Simple 8-Bit UART Communication Example – PSoC® 3 / PSoC 5

EP56061

Associated Part Families: CY8C38xx/CY8C55xx
Software Version: PSoC® Creator™
Related Hardware: CY8CKIT-001
Author: Anup Mohan

Project Objective

The project aims at communicating PSoC® 3 / PSoC 5 with UART HyperTerminal.

Overview

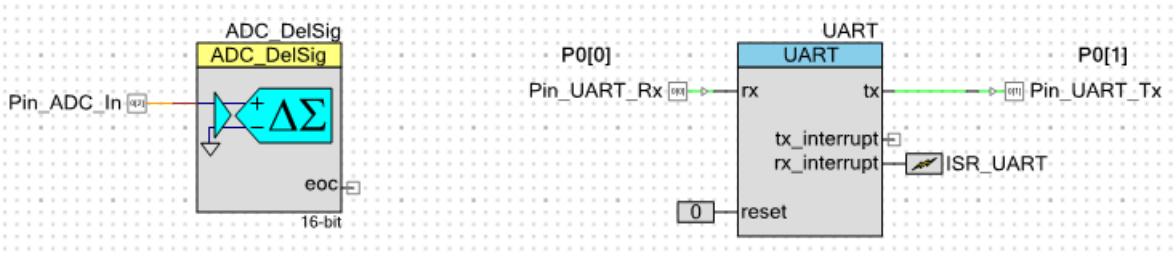
In this example, UART is used to communicate with HyperTerminal. UART is configured in 8-bit, full-duplex mode with a baud rate of 9600 bps. This project demonstrates the usage of UART Rx interrupts. In the project, the ADC data is sent to the HyperTerminal if the key 's' is pressed. If some other key is pressed, the data stored in UART_writeBuffer is sent to the UART HyperTerminal.

Component List

Instance Name	Component Name	Component Category	Comments
UART	UART	Communications	–
Pin_UART_Rx	Digital Input Port	Ports and Pins	Port direction configured as Input and Pin Drive mode configured as High Impedance Digital. Pin is mapped to P0[0]
Pin_UART_Tx	Digital Output Port	Ports and Pins	Port direction configured as Output and Pin Drive mode configured as Strong Drive. Pin is mapped to P0[1]
ZeroTerminal_1	Logic Low	Digital → Logic	–
ISR_UART	Interrupt	System	–

Top Design

The following figure shows the Components and their Routing.

