

GRADUATION PROJECT ELECTRIC WATER HEATER

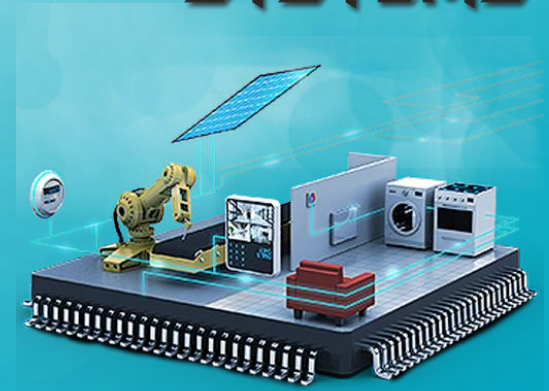
PRESENTED BY:

ENG/OMAR MOHAMED

ENG/YOUSSEF ASHRAF

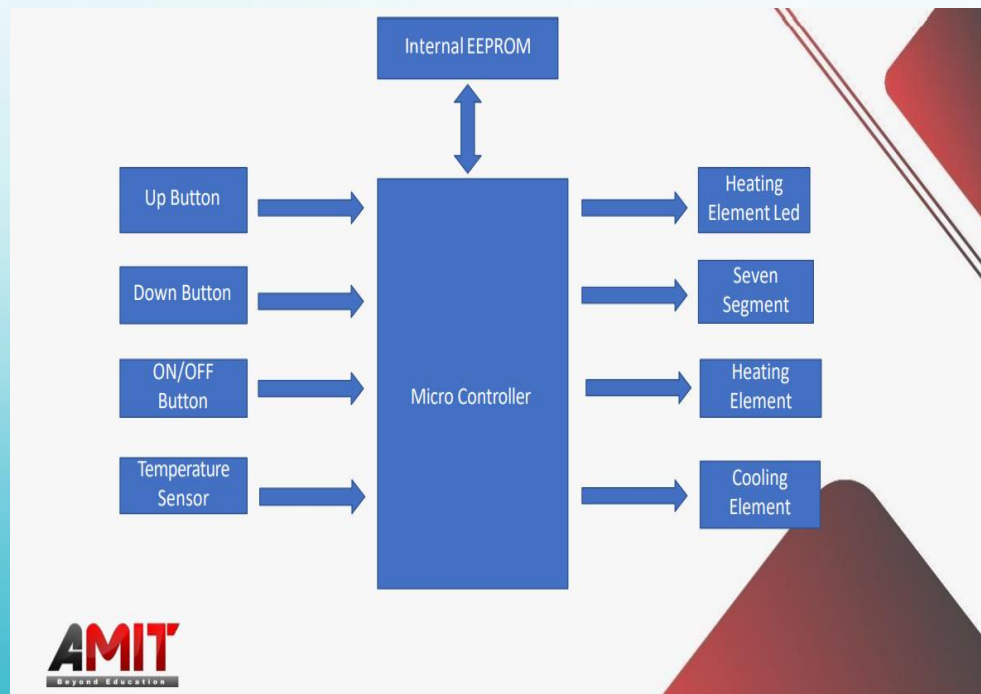
ENG/ABDELRAHMAN AHMED

**EMBEDDED
SYSTEMS**

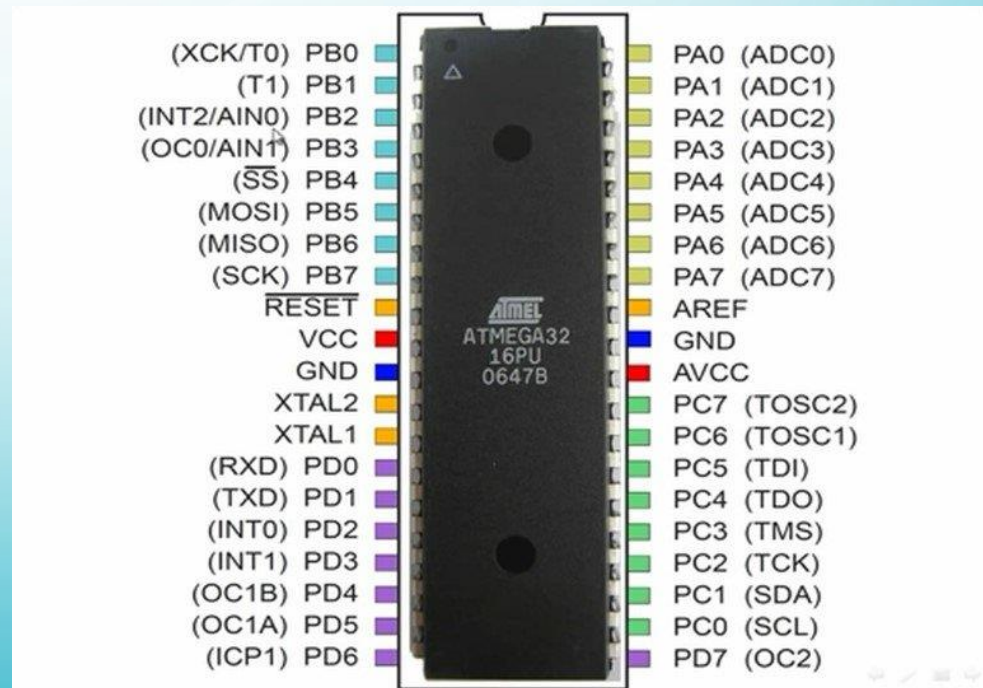


INTRODUCTION

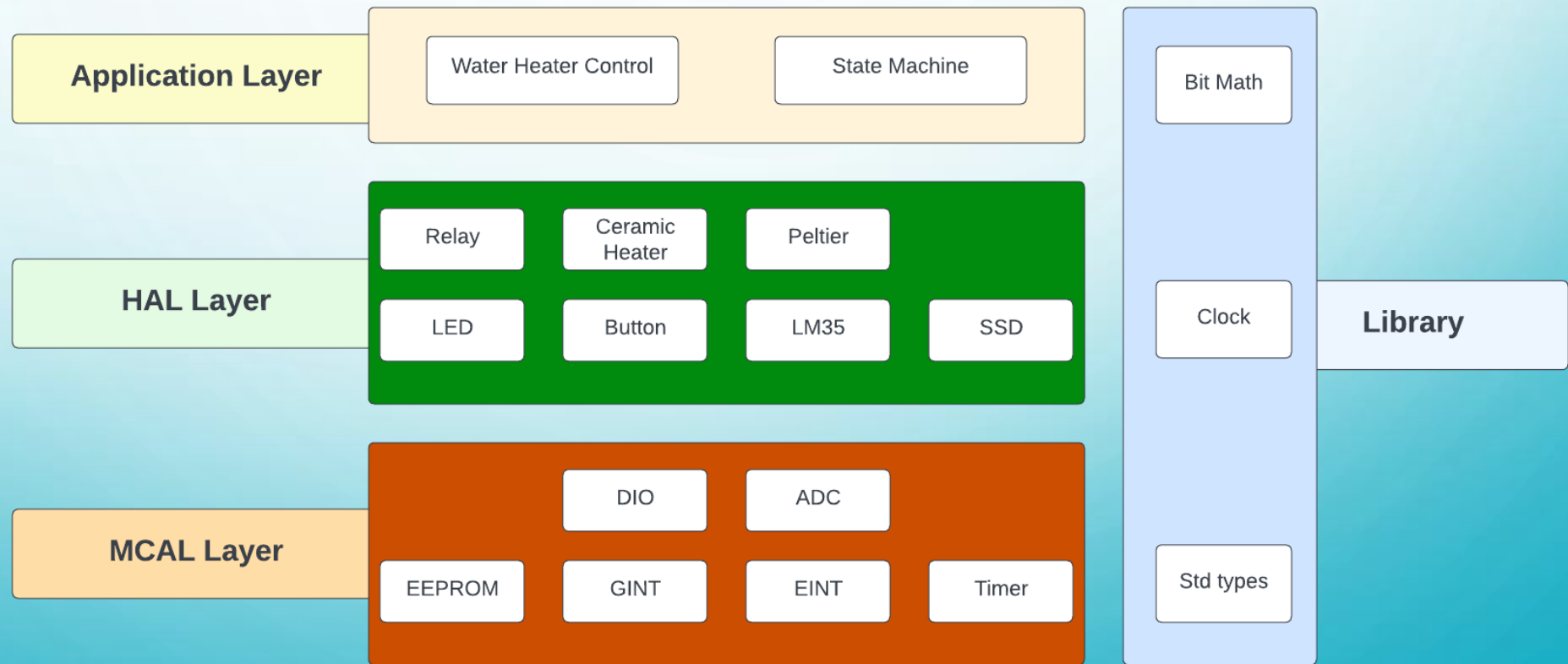
- COMPONENTS



- MICROCONTROLLER (ATMEGA32)



LAYERED ARCHITECTURE



MODULES API SPECIFICATION – MICROCONTROLLER ABSTRACTION LAYER

• DIO:

```
void DIO_Init (void);  
void DIO_WriteChannel (DIO_ChannelType ChannelId, STD_LevelType Level);  
STD_LevelType DIO_ReadChannel (DIO_ChannelType ChannelId);  
void DIO_ToggleChannel (DIO_ChannelType ChannelId);  
uint8 DIO_ReadPort (DIO_PortType PortId);  
void DIO_WritePort (DIO_PortType PortId, uint8 PortValue);  
