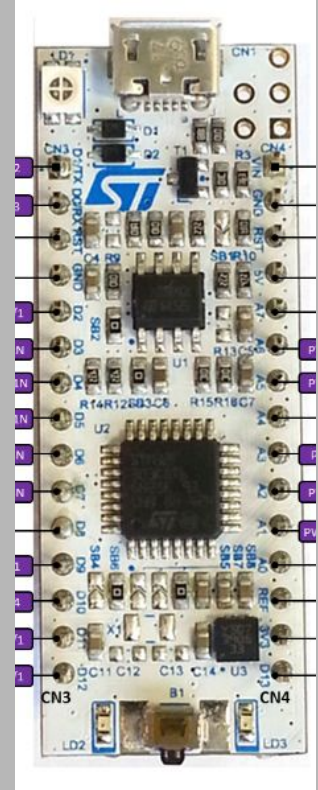
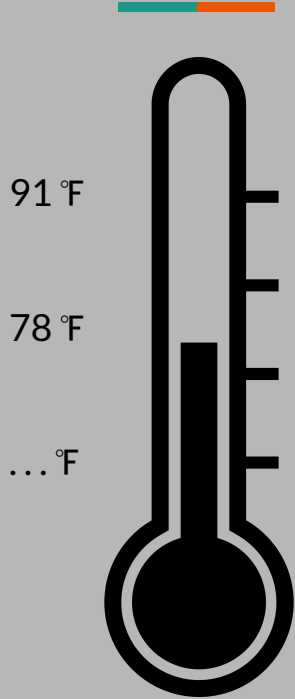


EEE 174 - CPE 185 FINAL PROJECT

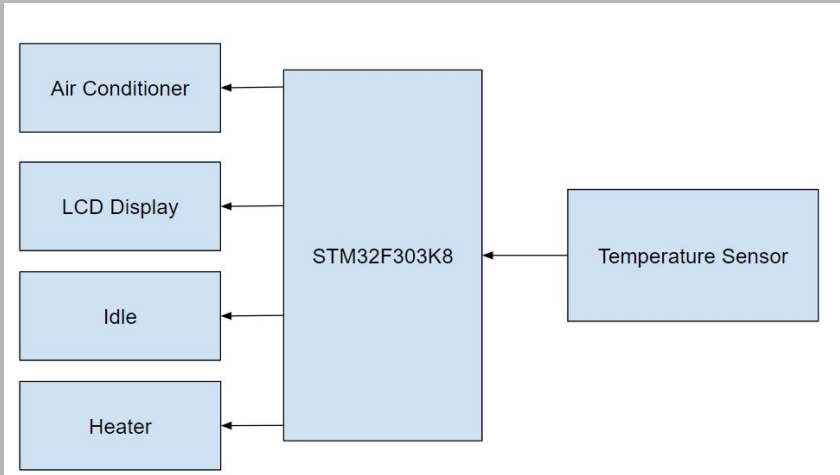
TEMPERATURE STABILIZER

TEAM MAGIC SPARKLES

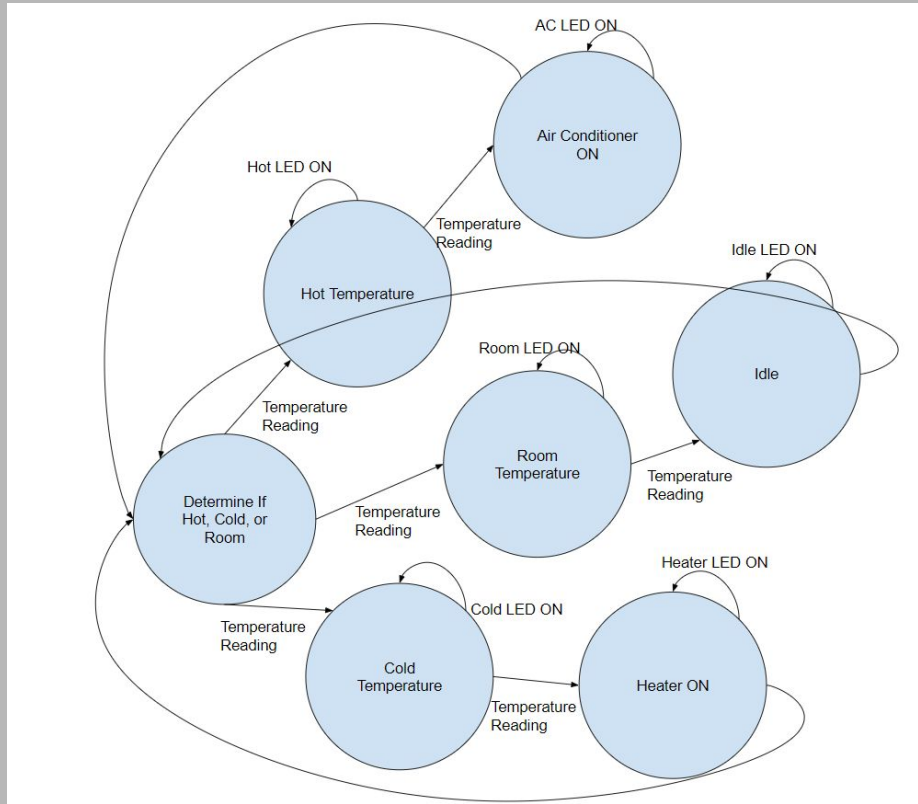
JOSHUA RIVERA, ANTHONY NGO, & NATALIA PEREZ



