CC32xx SPI Demo

Overview

The Serial Peripheral Interface (SPI) is synchronous serial interface the allows full duplex communication with an external SPI compliant Master or slave device. SPI devices operate in master-slave mode, where master initiates the communication, and can operate in two mode 4 wire or 3 wire mode. SPI specifies following four signal:

- **SCLK** : Serial clock from Master to Slave
- **MOSI** : Serial data from master to slave
- **MISO** : Serial Data from slave to master
- **CS** : Chip select(Only in 4 wire mode)

Application details

The demo application focuses on showing the required initialization sequence to enable the CC3200 SPI module in full duplex 4-wire master and slave mode(s).

Source Files briefly explained

- **main.c** - Initializes the SPI module in either master or slave 4 wire mode based on macro
- **pinmux.c** - Generated by Pinmux utility to mux out the SPI module signal to chip boundary.
- **uart_if.c** - Implements the UART terminal.
- **startup_ewarm.c** - Implements interrupt vector table when using IAR ewarm tool chain
- **startup_css.c** - Implements interrupt vector table when using CC tool chain

Usage

- Interconnect two LP board as shown below, where one LP board acts as SPI master and other as SPI slave.

LP 1 Pin	LP 2 Pin	Signal Name
P1.7	P1.7	SCLK
P2.3	P2.3	CS
P2.6	P2.6	MOSI
P2.7	P2.7	MISO
P2.1	P2.1	GND

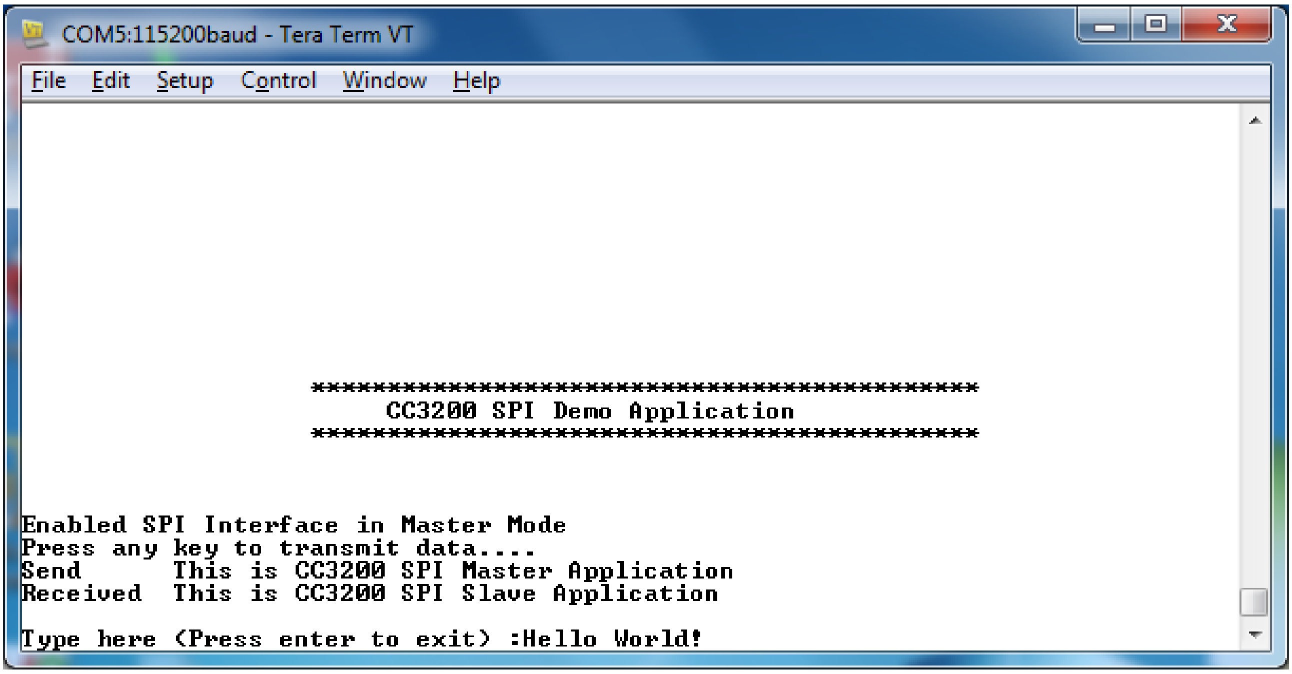
- Setup a serial communication application (HyperTerminal/TeraTerm) with following settings for both the LP(s). For detail info visit Terminal setup

