HAL_SPI_Master for KIT_T2G-B-H_LITE

Customer training workshop

Q3 2024

Scope of work

- This code example demonstrates the use of SPI (HAL) resource in the master mode. The SPI master is configured to send command packets to control a user LED on the slave. Both the slave and the master are on the same device.

- **Device**
  - The TRAVEO™ T2G CYT4BF8CDS device is used in this code example

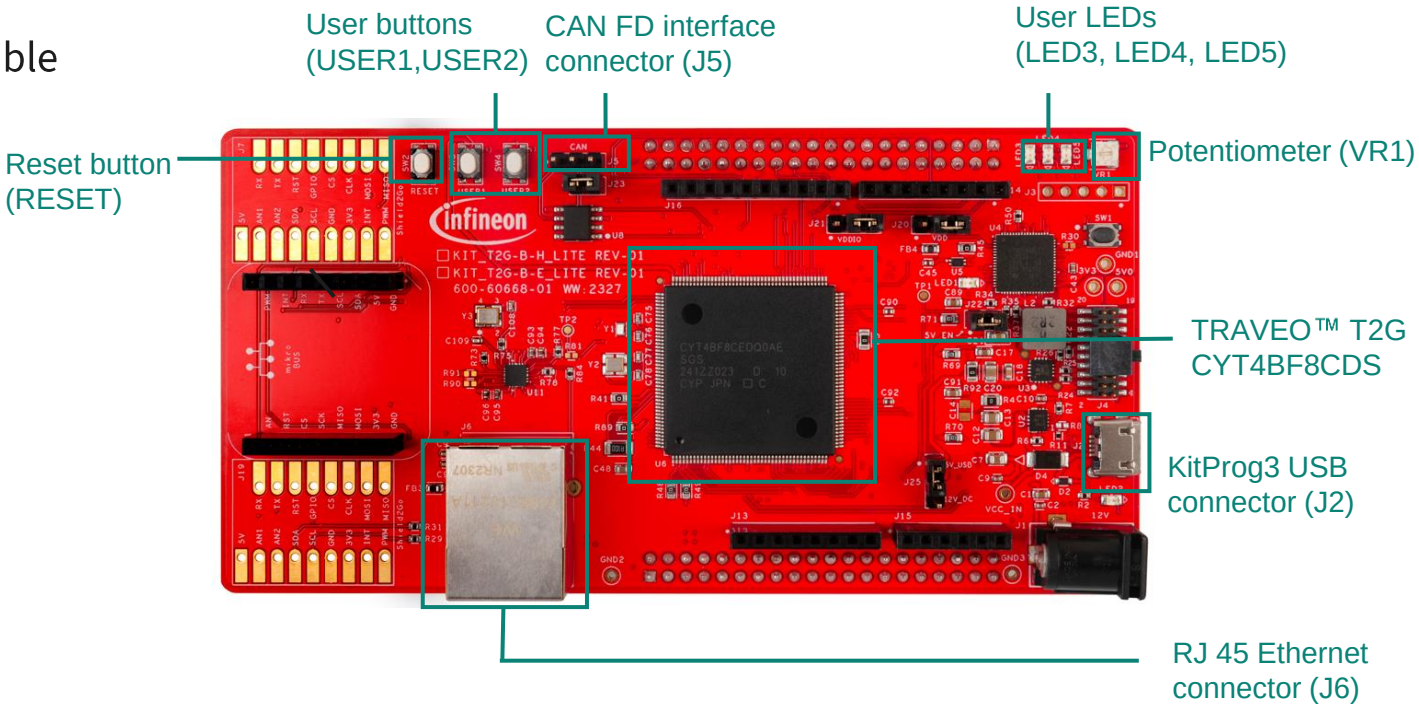
- **Board**
  - The TRAVEO™ T2G KIT_T2G-B-H_LITE board is used for testing

Introduction

- **Serial Communications Block (SCB) has the following features:**
  - Standard SPI master and slave functionality with Motorola, Texas Instruments, and National Semiconductor protocols
  - Standard UART functionality with SmartCard reader, local interconnect network (LIN), and IrDA protocols
  - Standard LIN slave functionality with LIN v1.3 and LIN v2.1/2.2 specification compliance
    The SCB has only standard LIN slave functionality
  - Standard I2C master and slave functionality
  - EZ mode for SPI and I2C slaves; allows operation without CPU intervention
  - CMD_RESP mode for SPI and I2C slaves; allows operation without CPU intervention and is available only on Deep Sleep-capable SCB
  - Low-power (Deep Sleep) mode of operation for SPI and I2C slaves (using external clocking), available only on Deep Sleep-capable SCB
  - Deep Sleep wakeup on I2C slave address match or SPI slave selection; available only on Deep Sleep-capable SCB
  - Trigger outputs for connection to DMA
  - Multiple interrupt sources to indicate status of FIFOs and transfers
  - Local loop-back control

Hardware setup

- This code example has been developed for the KIT_T2G-B-H_LITE board
- Connect the PC to the board using the provided USB cable through the KitProg3 USB connector (J2)



Implementation

- **Code example design**
  - The MCU controls both the master and the slave SPI resources. Configure this example to operate in both master and slave SPI modes to run this example using a single kit if the kit supports two SPI ports on its I/O header.
  - The master sends a packet to the slave with a command to turn ON or turn OFF the user LED1 (LED3). The packets are sent at an interval of 1 second. The slave receives the packet and controls the user LED1 (LED3) per the command.

