**EMBEDDED SYSTEM DESIGN**

**LAB ASSIGNMENT -8**

**INTERFACING I2C-EEPROM**

**FOR LPC2148 MICROCONTROLLER**

**SUBMITTED BY- SANCHIT AGRAWAL (2016H140106)**

**ABHINAV GARG (2016H140103)**

**OBJECTIVE**:

To Store the external interrupt counts in I2C serial EEPROM and update it continuously. To display the current value of count on the LCD display and display should be refreshed fast enough to see count in real time. The system should be able to retain the last count of the switch press even after power is lost and should start with the last updated value of count when the power resumes to the system.

**PROJECT WORKSPACE**

* Used keil u-vison5..
* Under ARM section, select the board as LPC 2148.
* Incude all the header files and the source files that are required for the program
* Workspace consists of all the library files.
* Then, after ARM programming, select the target device that is to be programmed and burn the bin file into the device.
* Provided on chip communication using I2C protocol.

**CRITICAL ISSUES:**

* Real time change of count when the switch is pressed fast  External interrupt generation on switch debouncing.
* Retain of count after 256.
* To increase count after 256.

.

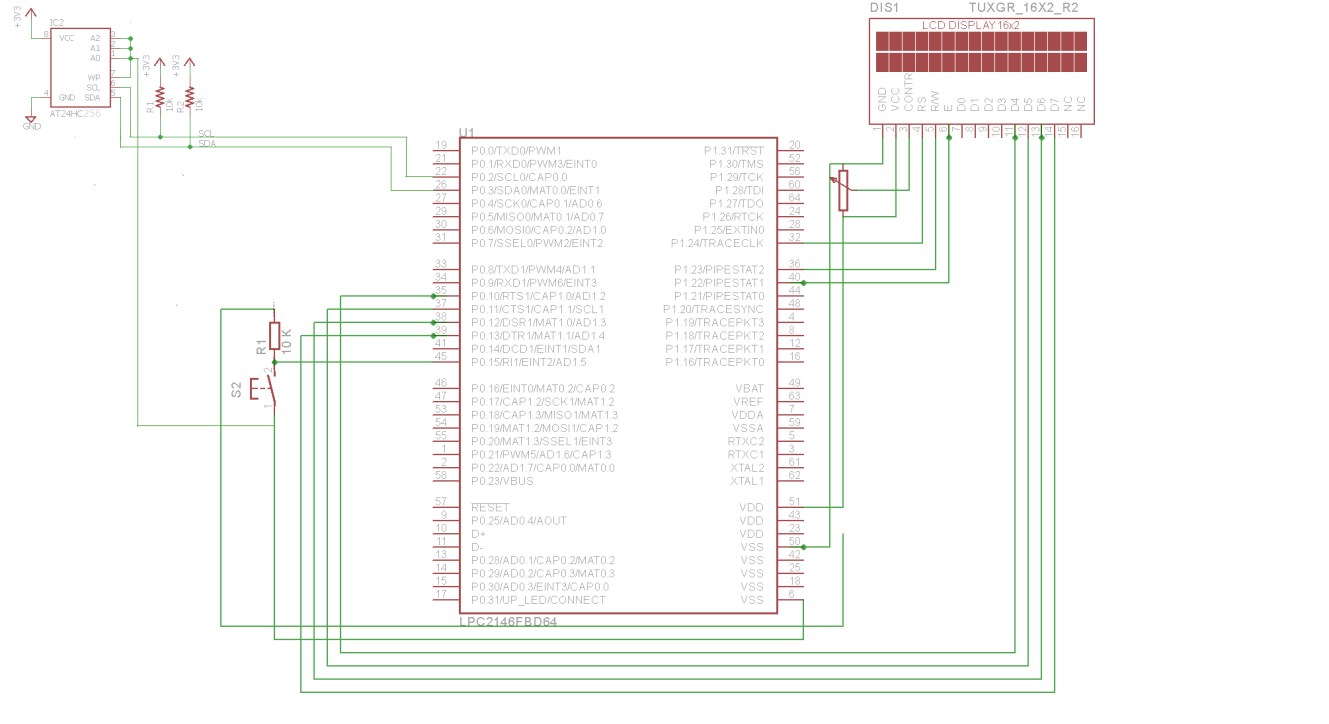
**REGISTERS:**

The code involves registers of I2C etc. The registers of I2C are as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Registers | | | |  | | Function | | | | | | | | | |
|  | **I2C0CONSET** | | |  | |  | This register control the setting of bits | | | | | | | |  |
|  | | | in the I2CON register that controls operation of the I2C interface. Writing a one to a bit of this register causes the corresponding bit in the I2C Control Register to be set.  Writing a zero has no effect | | | | | | | |
|  | **I2C0CONCLR** | | |  |  |  | This register control clearing of bits | | | |  | | | | |
|  | | |  | in the I2CON register the controls operation of the I2C interface. Writing a one to a bit of this register causes the corresponding bit in the I2C control register to be cleared.  Writing zero has no effect.  I2C0CONCLR contains following  control bits | | | |
|  | **I2C0STAT** | |  |  | |  | During I2C operation this register | |  | | | | | | |
|  | | provides detailed status codes that allow software to determine the next action needed. | |
|  | **I2C0DAT** |  | |  | |  | During master or slave transmit |  | | | | | | | |
|  | mode, data to be transmitted is written to this register. During master or slave receive mode, data that has been received may be read from this register. |
|  | **I2C0ADR** |  | |  | |  | This register is readable & writable, | | | | |  | | | |
|  | and is only used when I2C interface is set to slave mode. In master mode, this register has no effect. I2C0ADR | | | | |
|  | | |  | | |  | contains the 7-bit slave address for | | |  | | | | | |