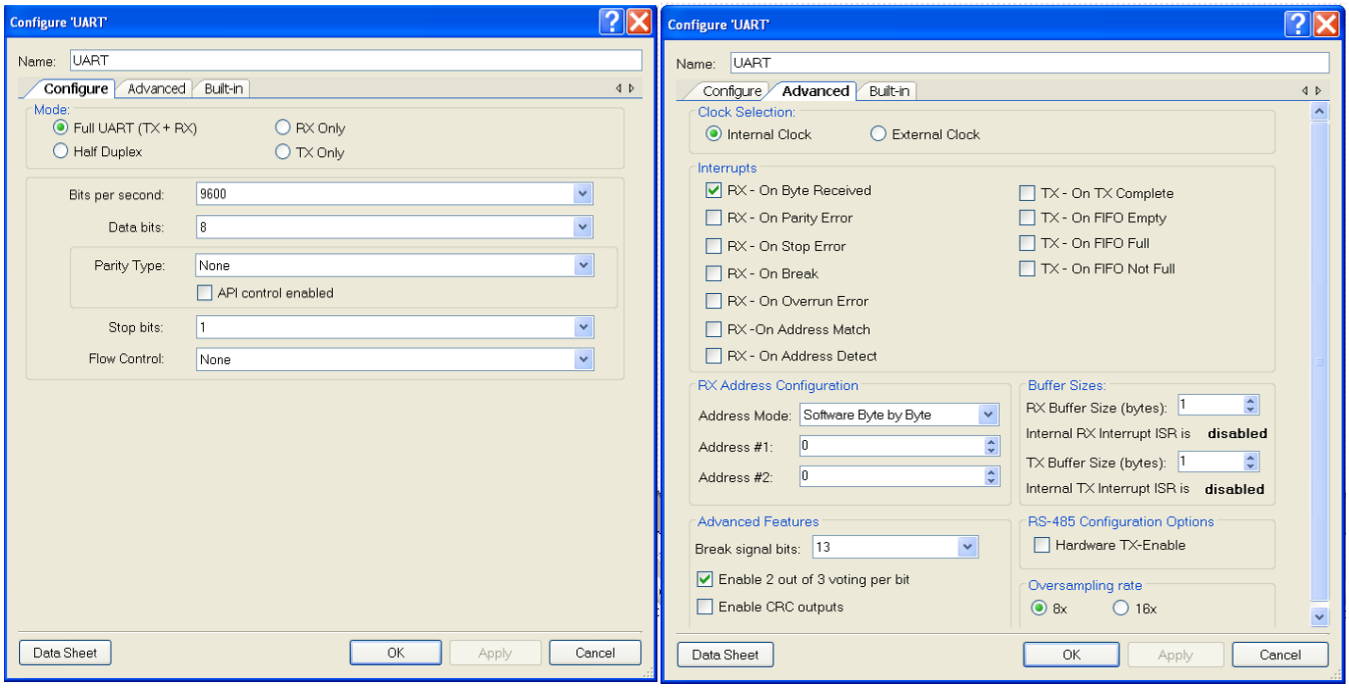


The following figure shows Pin Placement (as in *.cydwr* file)

Alias	Name	Pin	Lock
	Pin_UART_Rx	P0[0]	✓
	Pin_UART_Tx	P0[1]	✓
	Pin_ADC_In	P0[2]	✓

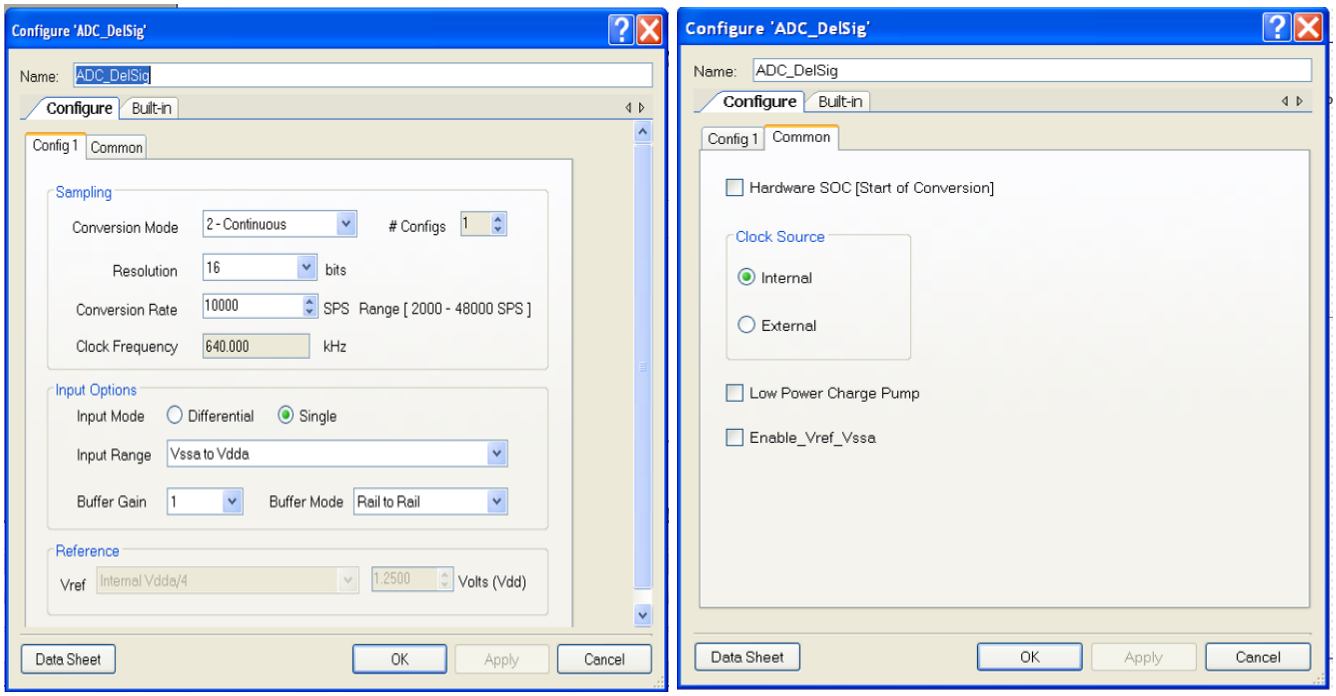
Component Configuration

UART



In the **Advanced** tab, the clock selection is selected as internal. On selecting internal clock, the PSoC Creator™ decides the clock frequency that is eight times the required baud rate. On selecting external clock option, you fix the frequency to eight times the baud rate required.