- **Follow these steps to configure this code example:**
  - STDOUT setting
  - GPIO port pin initialization
  - Configure SPI master / slave
  - Transmitting / Receiving data

- **STDOUT setting**
  - Call the ***cy_retarget_io_init()*** function to use UART as STDOUT
    - Initialize P0.1 as UART TX, P0.0 as UART RX (these pins are connected to KitProg3 COM port)
    - The serial port parameters change to 8N1 and 115200 baud

Implementation (contd.)

- **GPIO port pin initialization**
  - The *cyhal_gpio_init()* function initializes the GPIO port pin once
    - Initialize P5.0 as output (initial level = H, LED turns off)

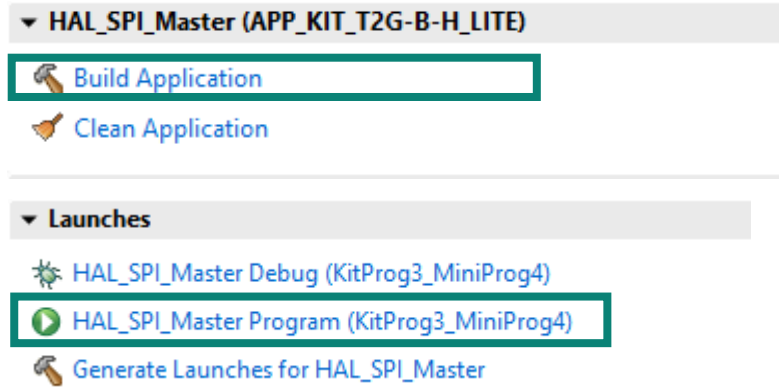
- **Configure SPI master/slave**
  - Call the *cyhal_spi_init()* function to initialize as master / slave
    - The SCB channel to be used is automatically allocated
    - The pins to be used are fixed-based on the board design
      - ***Master : KIT_SPI_MASTER_MOSI / KIT_SPI_MASTER_MISO / KIT_SPI_MASTER_SCLK / KIT_SPI_MASTER_SS***
      - ***Slave : KIT_SPI_SLAVE_MOSI / KIT_SPI_SLAVE_MISO / KIT_SPI_SLAVE_SCLK / KIT_SPI_SLAVE_SS***
  - Call the *cyhal_spi_set_frequency()* function to initialize baud rate of each channel
    - The cycle is changed by SPI_FREQ_HZ

- **Transmitting/receiving data**
  - Call the *cyhal_spi_send()* function to transmit via SPI as master
    - The data to be transmitted specifies the next state so that the user LED (LED3) is toggled
  - Call the *cyhal_spi_recv()* function to receive data from master as slave.
    - The received data is reflected to the user LED (LED3) by calling *cyhal_gpio_write()*

Compiling and programming

1. Connect to power and USB cable
2. Use Eclipse IDE for ModusToolbox™ software for compiling and programming
3. For compilation:
   a. Select the target application project in the Project Explorer
   b. In the Quick Panel, scroll down, and click **Build Application** in HAL_SPI_Master (KIT_T2G-B-H_LITE)
4. Open a terminal program and select the KitProg3 COM port. Set the serial port parameters to 8N1 and 115200 baud.
5. For programming:
   a. Select the target application project in the Project Explorer
   b. In the Quick Panel, scroll down, and click “HAL_SPI_Master Program (KitProg3_MiniProg4)” in the Launches

KitProg3 USB connector (J2)



