**Introduction**

This project is a reflection of a data logging system. Data loggers have very importance in the industry. Data logging systems continuously obtain data from different sensors after a specific period of time and store that data. User can collect that data from the data logger at any time. So that he could take appropriate measures. Aim of this project is to design a simple data logging system to get command on I2C, USART and ADC interface of the microcontroller. Microcontroller (16F877A) continuously tracks time with the help of real time clock (DS1307). After every 10 seconds it collects data from 1 potentiometer and 4 buttons and stores it in its RAM. Microcontroller is connected to the computer through USART. There are two control buttons. One button is to set date of the RTC and other is to collect the data stored in the RAM of the microcontroller.

**Description**

The aim of this project is to create a data logger. The microcontroller (16F877A) has I2C interface with Real Time Clock and serial interface with PC through USART. The Real Time Clock has its own power source. When the power to the micro-controller goes down, the power to the real time clock remains on and the real time clock continues to track the time. In this system 4 buttons/switches and 1 potentiometer are used to replace sensors. Every 10s the system checks the status of the 4 buttons/switches and the potentiometer and stores the status information in the PIC16f877a RAM. The data logger has two control buttons,

**Reset button:** When reset button is pressed, it causes an interface with a PC using USART to enter a prompt requiring someone to enter a new date. The system stays in this state until new date is not entered.

**Retrieve button:** When it is activated, microcontroller stops the polling of the real time clock and retrieves all the status data from the PIC16f877a RAM. The data from the RAM is then written to a monitor via the USART.

Circuit diagram of Data logging system is shown in Fig 1. Pin connections of buttons with microcontroller are listed below,

PIN RA0 is connected to the potentiometer

PIN RA1 is connected to switch/button 1

PIN RA2 is connected to switch/button 2

PIN RA3 is connected to switch/button 3

PIN RA4 is connected to switch/button 4

PIN RD0 is connected to “Reset” button

PIN RD1 is connected to “Retrieve” button



Fig 1: Circuit Diagram

**Software**

**Flow Chart**



Fig 2: Flow Chart

**Pseudo Code**

START PROGRAM

Declare variables

Initialize Ports

Initialize ADC

Initialize USART

Initialize I2C

Set initial time and date of RTC

Store output from Potentiometer on RAM

Store status of 4 input Buttons on RAM

Store initial time of RTC

WHILE True

Read time and date from RTC

Process received data

Subtract present time from reference time and find time difference

IF time difference > = 10 seconds THEN

Store output from Potentiometer on RAM

Store status of 4 input Buttons on RAM

Reset the time difference

Refresh the reference time

END IF

IF RD0 = = pressed THEN

Prompt the user to set date of RTC

Process received data

Set date of RTC

END IF

IF RD1 = = pressed THEN

Print the potentiometer value stored on RAM

IF button\_1 = =pressed THEN

Print “Button 1 is close”

ELSE

Print “Button 1 is open”

END IF

IF button\_2 = =pressed THEN

Print “Button 2 is close”

ELSE

Print “Button 2 is open”

END IF

IF button\_3 = =pressed THEN

Print “Button 3 is close”

ELSE

Print “Button 3 is open”

END IF

IF button\_4 = =pressed THEN

Print “Button 4 is close”

ELSE

Print “Button 4 is open”

END IF

END IF

END WHILE

END PROGRAM

**Simulation**

Initially the Real time clock is set to Time: 00-00-00 and Date: 00-00-00. We can change the initial value of the Real Time Clock in the code. It could be any value. The I2C debugger window shows that all the data transmission is occurring smoothly. This is illustrated in Fig 3.



Fig 3: Initialization

When the Reset button is pressed, it prompts the user to enter date in the format DD/MM/YY. If the user does not follow the format, circuit will not operate properly and will lead to garbage values. The entered date is then sent to the Real Time clock through I2C link. This is illustrated in the Fig 4.



Fig 4: When Reset button is pressed

When the Data Retrieve button is pressed, the microcontroller sends the data stored in its RAM related to the potentiometer and the 4 switched to the computer screen through USART link. In the Fig 5, it can be seen that the data sent to the virtual terminal is exactly the reflection of that of the states of the potentiometer and switches on the circuit board.



Fig 5: When Retrieve Data button is pressed

**Conclusion**

Objective of this project is to help us learn I2C interface and develop our programming skills. The data logging system is a simple and inexpensive solution to the collection of distant and spatial field data. One advantage of data logging system is its ability to collect data automatically for a long period of time. This system does not need any operator after activation. It can be integrated with different sensors and can be used in the hazardous environment to collect data without any risk. In this way we can get an accurate and comprehensive picture of the subject environment. It has already been employed extensively on the industrial scale. The cost of this system is decreasing gradually with time.