void DIO_ConfigureChannel (DIO_ChannelType ChannelId, DIO_DirType Direction);
```

• ADC:

```
void ADC_Init(void);  
int16 ADC_Read(ADC_ChannelType ChannelId);
```

• EEPROM:

```
void EEPROM_write(uint16 uiAddress, uint8 ucData);  
uint8 EEPROM_read(uint16 uiAddress);
```

• Timer:

```
/**      TIMER 0 MODULE APIS      */  
void Timer0_Init(Timer_PrescalerType prescaler, uint8 Value, boolean InterruptEnable);  
void Timer0_Start(Timer_PrescalerType prescaler);  
void Timer0_Stop (void);  
void Timer0_SetPreload(uint8 preloadValue);  
void __vector_11(void) __attribute__ ((signal)); /*ISR Function Prototype for TMR0 OVF*/  
void Timer0_SetCompareValue(uint8 compareValue);  
void __vector_10(void) __attribute__ ((signal)); /*ISR Function Prototype for TMR0 CTC*/  
uint8 Timer0_GetPreload(void);  
void Timer0_EnableInterrupt(void);  
void Timer0_DisableInterrupt(void);  
void Timer0_SetCallBack ( void(*Ptr2Func)(void) );  
void Timer0_PWM(uint16 freq, uint8 dutyCycle);  
  
/**      TIMER 2 MODULE APIS      */  
void Timer2_Init(Timer_PrescalerType prescaler, uint8 Value, boolean InterruptEnable);  
void Timer2_Start(Timer_PrescalerType prescaler);  
void Timer2_Stop (void);  
void Timer2_SetPreload(uint8 preloadValue);  
void __vector_5(void) __attribute__ ((signal)); /*ISR Function Prototype for TMR2 OVF*/  
void Timer2_SetCompareValue(uint8 compareValue);  
void __vector_4(void) __attribute__ ((signal)); /*ISR Function Prototype for TMR2 CTC*/  
uint8 Timer2_GetPreload(void);  
void Timer2_EnableInterrupt(void);  
void Timer2_DisableInterrupt(void);  
void Timer2_SetCallBack ( void(*Ptr2Func)(void) );  
void Timer2_PWM(uint16 freq, uint8 dutyCycle);
```