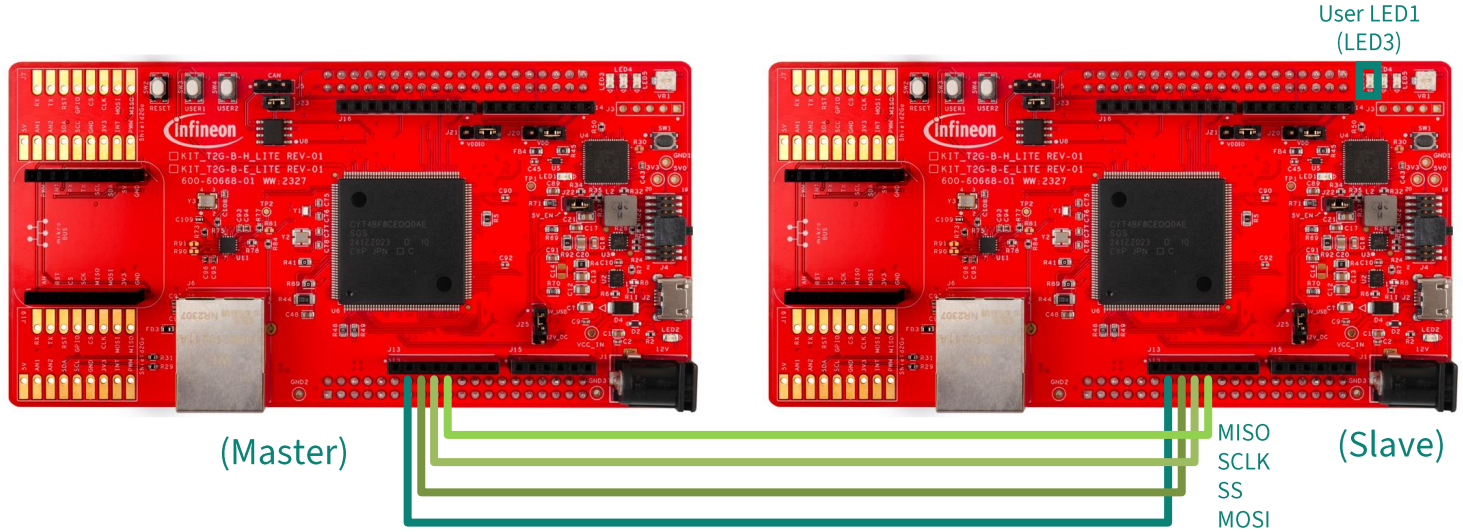
Run and test

After programming, the application starts automatically.

Ensure that the UART terminal for the master/ slave displays as shown in figure.

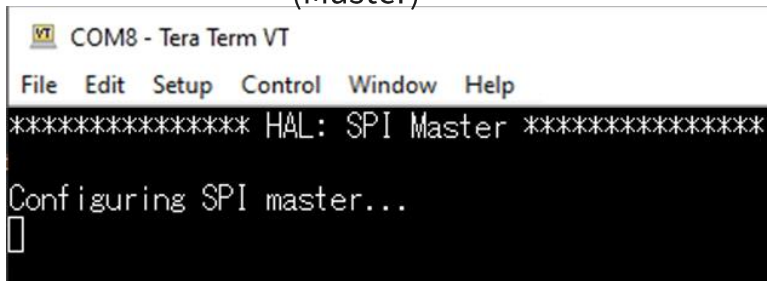
Do the following:

- Connect the jumper wires to the bottom side of both connectors (?) (See the table on the right)
- Observe that the user LED (LED1) of the slave board turns on/off every 1 second

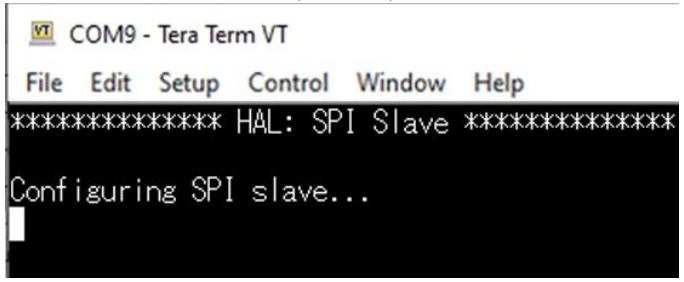


	Master		Slave
MOSI	X1:14	<->	X1:14
MISO	X1:20	<->	X1:20
SCLK	X1:18	<->	X1:18
SS	X1:16	<->	X1:16

(Master)



(Slave)



References

- Datasheet
  - CYT4BF TRAVEO™ T2G 32-bit Automotive MCU based on Arm® Cortex®- M7 dual
- Architecture reference manual
  - TRAVEO™ T2G Automotive MCU body controller high architecture reference manual
- Registers reference manual
  - TRAVEO™ T2G Automotive MCU: TVII-B-H-8M body controller high registers reference manual
- PDL/HAL
  - Peripheral driver library (PDL)
  - Hardware abstraction layer (HAL)
- Training
  - TRAVEO™ T2G training

Revision History

Revision	ECN	Submission Date	Description of Change
**	7781956	2022/07/05	Initial release
*A	7876252	2023/02/20	Updated the title Updated figures in “Compiling and programming”
*B	8080303	2024/10/02	Replaced development board from KIT_T2G-B-H_EVK to KIT_T2G-B-H_LITE