**HOMEWORK 7**

**1. I 2C:**

**(a) Describe I 2C.**

**Ans:**

I2C is abbreviated as Inter-Integrated circuit. It consists of two wire bus. It is serial protocol for two wire interfacing.it is used to connect low speed devices. e.g. line A/D, Digital to analog converters, EEPROM’s, microcontrollers or perform communication between the different elements of the chip. This protocol uses start and stop bit to send or stop data transfer. In this protocol more than one master devices can be used in the circuit. It can be added or remove easily from the system without affecting the rest of the system. Talking about its limitation is that it is half duplex communication protocol. All the processing is controlled by system CPU and if number of master devices are increased then the hardware complexities are also increased.

**(b) Describe how I 2C is implemented on the RPi.**

**Ans:**

On raspberry pi the inter-integrated circuit (I2C) is implemented using the Broadcom serial controller. The Broadcom serial controller support 7-bit/10-bit addressing mode and bus frequency up to 400KHz.Some latest controllers have ability to communicate at the frequency of 1MHz but this feature is not present in Raspberry pi. The I2C bus needs pull-up resistors with SCL and SDA lines. These resistors are called termination resistors and have range between 1kohm to 10kohm.Some optional resistors are added to avoid overcurrent and it is must keep in mind that the i2c bus is tolerant and if you want to connect it to 5v then you have to must add some circuitry to make voltage level same on both sides.

**(c) Describe how to enable how to enable the first and second I 2C bus on the RPi.**

**Ans:**

Latest Raspberry pi has two I2C buses. No bus is by default enabled. The primary I2C bus of the raspberry pi is enabled by opening the raspi-config tools. You can just open the terminal and type “sudo raspi-config”. you can go to the “interfacing option” and turn on the I2C bus. The bus will not be enabled at this point you have to restart it for saving system changes and apply new changes to the system. However, there are many ways to enabled the I2C bus. you can also edit boot.config fles in raspberry pi loadable kernel module (LKM).

In order to enabled the second I2C bus you have to edit the /boot/cmdline.txt of the raspberry pi to include the text, “bcm2708.vc\_i2c\_override=1”.Keep in mind that you are only allow to use the second I2C bus if you are not using HATs because the second I2C bus are automatically configured for the HATs. After editing the /boot/cmdline.txt of the raspberry pi restart the raspberry pi and now you have two I2C buses.

**2. Describe and give examples of how to use the Linux i2c-tools package.**

**Ans:**

The raspberry pi bios is Linux based and its kernel is also Linux based.so Linux provide a set of tools which are called I2C tools. These tools are used to connect with other devices by using I2C interface. These tools include a chip dumper, bus probing tool and register level access helpers. these tools are installed in raspberry pi by typing command “sudo apt install i2c-tools” in the terminal.

The example of using Linux I2C tool package is given below. Let an I2C device is going to interface with raspberry pi through I2C protocol so the steps are follows:

**i2cdetect:**

The first step in this process is to detect the whether the device is present or not. When both I2C are enabled you can check the device detected by using command “i2cdetect -l”.

**i2cdump:**

This command is used to read the values of the register of the device attached with I2C bus and display the result in hexadecimal block format.

**i2cget:**

The I2Cget command is used to test the device by reading the value of the register can be used to test the input for Linux shell script.e.g. in order to read the number of seconds on the clock type the command “i2cget -y 1 0x68 0x00”. so, in result you can see the number of seconds.

**3. Complete the I 2C and WiringPi section in chapter 8, page 327. Copy the results of the test for your answer and give a picture/video of your program as it executes.**

**4. Serial Peripheral Interface (SPI):**

**(a) Describe SPI.**

**Ans:** Serial peripheral interface (SPI) is synchronous serial communication. It is an interface bus commonly used to send data between the microcontrollers and shift register, SD cards, sensors etc. Serial peripheral interface is completely flexible for bit transformed. This protocol is not limit to 8-bit word. The maximum high speed in SPI protocol enables high speed that is more than I2C (inter integrated circuit). This protocol is also faster than asynchronous serial. It has duplex communication feature and also support multiple slaves.

**(b) Give a comparison of I 2C and SPI on the RPi.**

|  |  |
| --- | --- |
| **I2C** | **SPI** |
| I2C protocol has 2 wires in which up to 128 addressable devices are attached. | SPI protocol has 4 wires and it require additional logic if more slave device is added. |
| The speed of data rate in I2C is up to 400KHz.It uses half duplex communication. | Its speed is high (32MHz) and faster performance. It uses full duplex communication. |
| When attached with the hardware pull-up resistor is required. | When attached with the hardware no pull-up register is required. |
| I2C have multiple masters. slave has addresses acknowledge transfer and control the data flow. | SPI is very simple and easy. Only one master device is present while no slave device is required. |
| Intermittently accessed devices, e.g., RTCs, EEPROMs. | For devices that provide data streams, e.g., ADCs. |

**(c) Describe how to enable the SPI bus on the RPi.**

**Ans:**

The PSI bus on the raspberry pi is not enabled by default. You have to make this feature enabled. There are different way to enabled this. Open the terminal and type “sudo raspi-config”. Now go to interfacing option and turn ON the SPI bus. Restart you pi will start the system with new changes. or you can add few lines in /boot/config.txt in the boot file of the raspberry pi.

**5. Complete the SPI and WiringPi section in chapter 8, page 345. Copy the results of the test for your answer and give a picture/video of your program as it executes.**