BACKGROUND



Block Diagram (See above)



Finite State Machine (See above)

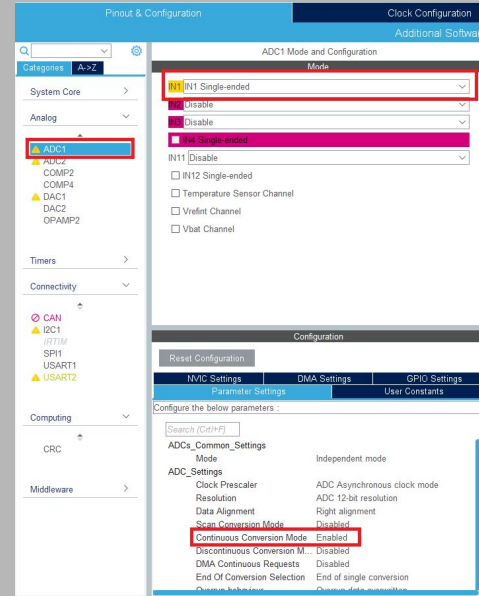
TEMPERATURE SENSOR

+ **Analog (ADC1 mode)** configurations in order to retrieve raw voltage values and continuously convert them to the appropriate values

- ADC channel
- Varies upon the environment

+ **UART2** to transmit the captured and calculated values to the **MobaXterm** interface from the **STM32F303K8** microcontroller and the sensor itself

- Printed as floats (Fahrenheit, Celsius, etc.)
- sprintf and character array



ADC Configurations (See above)

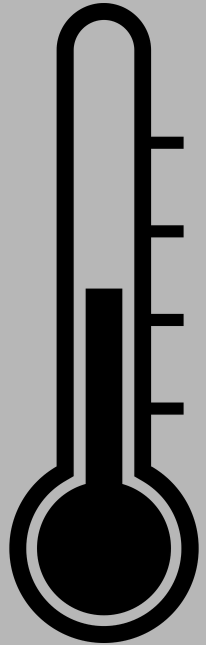
```
/* The following lines are used to compute the voltage to temperature units.*/  
HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);  
rawValue = HAL_ADC_GetValue(&hadc1);  
voltage = ((float)rawValue)/4095 * 3.3; // voltage = (rawVoltage/(2^bit resolution-1)*Vref)  
tempCel = (voltage - 0.5) * 100;  
tempF = 1.8*(tempCel) + 32.0;
```

Conversion Code (See above)