- **Port**: Enumerated COM port
  - **Baud rate**: 115200
  - **Data**: 8 bit
  - **Parity**: None
  - **Stop**: 1 bit
  - **Flow control**: None

- Run the application (by flashing the bin/IAR/CCS) on both the LPs, one compiled for master mode and another for slave mode by changing the following macro definition:

```
//*****************************************************************************
//
// Appplication Master/Slave mode selector macro
//
// MASTER_MODE = 1 : Application in master mode
// MASTER_MODE = 0 : Application in slave mode
//
//*****************************************************************************
#define MASTER_MODE      0
```

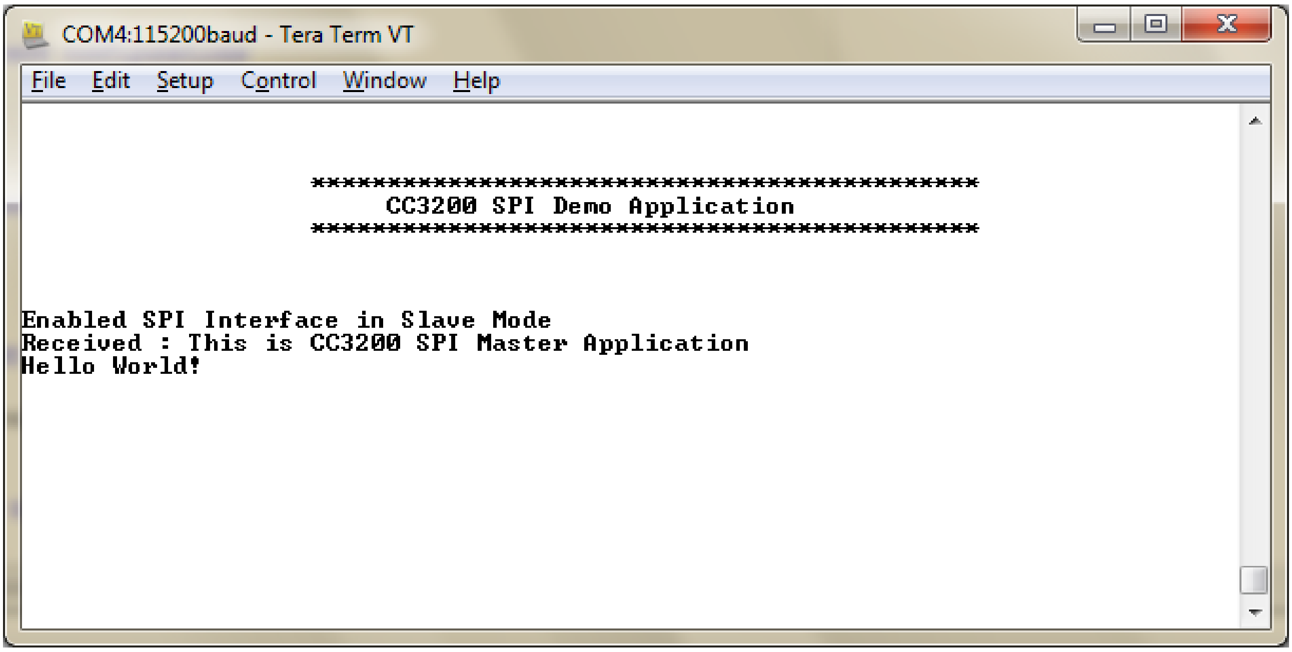
- Follow the instruction on the UART terminal.


COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
                    ********************************************
                         CC3200 SPI Demo Application
                    ********************************************


Enabled SPI Interface in Master Mode
Press any key to transmit data....
Send      This is CC3200 SPI Master Application
Received  This is CC3200 SPI Slave Application

Type here (Press enter to exit) :Hello World!
```


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
                    ********************************************
                         CC3200 SPI Demo Application
                    ********************************************


Enabled SPI Interface in Slave Mode
Received : This is CC3200 SPI Master Application
Hello World!
```

Limitations/Known Issues

- User should always execute master application followed by slave application to avoid slave SPI receiving garbage.

Article Sources and Contributors

CC32xx SPI Demo *Source*: http://processors.wiki.ti.com/index.php?oldid=178068 *Contributors*: Codycooke, Jitgupta, Malokyle

Image Sources, Licenses and Contributors

Image:cc3200_spi_master_terminal.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Cc3200_spi_master_terminal.png *License*: unknown *Contributors*: Codycooke

Image:cc3200_spi_slave_terminal.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Cc3200_spi_slave_terminal.png *License*: unknown *Contributors*: Codycooke