• GINT: `void GINT_Enable_AllInterrupts();`
`void GINT_Disable_AllInterrupts();`

• EINT: `void Ext_INT_Enable(ExtINT_Type INTId);`
`void Ext_INT_Disable(ExtINT_Type INTId);`
`void Ext_INT_SnControl(ExtINT_Type INTId, ExtISC_Type mode);`
`void Ext_INT_SetCallBack (void(*Ptr2Func)(void));`
`void __vector_1(void) __attribute__ ((signal)); /*ISR Function Prototype for Ext INT0 */`

MODULES API SPECIFICATION – HARDWARE ABSTRACTION LAYER

- **Button:**

```
void Button_Init(void);  
uint8 Button_getState(DIO_ChannelType ButtonNo);
```

- **LED:**

```
void LED_Init(void);  
void SetLED_ON (DIO_ChannelType ChannelId);  
void SetLED_OFF (DIO_ChannelType ChannelId);  
void Flip_LED (DIO_ChannelType ChannelId);
```

- **SSD:**

```
void SSD_Init(void);  
void SSD_WriteNum(uint8 FirstDigit, uint8 SecondDigit);
```

- **Relay:**

```
void Relay_Init(void);  
void Relay_switchON(DIO_ChannelType RelayNo);  
void Relay_switchOFF(DIO_ChannelType RelayNo);
```

LM35:

```
void LM35_Init();  
int16 LM35_Read(ADC_ChannelType ChannelId);
```

Ceramic Heater:

```
void Heater_Init(void);  
void Heater_SetMode(Heater_ModeType HeaterMode);
```

Peltier:

```
void Peltier_Init(void);  
void Peltier_SetMode(Peltier_ModeType PeltierMode);
```


MODULES API SPECIFICATION – APPLICATION LAYER

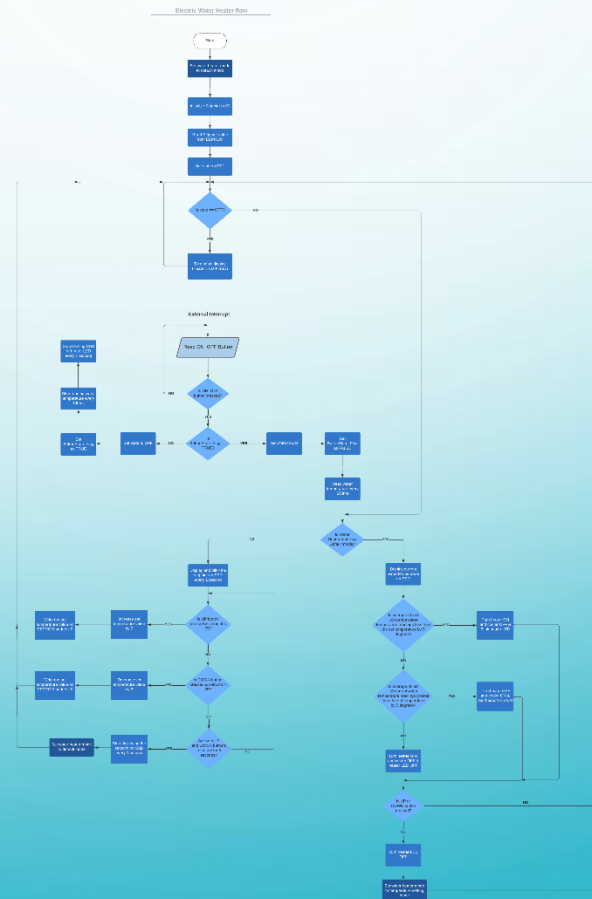
- **STATE MACHINE:**

```
void SM_Init(void);  
void SM(void);  
void Change_WaterHeater_State (void);
```