91 °F

78 °F

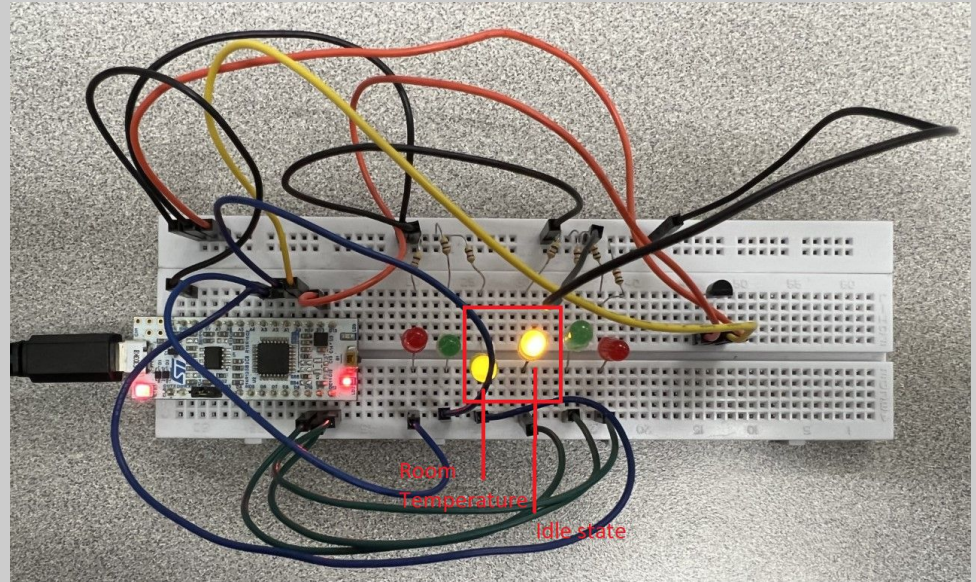
... °F



```
DETERMINING...  
RawValue: 891  
Voltage: 0.718022  
Celsius: 21.802198  
Fahrenheit: 71.243958
```

```
ROOM  
RawValue: 887  
Voltage: 0.714799  
Celsius: 21.479856  
Fahrenheit: 70.663742
```

```
ROOM - IDLE  
RawValue: 899  
Voltage: 0.724469  
Celsius: 22.446888  
Fahrenheit: 72.404396
```



Room temperature and Idle State LEDs on. Terminal shows the output in Fahrenheit, Celsius, and Raw Voltage. (See above)

AIR CONDITIONER

- + Temperature sensor determines temperature of the room.
- + When temperature is greater than 78 °F, the AC will be prompted to turn on.
 - Red LED will turn on indicating room is hot.
 - Green LED will turn on indicating AC is on.

```
/* If the temperature is determined to be hot, the LEDs resembling the air conditioner (AC)
 * and hot temperature will turn on, indicating the environment must be cooled down to
 * room temperature. The AC LED is green. */
TempState AirConditionerHandler(float tempF)
{
    HAL_GPIO_WritePin(GPIOA, HOT_RED_LED_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, ROOM_YELLOW_LED_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, COLD_GREEN_LED_Pin, GPIO_PIN_RESET);

    HAL_GPIO_WritePin(GPIOA, AC_GREEN_LED_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, HEATER_RED_LED_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, IDLE_YELLOW_LED_Pin, GPIO_PIN_RESET);

    HAL_Delay(2000);

    return DetermineState;
}

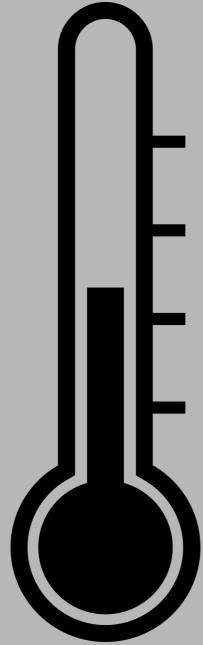
//insert into switchNextState
case AirConditionerState:
    sprintf(msg, "HOT - AC ON \r\n");
    HAL_UART_Transmit(&huart2, (uint8_t*) msg, strlen(msg), HAL_MAX_DELAY);
    NextState = AirConditionerHandler(tempF);
    break;
```

Air Conditioner State (See above)

