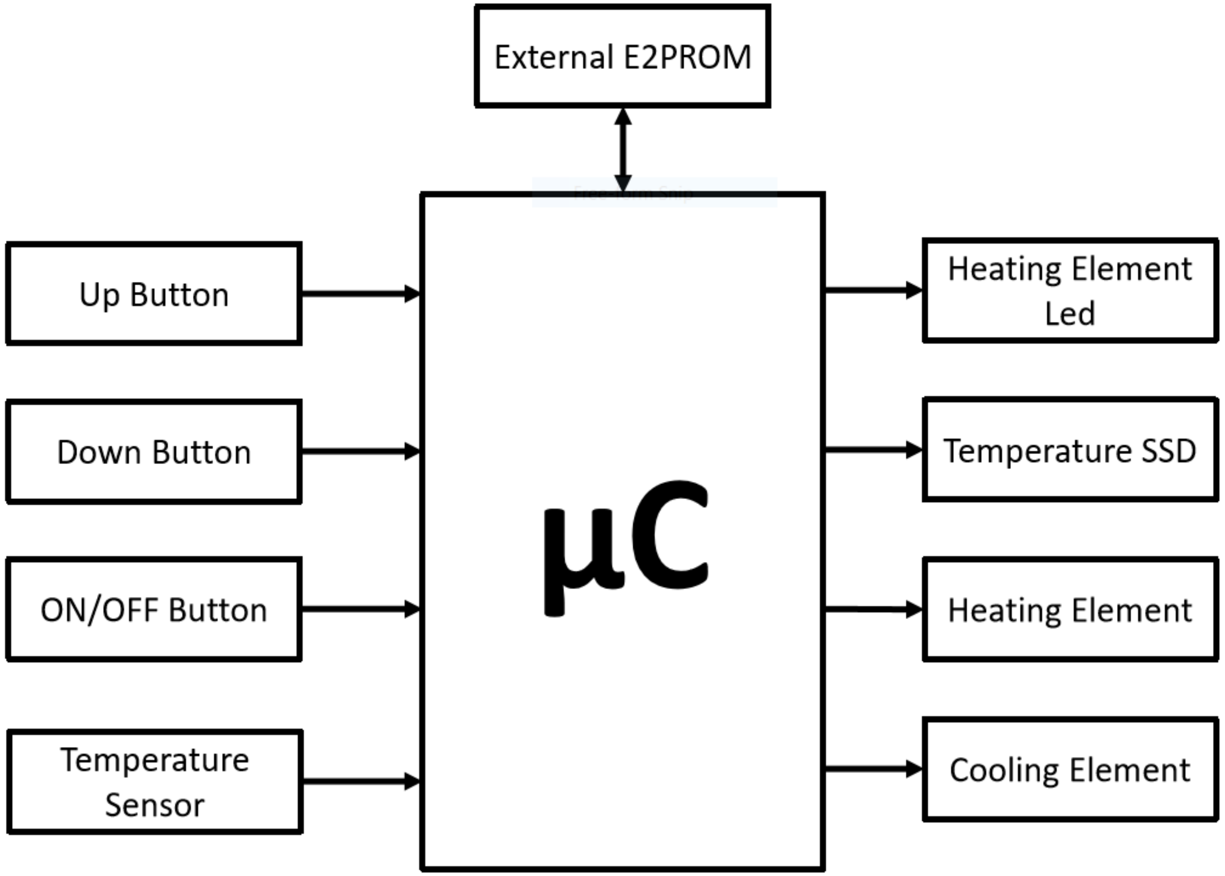
Electric Water Heater Project

Outline:

- Project Idea and System Diagram
- Specifications
- Time Triggered Design
- Os(Scheduler)
- Application Tasks
- Project Modules
- Software/Hardware Components Used

Project Idea : This Project demonstrates the idea of Electric Water Heater. The Project is designed based on “ Time Triggered Design “ .

