SCB_I2cCommMaster Example Project

1.0

Features

- Communication between I^{2}C master and slave
- Simple packet protocol with command and status byte

General Description

This example project demonstrates the basic operation of the I^{2}C master (SCB mode) component. The I^{2}C master sends the packet with a command to the I^{2}C slave to control the RGB LED color. The packet with a status is read back.

Development Kit Configuration

This example project is designed to be executed on the CY8CKIT-042 from Cypress Semiconductor. A full description of the kit, along with more example programs and ordering information, can be found at http://www.cypress.com/go/cy8ckit-042.

The project requires configuration settings changes in order to run on CY8CKIT-040 from Cypress Semiconductor. A full description of the kit, along with more example programs and ordering information, can be found at http://www.cypress.com/go/cy8ckit-040.

The second kit is required to implement the I^{2}C slave device to communicate with the master. The SCB_I2cCommSlave example project is provided for this purpose. Either CY8CKIT-042 or CY8CKIT-040 can be used. Refer to the SCB_I2cCommSlave example project datasheet for more information.

The SCL and SDA lines of the master and slave have to be tied together. These pins and RGB LED pin location for both kits are summarized in the table below.

Table 1. Pin assignment of the SCL, SDA and RGB LED

Pin Name	Development Kit	
	CY8CKIT-042	CY8CKIT-040
SCL	P3[0]	P1[2]
SDA	P3[1]	P1[3]
LED_GREEN	P0[2]	P1[1]
LED_RED	P1[6]	P3[2]

In order to switch from CY8CKIT-042 to CY8CKIT-040 following steps should be performed:

1. Change the project's device from CY8C4245AXI-483 to CY8C4014LQI-422 with a Device Selector called from the project's context menu.

2. Change the clock configuration. In the Workspace Explorer window, double-click the project's design-wide resource file and click on the **Edit Clocks…** icons on the **Clocks** tab. Set IMO frequency to **32 MHz**.

3. Change the assignment of the pin components to physical pins. In the Workspace Explorer window, double-click the project's design-wide resource file and assign the pins for I²C master and LED accordingly to the Table 1.

Project Configuration

The example project consists of the I²C master (SCB mode) and pin components. The design schematic is shown in Figure 1. The blue annotation components are used to represent the RGB LED installed on the Pioneer Kit. Two pin components are used to control the LED color. The I²C master sends a packet with a command to the slave and reads the back packet with a status every 500 milliseconds.