91 °F

78 °F

... °F

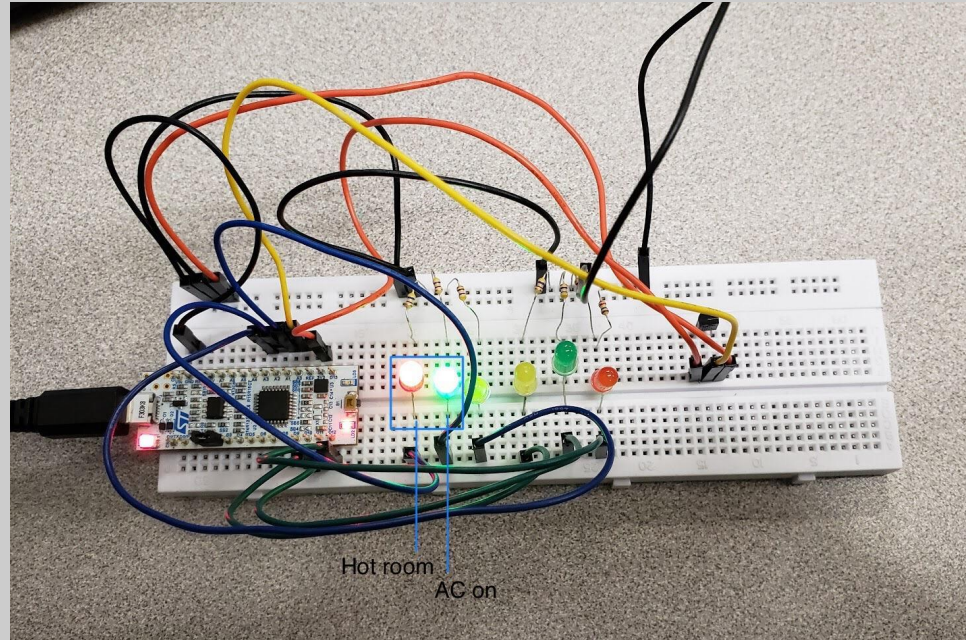



```
DETERMINING...
RawValue: 956
Voltage: 0.770403
Celsius: 27.040291
Fahrenheit: 80.672523

HOT
RawValue: 944
Voltage: 0.760733
Celsius: 26.073259
Fahrenheit: 78.931870

HOT - AC ON
RawValue: 951
Voltage: 0.766374
Celsius: 26.637363
Fahrenheit: 79.947250
```

MobaX display when temperature > 78 °F (See above)



Breadboard when temperature > 78 °F (See above)



HEATER

+ The heater will turn on when the temperature drops below 67 °F

- A green LED indicates that the room is cold
- A red LED indicates that the heater is on

+ The data is transferred to **MobaXterm**

- Temperature state
- Heater on
- Temperature

```
/* If the temperature is determined to be cold, the LEDs resembling the heater and cold temperature
 * will turn on, indicating the environment must be heated to room temperature. The heater LED is red.*/
TempState HeaterHandler(float tempF)
{
    HAL_GPIO_WritePin(GPIOA, HOT_RED_LED_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, ROOM_YELLOW_LED_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, COLD_GREEN_LED_Pin, GPIO_PIN_SET);

    HAL_GPIO_WritePin(GPIOA, AC_GREEN_LED_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, HEATER_RED_LED_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, IDLE_YELLOW_LED_Pin, GPIO_PIN_RESET);

    HAL_Delay(2000);

    return DetermineState;
}

case HeaterState:
    sprintf(msg, "COLD - HEATER ON \r\n");
    HAL_UART_Transmit(&huart2, (uint8_t*) msg, strlen(msg), HAL_MAX_DELAY);
    NextState = HeaterHandler(tempF);
    break;
```

