Final project report

Mohammad Shughri 110510196

Idea behind the project

This objective of the project is to design a smart water spraying fan that start spraying water according to the conditions around it such as temperature.

The main idea of the project comes from a real-life example that exists in Mecca, Saudi Arabia. The weather in Mecca has usually high average temperature and humidity and with such a huge amount of people every day this application is a great solution to this issue.

Video about it can be found here https://youtu.be/D18PFyJWhNQ.

List of tools and features

Tools	**Usage**
GPIO	GPIO will be used for an LED that will light up when the temperature reaches certain value which will trigger the fan.
ADC	ADC will be used to get the temperature values from the sensors into the microcontroller.
Interrupt	The interrupt will be used to check the sensors regularly.
Communication	Communication and timers will be used to analyze values of the sensors using a terminal and the LCD display.
Timers	The timers are used to indicate when the fan goes into a different mode where the LEDs will start lighting up accordingly.
Sensors	We will use a temperature sensor.
Displays	The display will showcase the current mode of the fan for example, ideal mode, power mode, and UltraPower mode.
Smart Algorithm	We are going to have three different modes. The first one is ideal mode which will only run the fan without spraying water. The other mode will be power mode which will enable the fan to work normally and spray water. The third and last mode is the UltraPower mode which will use the full force of the fan and more water from the water moister. All of modes will change based on the temperature readings from the sensor.

Tools	Price
Fan	150-250 TL
STM32	35 TL
DHT11	12.25 TL
LCD	40 TL
Water mister	200 TL
Total price: 337.25-437.25 TL	

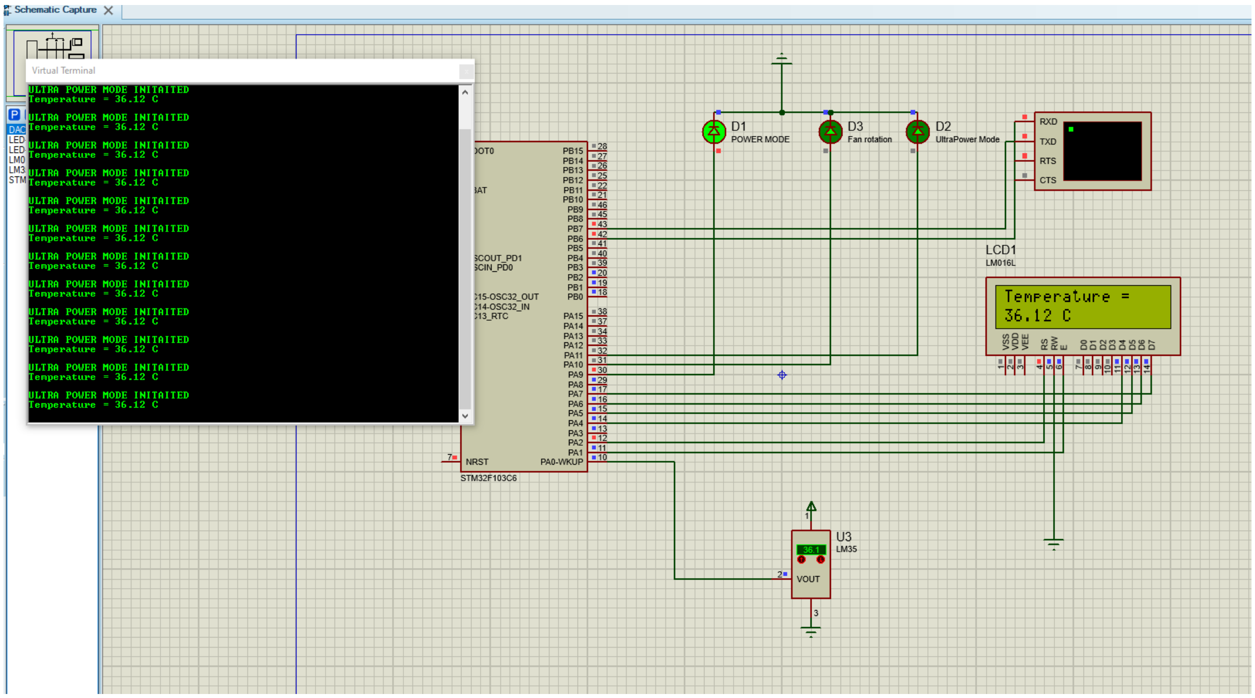
Design steps

The first step of the design was to look for the best sensor for such a project. I have experimented with many different sensors and decided to stick with the simple temperature sensor called LM35. After initializing the base of the project, I tested the LM35 temperature sensor with an ADC and a GPIO and it worked. Then, I started working on USART communication in order to show the data on the virtual terminal. After that, I tried many different approaches for the LCD screen and in the end it worked perfectly. Moreover, I used PWD timer to showcase when the Power mode is triggered using channel 2 in the timer and PA9 on the board. I also used PWD to showcase the constant rotation of the fan where it will keep rotating for 2/3 cycles of 63999 counter period which equals to 42666. Afterwards, I used the Output capture in the timer in channel 4 and PA11 on the board to showcase when the UltraPower mode is triggered using LEDs.

Challenges

I started by looking for a suitable sensor that can both capture humidity as well as temperature and tried up to 5 different sensors. However, most of these sensors did not have a suitable library as well as a beneficial datasheet with STM32. So, I decided to stick with the temperature sensor LM35. Then, I had an issue with showing the ADC value of the sensor on the virtual terminal using USART communication. I fixed this issue by changing the value of USART's baud rate to 9600 Bits/s where the terminal was not showing any values at all. Then, I wanted to show the temperature sensor as a float but the sprintf method only accepted integers. I tried multiple solutions but I finally solved it by applying a trick that we were given in a previous lab and I was able to display the temperature value of the ADC in the terminal as a float. After that, I started working on the LCD screen were I also struggled a lot in and tried multiple different methods and approaches but it didn't work out. On my last try, I opened a new protues project and ran it and it suddenly worked. The next challenge was to figure out the right cycle value for the timers to where it starts blinking when it's above a certain threshold. Trying out different values to cycles yielded interesting results such as the reverse of what was intended as well as the LEDs not going off at all. I ended up using the sweet spot for the cycle value which was 42666.

Screenshot of the design working



NOTE: This project was done individually.