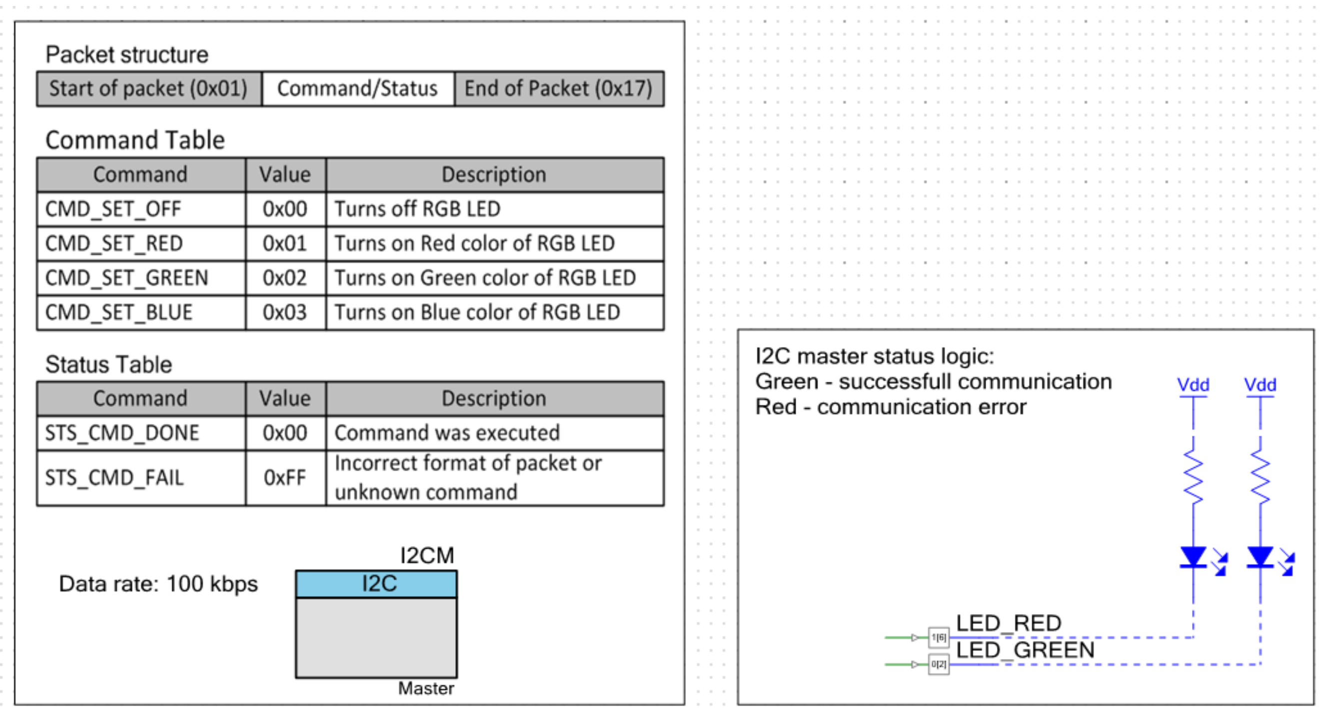

Figure 1. Example project design schematic

The I²C master is configured to operate with the data rate of 100 kpbs. The component configuration window is shown below.

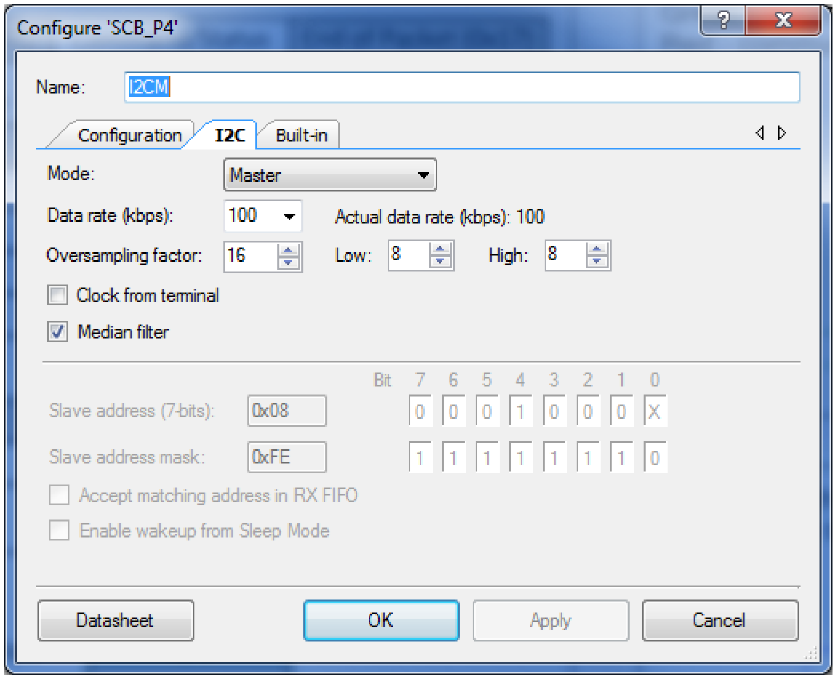

Figure 2. I2C master (SCB mode) component configuration

Project Description

In the main firmware routine, the I^{2}C master component is started. The LED is turned off. Interrupts are enabled to the CPU core as required by the I^{2}C master component operation.

The main loop starts communication with the slave every 500 milliseconds to. The I^{2}C master initializes a packet with a command and setups write transfer. The initial command is CMD_SET_RED. The packet structure and table with commands are shown below. The code polls I2CM_I2CMasterStatus() API for a completion of the write transfer. After the write transfer is completed the errors status bit is checked to update the LED color: green – a successful transfer or red – any error occurred during the transfer. In the case of an error the following read transfer does not take place and the same command will be sent again. Otherwise the I^{2}C master allocates a read buffer and setups the read transfer to receive the status packet. The code polls I2CM_I2CMasterStatus() to complete a read transfer. After the read transfer is completed, the errors status bit is checked to update the LED color: green – a successful transfer and red – any error occurred during the transfer. If an erroroccurs, the same command will be sent again. Otherwise the basic checks of the packet structure are done: the length of the packet, the start and end of the packet byte. The status byte is checked afterwards. If all the checks are successful, the command is considered as executed. The command is updated to the next one. The command sequence is the following: CMD_SET_RED, CMD_SET_GREEN, CMD_SET_BLUE, CMD_SET_OFF, CMD_SET_RED and so on.

The packet structure.