91 °F

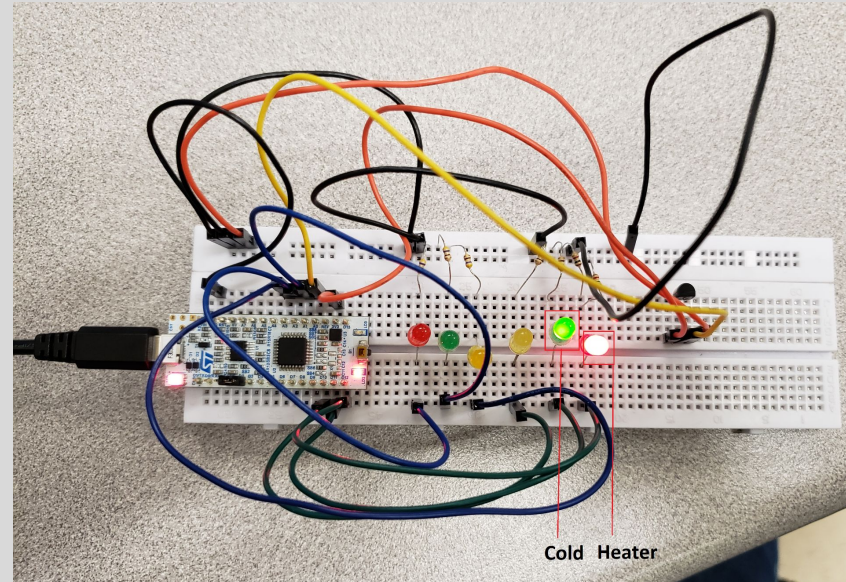
78 °F

... °F

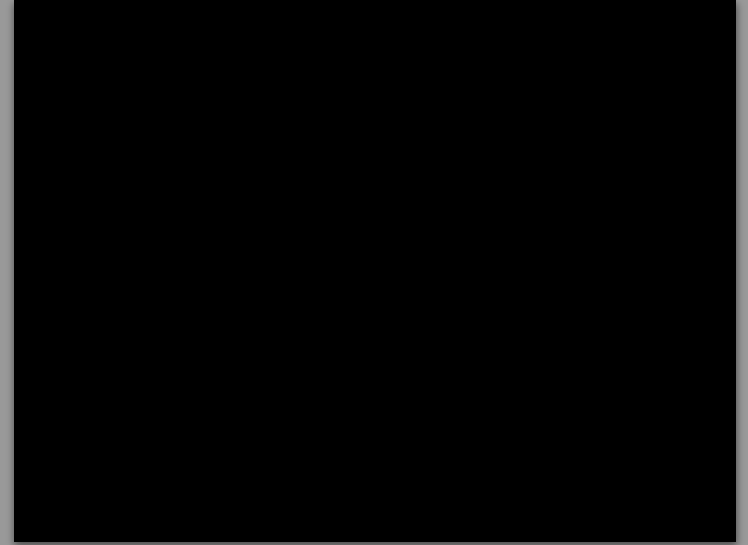
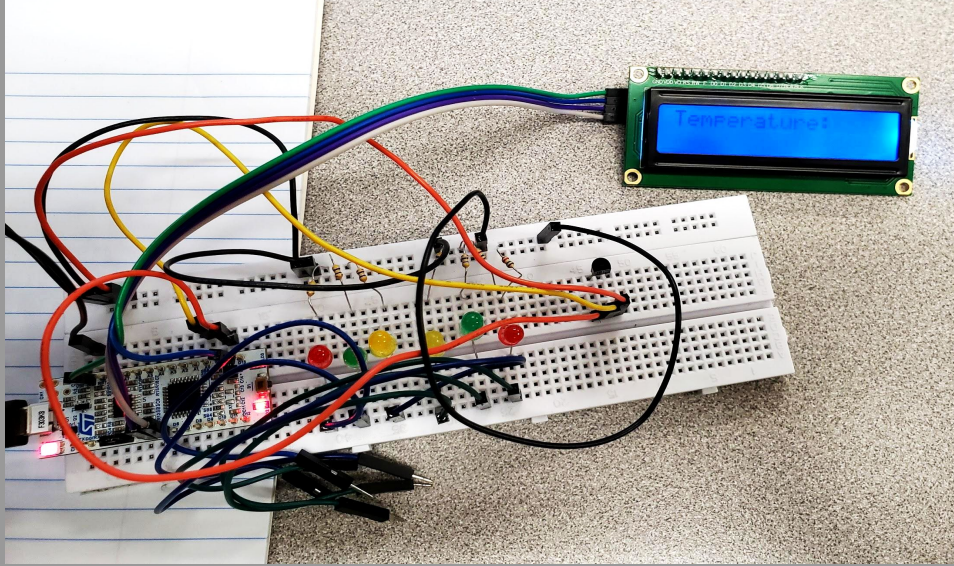


```
DETERMINING...  
RawValue: 853  
Voltage: 0.687399  
Celsius: 18.739927  
Fahrenheit: 65.731873  
  
COLD  
RawValue: 834  
Voltage: 0.672088  
Celsius: 17.208790  
Fahrenheit: 62.975822  
  
COLD - HEATER ON  
RawValue: 855  
Voltage: 0.689011  
Celsius: 18.901098  
Fahrenheit: 66.021980
```

MobaX display when temperature $\leq 67^{\circ}\text{F}$ (See above)



Breadboard when temperature $\leq 67^{\circ}\text{F}$ (See above)



LCD Displaying "Temperature: " (See above)



REFERENCES

[61. STM32CubeIDE LCD 1602 Display. I2C 16x2 with STM32F446RE](#)

[LCD 1602 I2C Display with Nucleo STM32F446RE using STM32CubeIDE](#)

[NUCLEO-F303K8 | Mbed](#)

[Getting Started With STM32 and Nucleo Part 2: How to Use I2C to Read Temperature Sensor TMP102](#)

[TMP35/TMP36/TMP37 \(Rev. H\)](#)

[Temperature Sensor Digital Read-Out](#)

[STM32 Nucleo - Single Mode ADC | LM35 Temperature Sensor | ADC Resolution](#)

-STM32 UART/USART Lab - F303K8 Version

-STM32 ADC Lab - F303K8 Version