

1. Introdução

Ter um constante monitoramento em tempo real de temperatura de forma precisa em determinadas áreas é algo extremamente necessário em vários ramos diferentes. As áreas monitoradas podem ser fechadas ou abertas, e alguns dos exemplos mais conhecidos são na indústria farmacêutica, para o controle das condições do ambiente em que os remédios mais sensíveis a temperatura estão armazenados, na área alimentícia, com o objetivo de se conseguir preservar alimentos por mais tempo, e na agricultura, com o intuito de se obter as condições necessárias no solo para o plantio.

A depender da situação a qual será empregado, pode ser necessário utilizar várias unidades de um sistema para a coleta dos dados de temperatura, sendo dispostas ao longo de galpões ou estantes, por exemplo. Dessa forma, é relevante que se tenha opções eficazes e simplificadas para implementação de tal sistema, haja vista sua importância e ampla utilização.

Para se obter um sistema como esse, é possível associar apenas um sensor para a coleta dos parâmetros envolvidos, no caso, temperatura, com as informações sendo apresentadas em uma tela de LCD, e o controle sendo realizado por um microcontrolador. Dessa maneira, os dados do sensor são coletados e direcionados para serem exibidos na tela de LCD, com a correta programação do microcontrolador.

Nesse contexto, este trabalho tem o objetivo de implementar um sistema de monitoramento de temperatura, utilizando o microcontrolador STM32F103C6. Sendo realizado via simulação com o auxílio dos softwares Proteus e STM32CubeIDE.

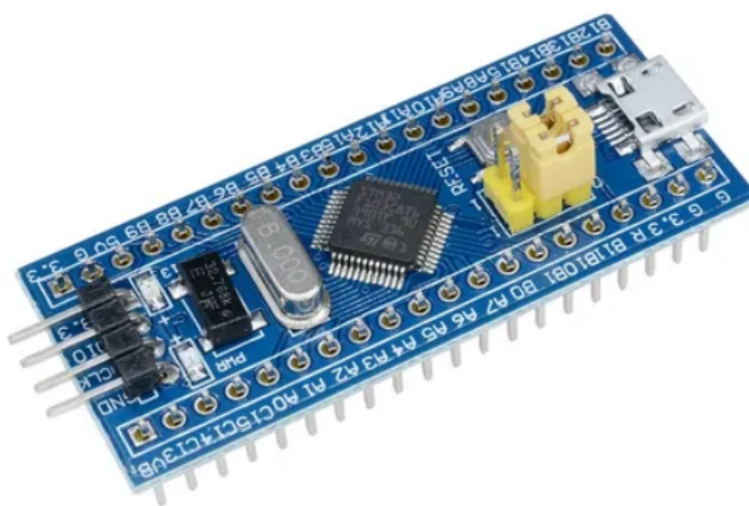
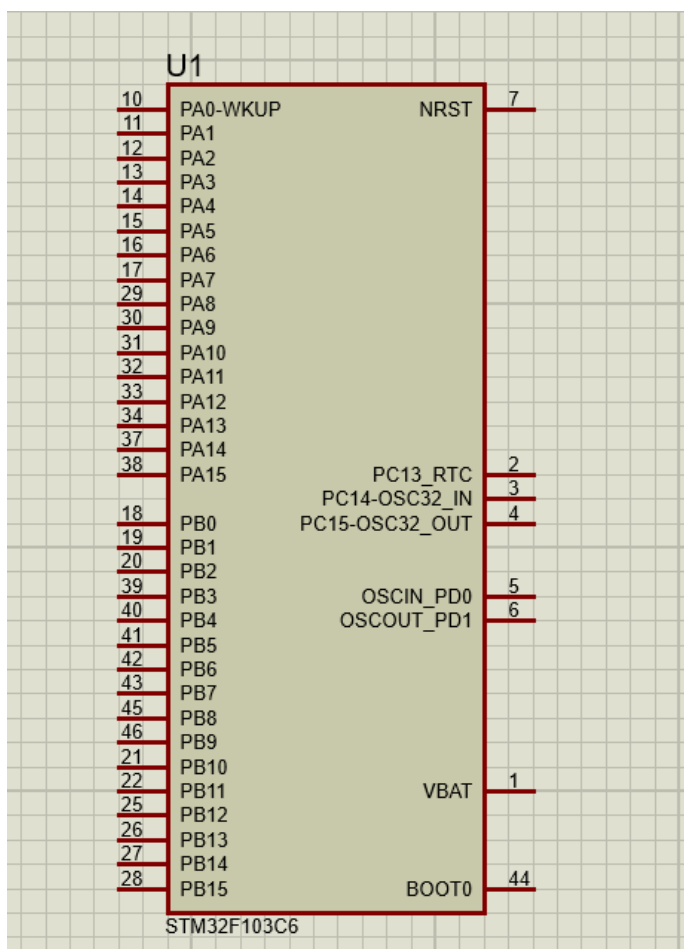
2. Objetivos

1. Detectar com precisão a temperatura de determinada área.
2. Apresentar os dados em um display de LCD para monitoramento.

