Complex Engineering Problem (CEP) & Lab Project

EE223 (Spring 2025)

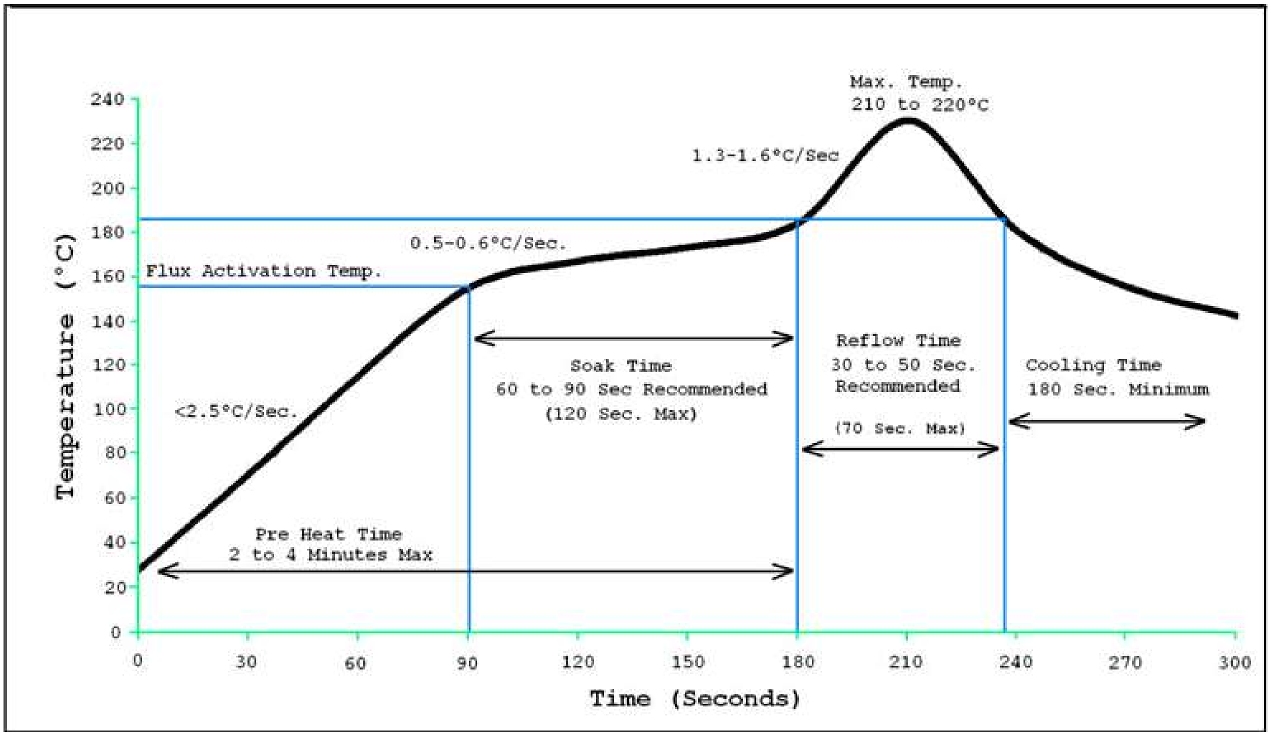
Microprocessor Systems

Introduction

8-Bit microcontrollers are most widely used in embedded systems for consumer electronics, automation, robotics, automobiles, display drivers, etc. Microchip is the world's leading manufacturer of microcontrollers and has a major market share in most of the mentioned industries. Among the end application areas, consumer electronics including washing machines, microwave ovens, air conditioners, thermostats, IoT based home automation systems, refrigerators, displays, digital clocks, food processors, climate controllers, home and lab ovens, automobiles, etc. constitute the major portion. Solder reflow ovens are one of the applications where microcontroller based embedded systems are used to control the overall working of the oven.

Problem statement

You are required to design a complete embedded control system for a basic solder reflow oven. The general temperature-time curve of the oven is given below:



https://www.itechsmt.com/blogs/news/each-temperature-zone-of-smt-reflow-oven

First, do a literature review of the actual functions of how a solder reflow oven oven works, what are the components in its embedded control system, what are its features, how many minimum inputs and outputs are required, what are the limitations, etc. Then, start designing the system yourself including the interface circuitry etc. The interface circuitry should be designed based on realistic requirements of the oven. While selecting the microcontroller, please make sure that the said microcontroller is available for simulation in the Proteus Software version that you have. PIC18F47K42 is not available in the simulation environment and so cannot be used for this CEP.

Objectives:

Students are required to:

1. **(CLO-1; PLO-1 & PLO-12; C-2)** Understand the working and features of a solder reflow oven and the basic interfacing circuitry. Select a suitable microcontroller from PIC18 family of microcontrollers with suitable features and peripherals to implement the desired embedded system. Understand the architecture, features, and peripherals of the selected microcontroller. Also select suitable interfacing circuit components that will be used to design the rest of the circuitry.
2. **(CLO-2; PLO-2; C-3)** Develop a code to implement the desired system in bare-metal Embedded C programming language using MPLabX IDE for the selected microcontroller.
3. **(CLO-3; PLO-3; C-4)** Design and simulate the interfacing circuit to fully implement the complete system. Run the code on the simulation and verify the results. Make corrections in the code and/or circuit design in case of undesired results.
4. **(CLO-2L; PLO-5; P-4)** Implement the circuit of hardware and test its working and output. Demonstrate the actual working system's output to your instructor.
5. **(CLO-4L; PLO-4; C-4)** Debug and rectify the hardware in case of undesired results.
6. **(CLO-6L; PLO-10; C-3)** Write a detailed report showing the design and analysis, algorithms, flow charts, circuit diagrams, simulation results, hardware results/output, and discussion on your results (both simulation and hardware)

Requirements:

1. You must use ADC, CCP/ECCP/CWG, UART, SPI, & Interrupt modules in your system.
2. The temperature sensor should be a digital sensor, not analog sensor.
3. The real-time temperature vs time curve should be plotted on a GLCD or serial terminal plotter on a computer.
4. The temperature control must be based on PID controller and not just an on/off or a proportional controller.
5. There must be a heater to increase the temperature and a fan to decrease the temperature, both of which must be controlled using PWM.
6. You may use transistors as switches to control the fan and heater. You may use a small (40-60 Watt) DC heating element and a small DC motor for the task.
7. If you cannot achieve a temperature as high as the one given in the curve above, you may scale

it down by a maximum factor of 3 but keep the slopes and times unchanged. You have to demonstrate the correct working of your system by overlaying the given curve and your actual output curve to check for the error. The overlayed images must be included in your reports and your project/CEP presentations.

8. You will be required to submit the following files in soft form as CEP submission:
   a. Report
   b. Code (.C file)
   c. Proteus Project (.pdsprj file)