System Diagram :



Specifications :

- Up and Down Buttons are represented as Push buttons to raise or decrease the temperature
- On/OFF Button is represented as Push Button to turn ON/OFF the Heater
- Temperature Sensor is used to sense the Current Temperature of the Water
- External EEPROM is used to store the Set Temperature
- Heating Element Led is used to indicate if the Heating/Cooling Elements are operating
- Temperature SSD is used to display the Temperature
- Heating Element(Heating Resistance) is used to heat the Water
- Cooling Element(Fan) is used to Cool the Water

Time Triggered Design :

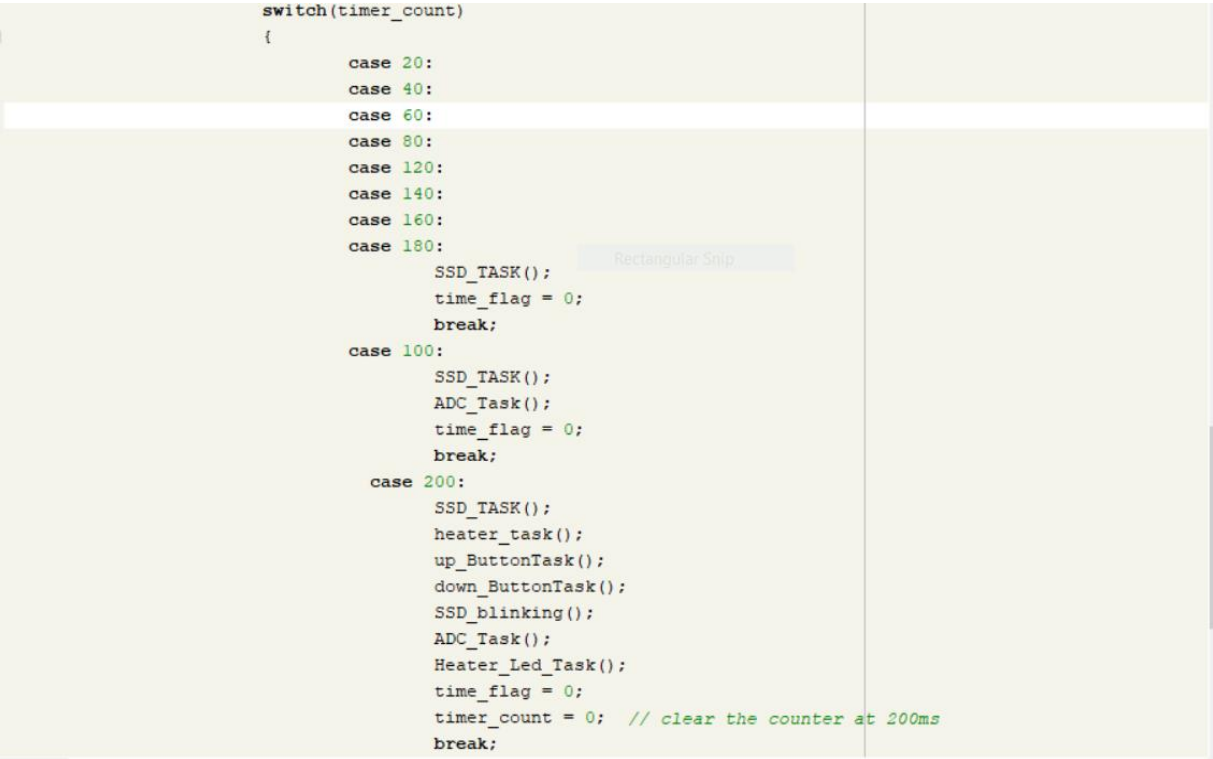
It is a design where Tasks within the system follow a statically computed schedule (i.e., they are allocated time slots during which they can take place) and thus by nature are predictable.

The System perform tasks based on a base time(certain time). In This Project I have designed a scheduler that works with a base time of “20ms“ where the CCP module(Compare mode) fires an interrupt every 20ms to perform the System Tasks depending on their time Analysis and the time of each task. After reaching 200ms the counter of the time will be cleared and so on.

