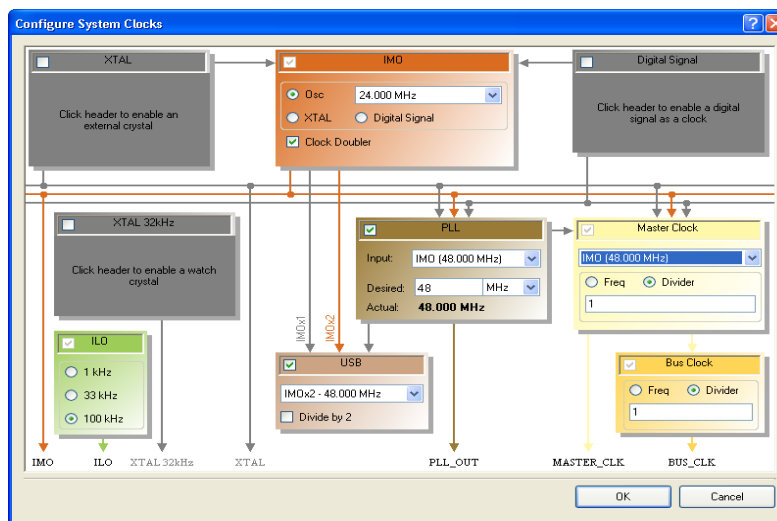


**Note** Each USB product must have a unique combination of Vendor ID (VID) and Product ID (PID). The default values provided here are used as an example, but a unique VID (assigned by the USB Implementers Forum) and PID must be used in a final product.

## Design Wide Resources

Changes must be made in the **Clocks** tab in the design wide resources file (.cydwr) to use USB on the PSoC 3 device. The clocks can be configured by clicking **Edit Clock** in the **Clocks** tab. You must make the following changes.

- **IMO**: Select Osc 24.000 MHz and enable the Doubler
- **ILO**: Select 100 kHz.
- **USB**: Enable and select IMOx2 – 48.000 MHz.
- **Master Clock**: Select IMO – 48.000 MHz.



## Operation

This project echoes any data sent to it through the PC COM terminal. During initialization, global interrupts are enabled and the USBUART component is started and enumerated. In the main loop, the receive buffer is checked for data. If data exists, the data is read out of the buffer and written back to the PC. After writing data to the PC, the PSoC must wait for the transmission to complete before continuing; otherwise the PSoC may send new data to the PC before it is ready. This code is shown in [Code 1](#).

**Code 1 USBUART Example Code**

```
uint8 Count;
uint8 Buffer[128];

void main()
{
    /* Initialization Code: */
    CYGlobalIntEnable;
    USBUART_1_Start(0, USBUART_1_3V_OPERATION);
    while(!USBUART_1_bGetConfiguration());
    USBUART_1_Init();

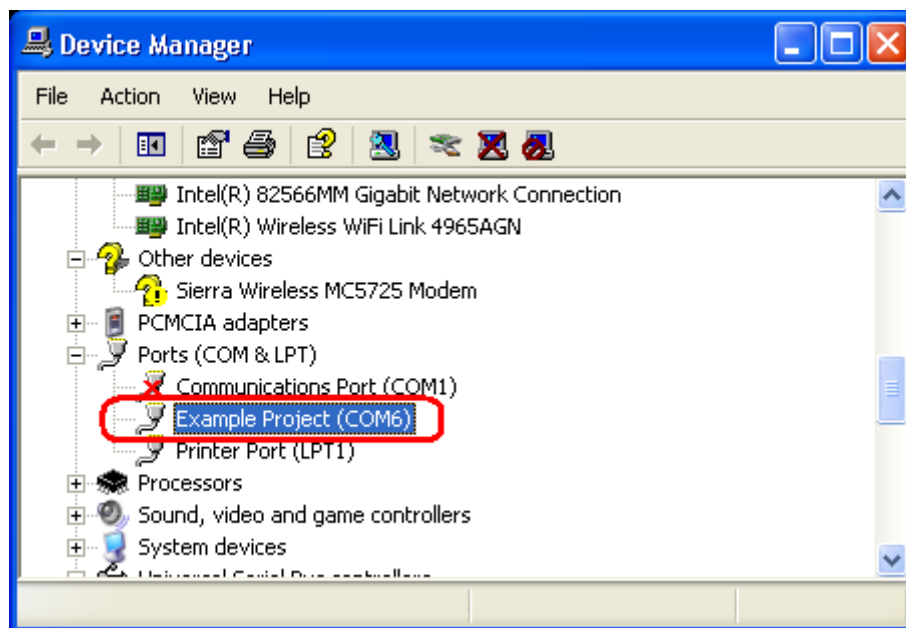
    /* Main Loop: */
    for(;;)
    {
        Count = USBUART_1_bGetRxCount();
        if(Count != 0)                                /* Check for input data from PC */
        {
            USBUART_1_ReadAll(Buffer);
            USBUART_1_Write(Buffer, Count);           /* Echo data back to PC */
            while(!USBUART_1_bTxIsReady()) {}         /* Wait for Tx to finish */
        }
    }
}
```

## Hardware Connections

- The default jumper settings of CY8CKIT-001 that is available in [CY8CKIT-001 PSoC Development Kit Board Guide](#) can be used.
- Jumper 8 may be changed the **VBUS** location, if you wish to power the board through USB.

## Output

- Build the Project and Program the chip. This project has been designed for both PSoC 3 and PSoC 5. The default device is PSoC 3 (CY8C866AXI-040). To change to another device, including PSoC 5 devices, go to **Project → Device Selector** and select the device. For further information see the PSoC Creator Help article Device Selector.
- Plug the USB cable into the DVK and reset the board (SW4). When you do this for the first time, Windows should prompt you with **Found New Hardware Wizard**. Direct the wizard to the `...\\USBUART_Example_Project\\USBUART_Example_Project.cydsn\\Generated_Source\\PSoC 3` directory, where the `USBUART_CDC.inf` file is located. This allows the device to enumerate correctly.
- Check to see which COM port the device is mapped to in the **Device Manager**.



- Open a COM terminal (such as HyperTerminal) and select the appropriate port. The baud rate selected does not matter. Ensure that the local echo is turned OFF.
- When a connection is established, the PSoC returns any information typed in the terminal.

PSoC is a registered trademark of Cypress Semiconductor Corp. PSoC Creator is a trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor  
198 Champion Court  
San Jose, CA 95134-1709  
Phone: 408-943-2600  
Fax: 408-943-4730  
<http://www.cypress.com/>

© Cypress Semiconductor Corporation, 2010. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.