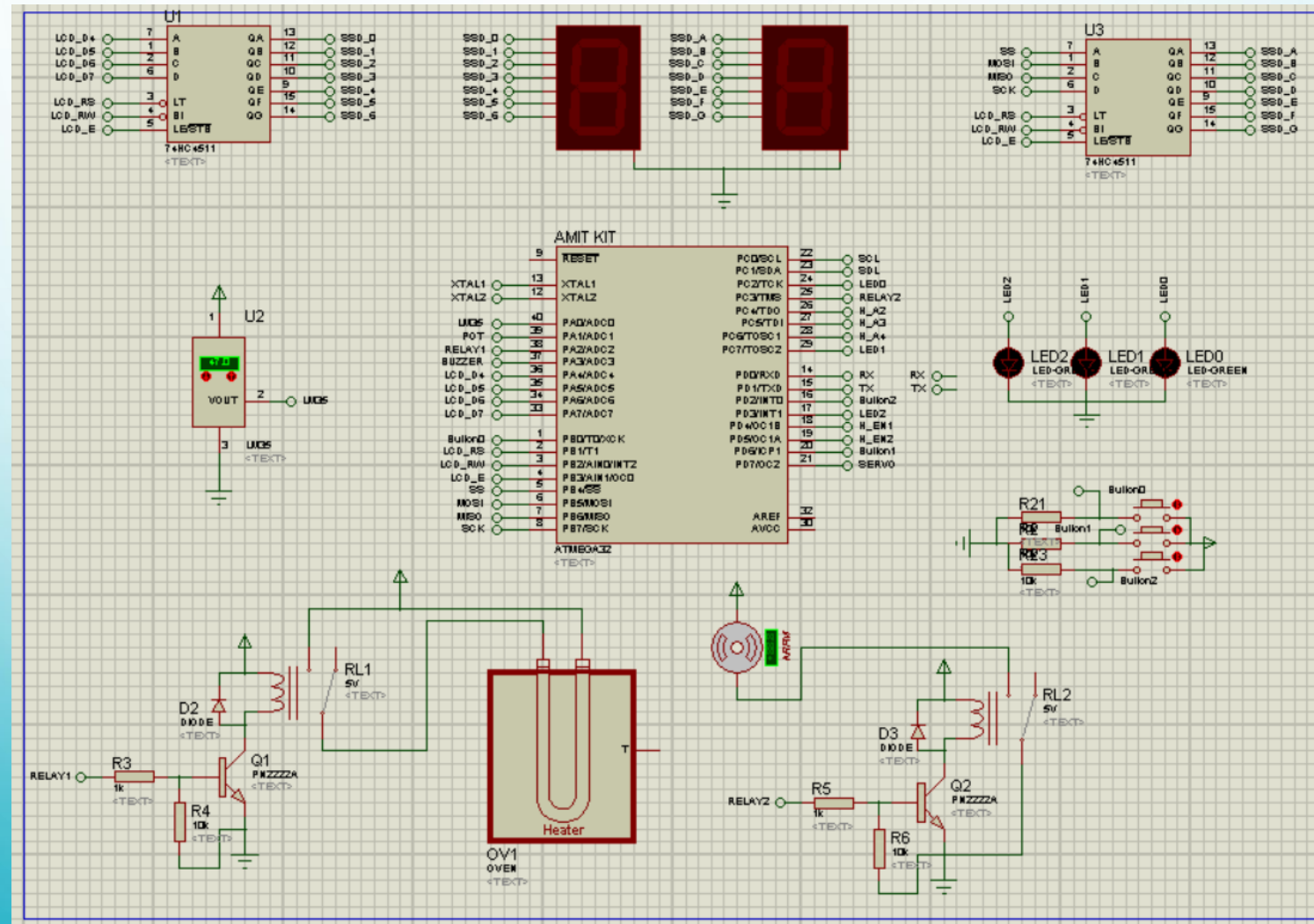
- **WATER HEATER CONTROL:**

```
void WaterHeater_Control_Init(void);  
void WaterHeater_Control(void);  
void Read_WaterTemperature(void);  
void Blink_SetPointTemperature_HeaterLED(void);  
void Stop_WaterHeater_Operation(void);
```