- Note: The System has some shared resources among the files but I ensured that these resources will not be corrupted as my design is not an interrupt based and there is no interrupts in my design except the timer interrupt which will not corrupt any of the shared resources

OS (Scheduler) :

To implement the Scheduler I used the CCP module(Compare mode).The scheduler schedule the tasks based on a Timing analysis and computed schedule of a base time equals 20ms.I have chosen 20ms as I have to display the temperature on 2 SSD at the same time so I have to close one of them and open the other quickly where the human eyes couldn’t see that action. So 20ms means 50HZ and this is a frequency which cant be realized by the naked eye. After that I have to choose a certain time at which all of the tasks will terminate which is 200ms so at this instance I will clear the timer counter and start counting again every 20ms and so on. We will talk later about every task and its timing.



```
switch(timer_count)
{
        case 20:
        case 40:
        case 60:
        case 80:
        case 120:
        case 140:
        case 160:
        case 180:
                SSD_TASK();
                time_flag = 0;
                break;
        case 100:
                SSD_TASK();
                ADC_Task();
                time_flag = 0;
                break;
          case 200:
                SSD_TASK();
                heater_task();
                up_ButtonTask();
                down_ButtonTask();
                SSD_blinking();
                ADC_Task();
                Heater_Led_Task();
                time_flag = 0;
                timer_count = 0;  // clear the counter at 200ms
                break;
```

Application Tasks :

- Init_Task(): Task to initialize the ADC and (GIE/PIEIE) bits (Performed once when system starts)
- SSD_Task(): Task to display the Temperature on the 2 SSD (Performed every 20ms)
- up_ButtonTask(): Task to Raise the Temperature in the Setting mode (Perfomed every 200ms)
- down_ButtonTask(): Task to Decrease the Temperature in the Setting mode (Perfomed every 200ms)
- Heater_Led_Task(): Task to operate the Led depending on the Heating/Cooling elements (Perfomed every 200ms)
- ADC_Task(): Task to get Readings from the Temperature Sensor to control the Water temperature(every 100ms)
- heater_Task(): Task to turn ON/OFF the Electric Water Heater (Perfomed every 200ms)
- SSD_blinking(): Task to blink the SSD when the system is in the Set Mode (Perfomed every 200ms)

CCP Module(GPT) Functions:

void Timer1_start(uint8 my_tick) : Function to start the timer1 which is used in the CCP Module(Compare Mode).It takes one parameter which is the number of ticks of the timer

Os Module Functions :

- void Os_start(void) : Starts the CCP Module(Timer) and the Scheduler
- void Os_scheduler(void) : The System Scheduler
- void Os_newTimerTick(void) : Increments the Timer Counter with the Base Time

EEPROM Module Functions :

- void EEPROM_init(void) : Initializes the I2C Bus which is used to Send and Receive Data from the External EEPROM

- void EEPROM_writeByte(unsigned short my_data,uint8 add) : Function Used to write data to the External EEPROM. It takes 2 parameters which are the address of the memory to store data and the data to be stored

- uint8 EEPROM_readByte(uint8 add) : Function Used to Read Data from the EEPROM. It takes one parameter which is the address of the memory to read data and It returns the data in an unsigned char variable

Note : I have Used I2C built in functions in MikroC Compiler to implement the Driver of the External EEPROM

Helper Modules:

- Std_types.h : It is a module that contains the Standard Types used in this Project, the typedef of the NULL Pointer and some other values.

Software Components Used :

- MikroC Compiler
- PicsimLab Simulator

Hardware Components Used :

- PIC16F877A Microcontroller
- PicGenios Board
- 2 SSD
- 3 Push Buttons
- Led
- Heating Element (Heating Resistance)
- Cooling Element (Fan)
- External EEPROM
- Temperature Sensor(LM35)