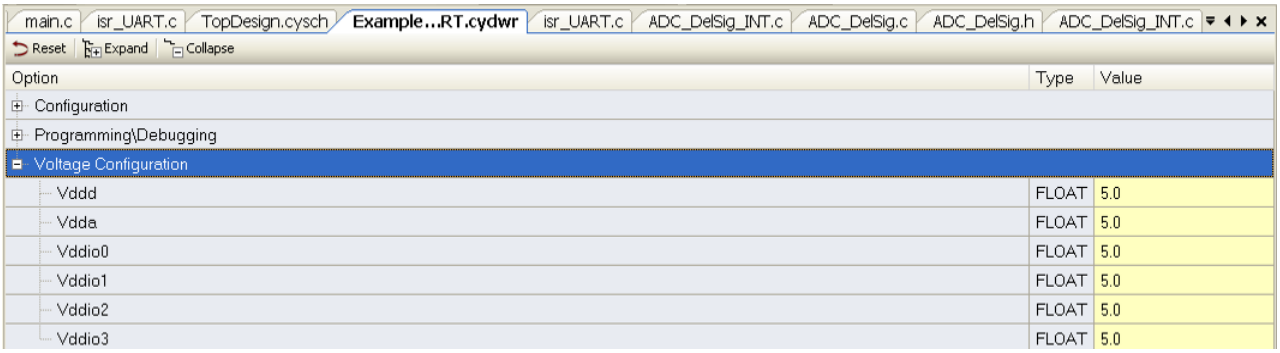
ADC



Design Wide Resources

Update the voltage configuration in *Example_UART.cydwr* file with the correct Vdd settings used to test the project.

Note In the latest ADC component, if the input range of ADC is set to Vssa to Vdda, the Vref value of ADC is determined by the voltage setting in the *cydwr* file. In this example, Vdd is set to 5 V.



All the remaining Design Wide Resources are set to the default value. Refer to the *Example_UART.cydwr* file for the settings.

Operation

The brief overview of the operation is given in this section. On pressing a key in the HyperTerminal, the UART receives a byte and generates an interrupt.

The internal interrupts of the UART component are enabled only if the buffer size is greater than or equal to 4. This is because the hardware FIFO is 4 bytes deep and internal interrupts are required to manage the software buffer when the buffer size is set to greater than 4. The interrupt file *UART_INT.c* is generated even if the buffer size is less than or equal to 4, but the code in that file will not be compiled unless internal interrupts are enabled.

This example shows how to use UART interrupt if the buffer size is less than or equal to 4. Here the RX Buffer size is set to 1 and hence internal interrupts are disabled. In order to receive an interrupt on byte receive, you have to check the 'RX on byte receive' interrupt option as shown in UART component configuration above. Connect an interrupt component (Renamed as ISR_UART) to the rx interrupt line. This will trigger the ISR_UART on every byte receive and code written in interrupt file (ISR_UART.c) is executed.

In this example a flag, 'flag_keyPress' is set inside *ISR_UART.c*. Inside the main code, the program reads checks if the flag is set and then reads the received data. If the received data is 's', the ADC sample is read, converted to ascii and is send to HyperTerminal over UART. If the received data is anything other than 's', the message stored in UART_writeBuffer is send to HyperTerminal. The UART_writeBuffer contains the message 'Press S to get ADC data'.

For sending buffer data, either of the following APIs could be used:

`UART_PutArray`: Places data from a memory array into the memory buffer for transmitting

or send byte by byte using

`UART_PutChar`: Puts a byte of data into the transmit buffer to be sent when the bus is available

