USBUART

1.0

General Description

- The USBUART Device uses a USB interface to emulate a COM port.
- The required .inf file is automatically generated based on your component settings.
- *This is a custom component and has not been fully tested. Use at your own risk.*

Quick Start

1. To add this component to your project, simply add a dependency in your project that points to the library project containing the USBUART component. The USBUART component will then be available in your Component Catalog under ‘Custom -> Communications -> USBUART’.
2. Drag a USBUART component from the Component Catalog onto your design.
3. Notice the clock errors in the Notice List window; double-click on an error to open the System Clock Editor.
4. Configure the following clocks:
   - **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.
5. Select Build to generate APIs; refer to the Sample Firmware Source Code section for an example demonstrating the basic functionality of the USBUART component, as well as the basic setup instructions.

Input/Output Connections

None.

Parameters and Setup

Vendor ID

Each USB product must have a unique combination of Vendor ID (VID) and Product ID (PID). This 2-byte string contains the Vendor ID.

PRELIMINARY

Product ID

Each USB product must have a unique combination of Vendor ID (VID) and Product ID (PID). This 2-byte string contains the Vendor ID.

Device Release

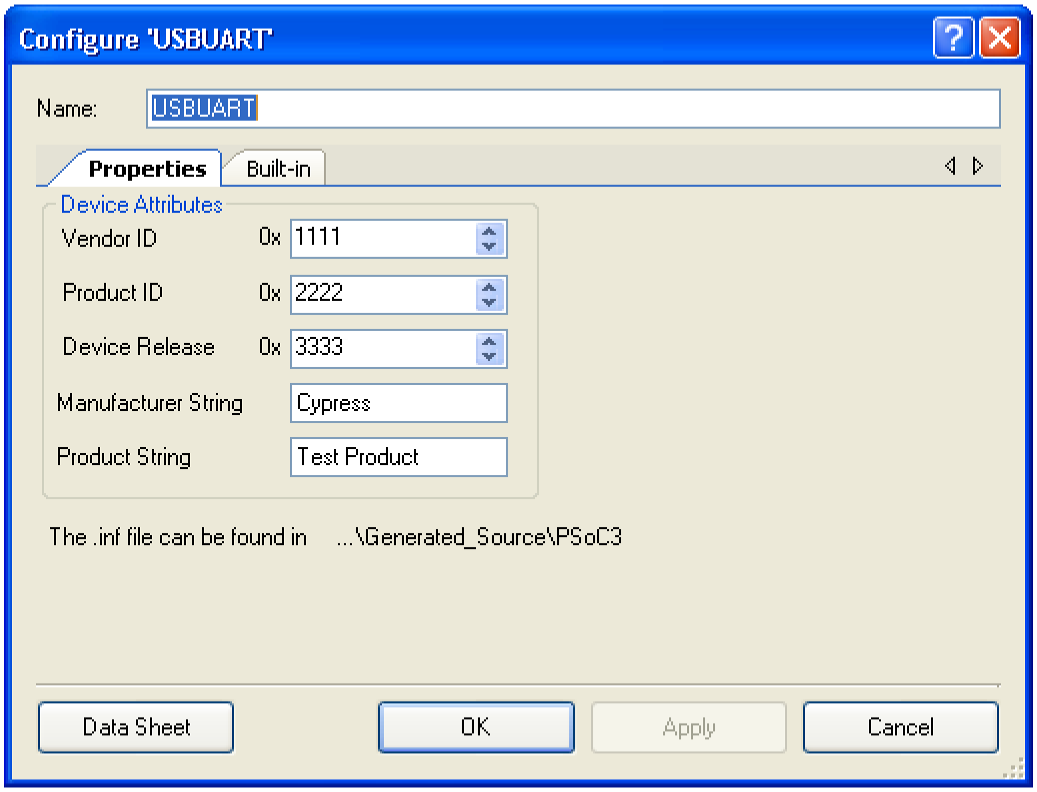
Use this field to keep track of your current release.

Manufacturer String

This string is automatically placed in the manufacturer field within the .inf file and will show up when the device enumerates and is views in the Device Manager on the PC.

Product String

This string is automatically placed in the description field within the .inf file and will show up when the device enumerates and is viewed in the Device Manager on the PC.



Name: USBUART

Device Attributes

Vendor ID	0x	1111
Product ID	0x	2222
Device Release	0x	3333
Manufacturer String		Cypress
Product String		Test Product

The .inf file can be found in ...\Generated_Source\PSoC3

Device Driver:

This component automatically creates a basic .inf file to allow your USBUART project to connect to your PC. The .inf file will be located in …YourProject/Generated_Source/PSoC3/. When you plug your device in for the first time, point Windows to this directory to find the .inf file and install the necessary drivers.

Application Programming Interface

This component has all of the standard USBFS API available. Please refer to the USBFS component datasheet for details on using this functionality.

Only a small subset of the USBUART functions that were previously available on PSoC1 have been implemented in this component.

Function	Description
uint8 USBUART_bGetRxCount(void)	Get the size of valid data in the receive buffer.
void USBUART_ReadAll (uint8 *pData)	Reads all data from the receive buffer into pData.
void USBUART_Write(uint *pData, uint8 bLength)	Writes the data to the Tx Buffer.
uint8 USBUART_bTxIsReady(void)	Checks to see if the device is ready to send data. If not, 0 is returned. Otherwise the Tx buffer size is returned.
uint8 USBUART_Init(void)	Initializes the component.

Sample Code

This example will echo any characters typed into a terminal program on the PC.

```
uint8 bCount;                                              /* count of data */
uint8 bIndex;
uint8 baBuffer[128];

uint32 wGetDTERate;

void main()
{
    CYGlobalIntEnable;                              /* Enable Global Interrupts     */

    USBUART_Start(0, USBUART_3V_OPERATION);         /* Start USBFS Operation/device 0
    and with 3V operation */
    while(!USBUART_bGetConfiguration());            /* Wait for Device to enumerate */
    USBUART_Init();                                 /* initialize the USBUART */

    while(1)
    {
      /* Get Baud Rate (not used in this example): */
      //USBUART_dwGetDTERate(&wGetDTERate);

        /* Receive Data and Echo: */
        bCount = USBUART_bGetRxCount();              /* get the USB Rx data size */
        if( bCount != 0 )
        {
            USBUART_ReadAll(baBuffer);               /* copy data to baBuffer */

            USBUART_Write( baBuffer, bCount );       /* echo / send out data */
                  while(!USBUART_bTxIsReady()){}
        }
```

PRELIMINARY

```
        }
    }
```