| operation of the I2C interface in slave mode. The least significant bit (LSB) determine whether a slave respond to the general call address | | |
|  | **I2C0SCLH** | |  | | |  | This register determines the high time | | | | | | |  | |
|  | | of the I2C Clock (contains the SCL high duty cycle count) | | | | | | |
|  | **I2C0SCLL** | |  | | |  | This register determines the low time | | | | | |  | | |
|  | | of the I2C Clock. (Contains the SCL low duty cycle count). I2C0SCLL and I2C0SCLH together determine the clock frequency generated by an I2C master and certain times used in slave mode. | | | | | |

**HARDWARE DESIGN**:

LPC2148 is required as the hardware. The **schematic** is given below.



Schematic diagram showing interfacing of EEPROM with LPC2148

**SOFTWARE DESIGN:**

* Firstly, we compiled the I2C initialization code and made it as header file which is then included in the main program. The I2C initialization code consists of I2C protocols and port pins.
* Used I2C0 for operation. Power on I2C0 peripheral by PCONP register.

Using PINSEL0 we defined port pin as SDA and SCL.

* Bits of I2CON register are cleared by I2C0CONCLR register. I2EN is set by the I2C0CONSET register. I2C Clock Duty Cycle (high and low) is done by the I2C0SCLH and I2C0SCLL registers.
* Then, code for LCD is separately compiled and included as header file in the main program. It contains all the initialization commands and the function to pass the string to the LCD.
* Functions I2C\_ReadFromEEPROM and I2C\_WriteToEEPROM are defined and used to read and write the count from EEPROM respectively
* A separate header file is created for the interrupts in which functions init\_ext\_interrupt(void) and irq void Ext\_ISR(void) are defined.
* These were used to initialize the external interrupt and interrupt service routine routine.
* The count value is incremented in the isr routine. The interrupt was falling edge sensitive and it occurred on Pin P0.15.
* In the main function firstly, all the respective function that were defined and initialised earlier are called.
* Then, external interrupt was enabled when the switch is pressed.
* Each time the switch is pressed, the count value is incremented by one.
* The updated value of count written in the EEPROM using I2C\_WriteToEEPROM .
* The count value is read from the EEPROM using I2C\_ReadFromEEPROM and displayed on the LCD.

**OBSERVATION:**

The count from 0 to 65536 is displayed and updated on the LCD when the switch is pressed. The correct value is retained after the power off.

**RESULT:**

The correct count value was shown in the LCD when the switch is pressed and value is retained even after power off.

**REFERENCES:**

[www.ngx.com](http://www.ngx.com/) [www.nxp.com](http://www.nxp.com/)

arm reference manual (UM10139)