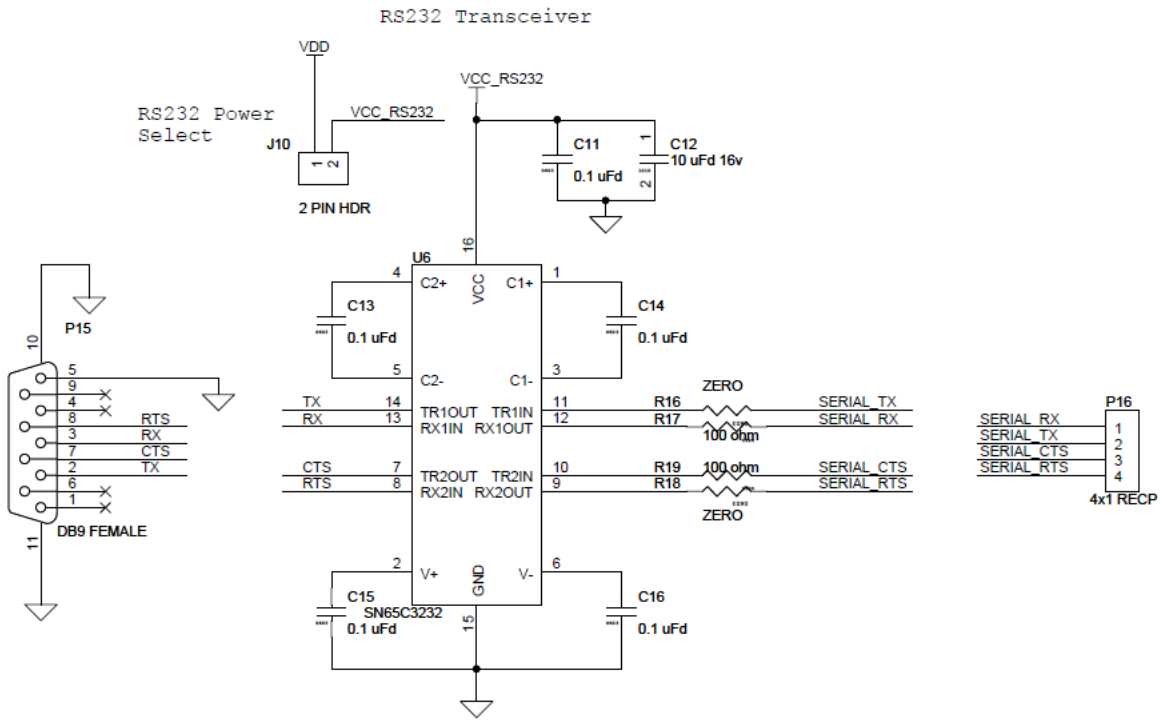
Note If you use the API 'UART_WriteTxData', it sends a byte without checking for buffer room or status. Hence, if this API is used instead of 'UART_PutChar', the previous data in Tx buffer might get overwritten.

Hardware Connections

This project can be tested on the CY8CKIT-001 development board. The following connections are done on the board to make the project work.

- Enable the UART power by setting jumper J10 to ON position
- Connect pin P0 [0] to UART Rx and pin P0[1] to UART Tx. UART Rx and Tx pins are available on the P16 header in the DVK
- Connect P0 [2] to potentiometer (VR on P14) of DVK. Make sure jumper J11 is in place to power the VR
- Connect RS232 serial cable to the serial port (P15) of the DVK and to the serial port of the computer
- For rest of the basic settings of the DVK, refer to the CY8CKIT-001 PSoC Development Kit Board Guide that is supplied with the kit.

Schematics for RS232 transceiver section in CY8CKIT-001:

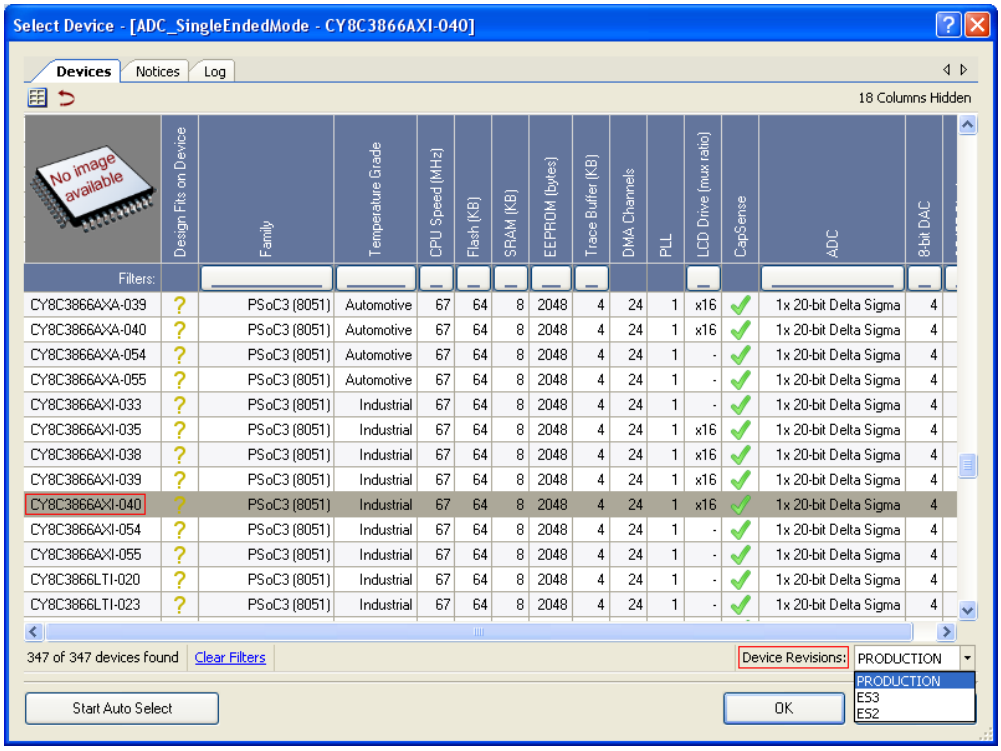


HyperTerminal Settings

- Click Start → Program Files → Accessories → Communication → HyperTerminal
- Enter a Name and select **OK**.
- In **Connect To** option, use the 'Connect using' to select the COM port to which the serial cable is connected and Click **OK**
- In the COM properties enter the following details and click **OK**
  - Bits per second = 9600
  - Data bits = 8
  - Parity = None
  - Stop Bits = 1
  - Flow Control = None
- Click **File** → **Properties** → **Settings** → **ASCII** setup and enable 'Echo typed characters locally' (The HyperTerminal should be in disconnected mode while setting this option).

Output

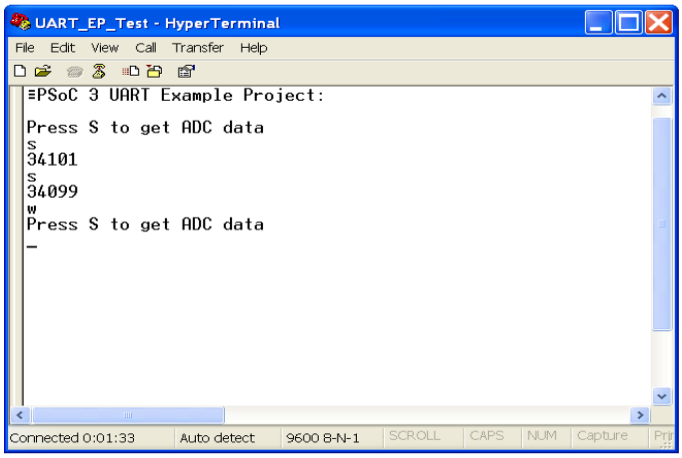
- Make hardware connection and setup the HyperTerminal as mentioned in the HyperTerminal settings.
- Use device selector (Project → Device Selector) window in PSoC Creator to select the appropriate device and Device Revision.
- If you are using PSoC 3 device, for example, CY8C3866AXI-040 with production revision, then use the following selection.



- Similarly, select appropriate device number to work with PSoC 5 Device family, for example, CY8C5588AXI-060

  Note For engineering samples, device revision is marked on the package as part of the device number. Production silicon will not have ES marking.
- Build the Project and Program the device
- Press SW4 (Reset Switch) to reset the device
- The ADC data is displayed in HyperTerminal when the key ‘s’ is pressed. If some other key is pressed the message ‘Press S to get ADC data‘ is displayed.

  A sample output as seen in the HyperTerminal window is as follows.



Related Application Notes and Example Projects

- 9-bit UART Communication in PSoC® 3 / PSoC 5
- USBUART in PSoC® 3 / PSoC 5
- 8-bit Universal Asynchronous Receiver Transmitter – PSoC® 1

Document History

Document Title: Simple 8-Bit UART Communication Example – PSoC® 3 / PSoC 5

Document Number: 001-56061

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2765427	ANUP	09/23/09	New example project
*A	2943815	ANUP	06/09/10	Updated to PSoC Creator Beta 4.1 and made it PSoC 5 compatible
*B	3102166	ANMD	01/07/11	Component version number removed from the component table. New figures and explanation added in System Wide Resource section. Updated the Output section.

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