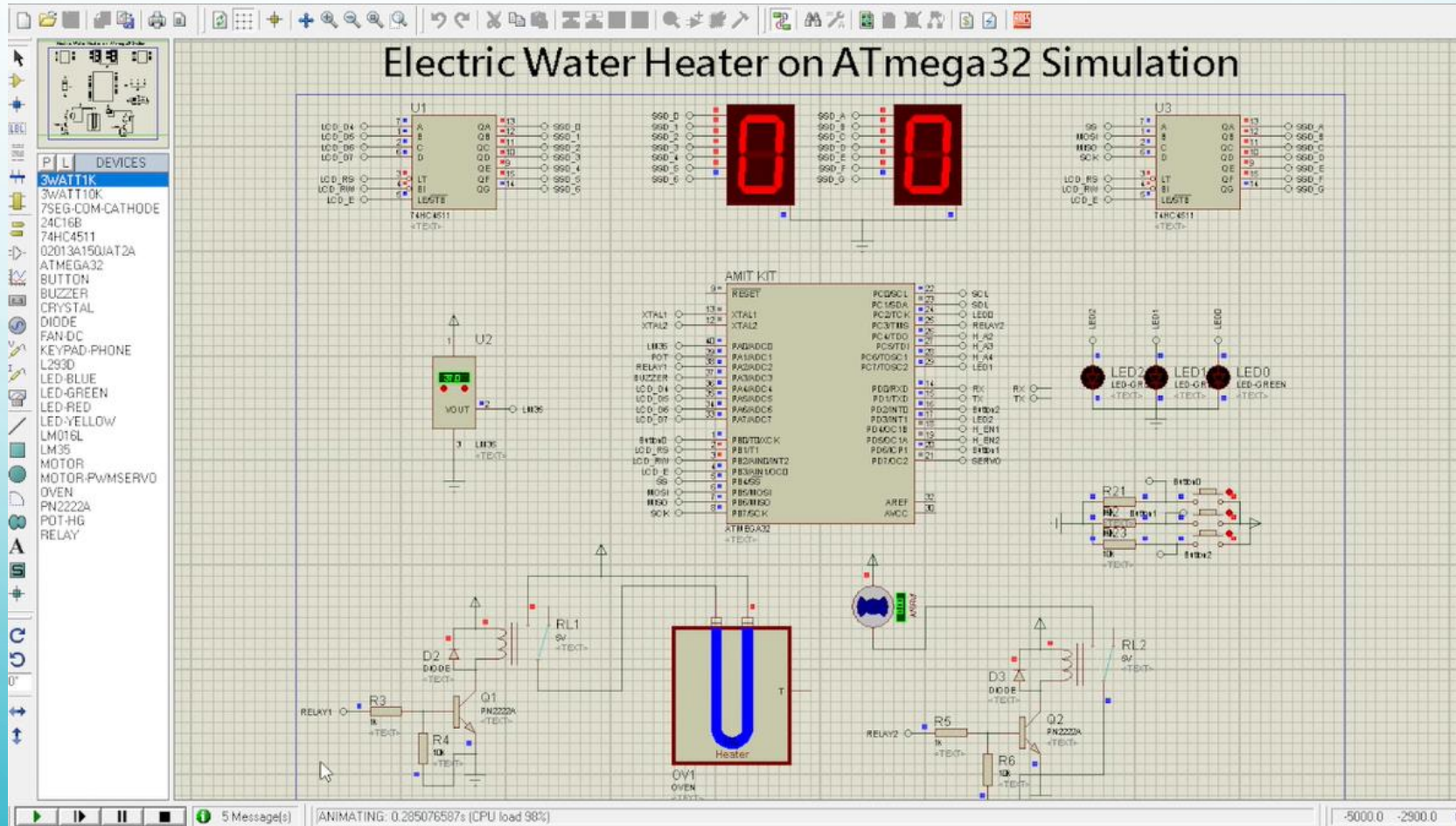
OPERATING FLOW CHART DIAGRAM



PROJECT LAYOUT



SIMULATION



PROBLEMS ENCOUNTERED

- Button Debouncing and Invalid Set Temperature Modification

```
if( Button_getState(UP_Button) == BUTTON_PRESSED || Button_getState(DOWN_Button) == BUTTON_PRESSED )
{
    Timer2_Stop();
    LED_Blink_Flag = FALSE;
    StartLEDTimer_Flag = TRUE;
    SetLED_OFF(Heater_LED);

    WaterHeater_mode = TEMPERATURE_SETTING_MODE;
    StartSSDTimer_Flag = TRUE;

}
else
{}

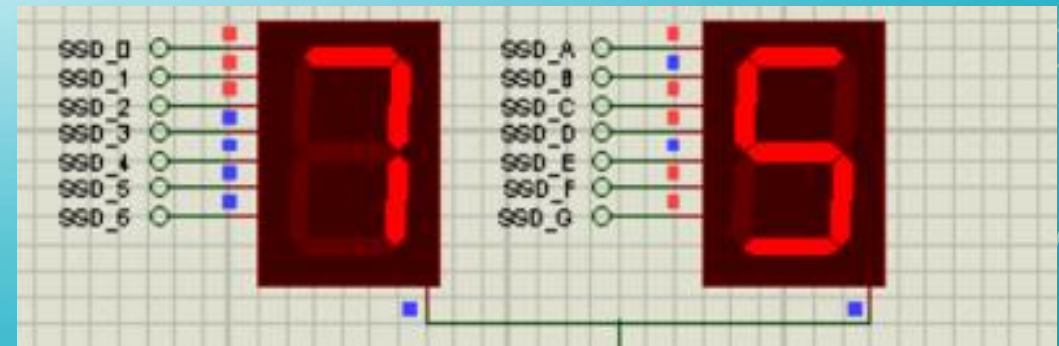
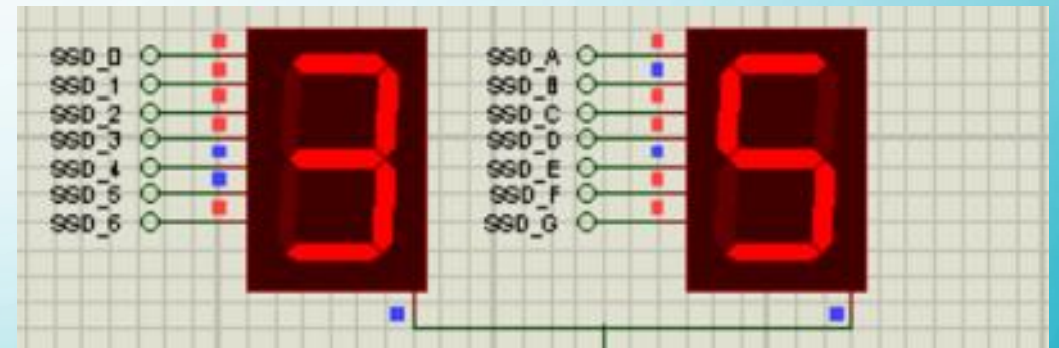
break;
}

case(TEMPERATURE_SETTING_MODE):
{
    if(StartSSDTimer_Flag == TRUE)
    {
        Timer2_Start(No_Prescaler);

        SSD_Blink_Flag = TRUE;
        StartSSDTimer_Flag = FALSE;
    }
    else {}

    if( Button_getState(UP_Button) == BUTTON_PRESSED && Setpoint < 75 )
    {
        Setpoint += TEMP_SETPOINT_OFFSET;
        EEPROM_write(EEPROM_SP_Address, Setpoint);

        ButtonsReleased_Counter = 0;
    }
    else if( Button_getState(DOWN_Button) == BUTTON_PRESSED && Setpoint > 35 )
    {
        Setpoint -= TEMP_SETPOINT_OFFSET;
```



The background features a blue gradient with faint, concentric circular patterns. In the corners, there are decorative line art elements resembling circuit boards or neural networks, with lines and small circles.

ANY QUESTIONS...?



THANK YOU!