Start of packet (0x01)	Command/Status	End of Packet (0x17)

Table with command constants

Command	Value	Description
CMD_SET_OFF	0	Turns off RGB LED
CMD_SET_RED	1	Turns on Red color of RGB LED
CMD_SET_GREEN	2	Turns on Green color of RGB LED
CMD_SET_BLUE	3	Turns on Blue color of RGB LED

Table with status constants

Status	Value	Description
STS_CMD_DONE	0x00	Command was executed
STS_CMD_FAIL	0xFF	Incorrect format of packet or unknown command

The packets with a command and status are converted into the following I^2C master transfers. The packet with a command has a write direction set in the address byte and the packet with a status has a read direction set appropriately.

Packet with command

S	ADDR = 0x08	W	A	SOP = 0x01	A	Command	A	EOP = 0x17	A	P

Packet with status

S	ADDR = 0x08	R	A	SOP = 0x01	A	Status	A	EOP = 0x17	$\overline{A}$	P

- Master drives the bus - Slave drives the bus

Expected Results

Connect two boards as explained in the Development Kit Configuration section.

Build and program the SCB_I2cCommSlave example project.

The CY8CKIT-042 kit does not provide a pull-up on the I^2C bus unless the Bridge Control panel enables it. If both master and slave examples run on this kit a pull-up has to be enabled. Run the Bridge Control Panel software shipped with the PSoC Creator. Select the KitProg device, which was programmed with SCB_I2cCommSlave example project, from the list of the Connected Ports. Refer to Figure 3. Bridge Control Panel enables pull-ups on I2C bus.

The CY8CKIT-040 kit provides a pull-up on the I^2C bus when USB mini-B cable is plugged-in therefore no extra actions are needed before programming the SCB_I2cCommMaster example project.

Build and program the SCB_I2cCommMaster example project.

Observe that RGB LED changes its color on the kit which serves as an I^2C slave.

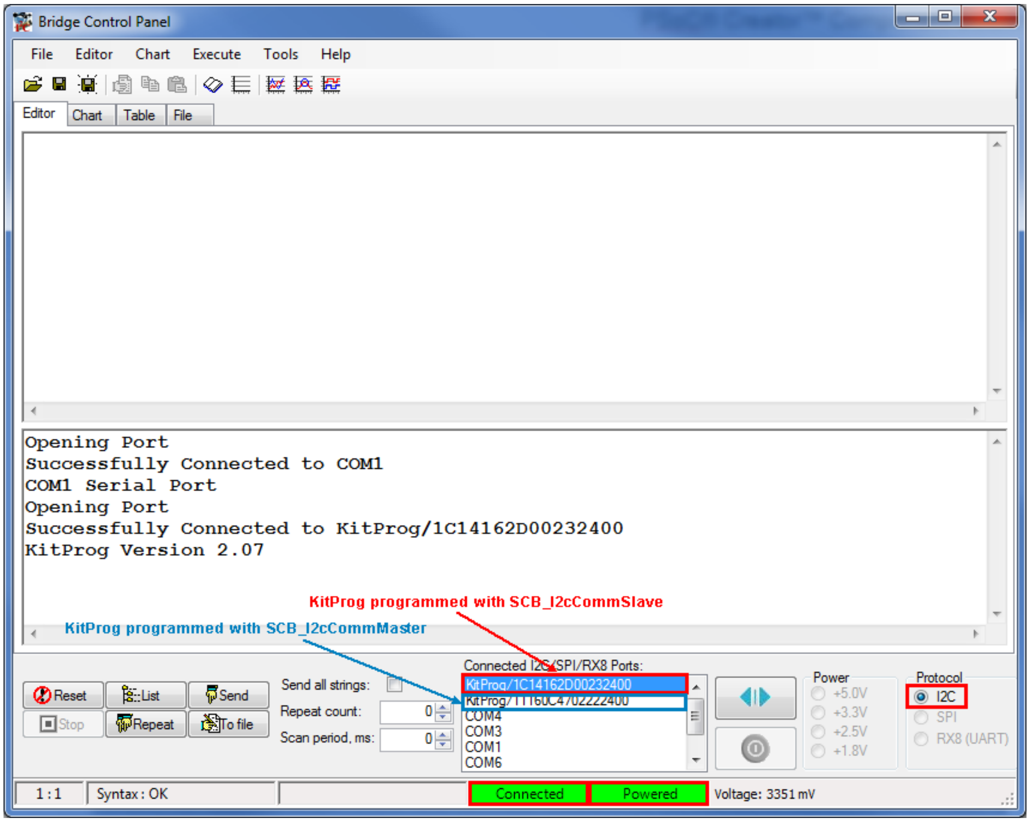

Figure 3. Bridge Control Panel enables pull-ups on I2C bus