3. Especificações

Abaixo estão listados os componentes utilizados para este projeto:

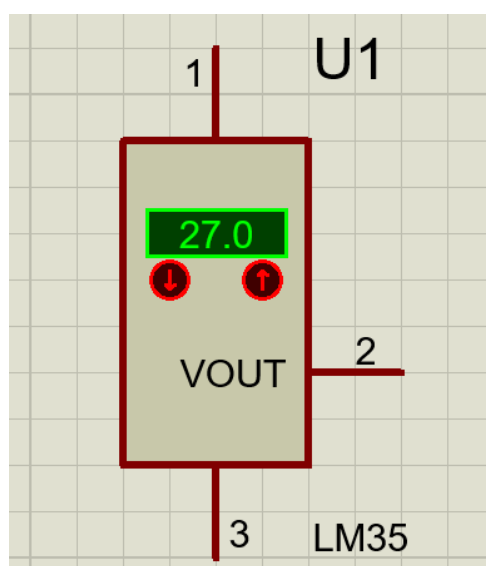
1. Microcontrolador STM32F103C6



O microprocessador STM32F103C6 contém o núcleo RISC de 32 bits ARM® Cortex™ -M3 de alto desempenho com operação em uma frequência de 72 MHz, memórias incorporadas de alta velocidade (memória Flash de até 32 Kbytes e SRAM de até 6 Kbytes), além de várias portas de I / Os e periféricos conectados a dois barramentos APB. Podendo operar com uma fonte de alimentação de 2,0 a 3,6 V.

No	Pin Name	Use
1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	Ground	Connected to the ground of the circuit

2. Sensor de temperatura LM35



LM35 é um dispositivo de temperatura de circuito integrado com uma tensão de saída linearmente proporcional à temperatura centígrada. Como o LM35 não é calibrado em Kelvin, o usuário não precisa subtrair uma grande tensão constante da saída para obter uma escala centígrada conveniente.

Características do LM35

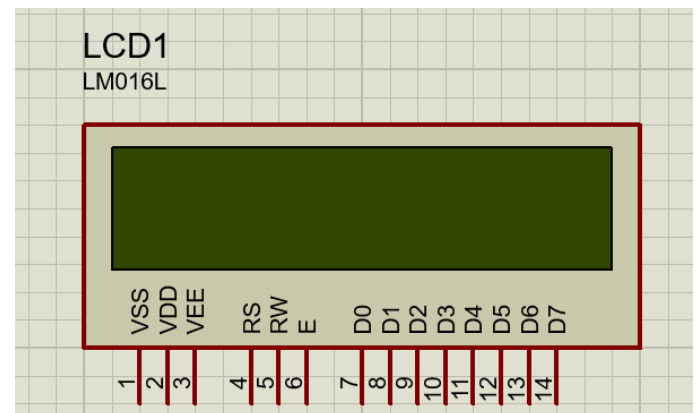
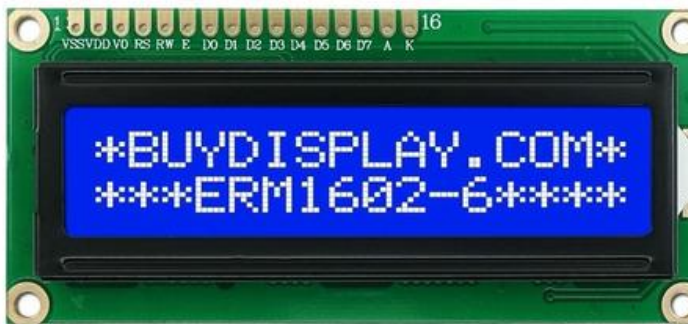
- Calibre diretamente em Celsius
- Precisão garantida de 0,5 ° C (a 25 ° C)

- Avaliado para faixa total de -55°C a 150°C
- Adequado para aplicações remotas
- Opera de 4V a 30V
- Dreno de corrente inferior a $60\mu\text{A}$
- Autoaquecimento baixo, $0,08^{\circ}\text{C}$ em ar parado
- Não linearidade apenas $\pm 1/4^{\circ}\text{C}$ típica
- Saída de baixa impedância, $0,1\ \Omega$ para carga de 1 mA

- Pinagem no sensor LM35:

Vcc é fornecido ao pino 1 e o pino 3 é conectado ao aterramento. O pino 2 será a entrada para o pino ADC do microcontrolador.

3. Display LCD



Display que apresenta uma interface visual simples, exibindo duas linhas de texto com até 16 caracteres. Neste projeto dois displays são utilizados para o sistema.

- Pinagem do LCD LM016L (No proteus os pinos 15 e 16 não aparecem)

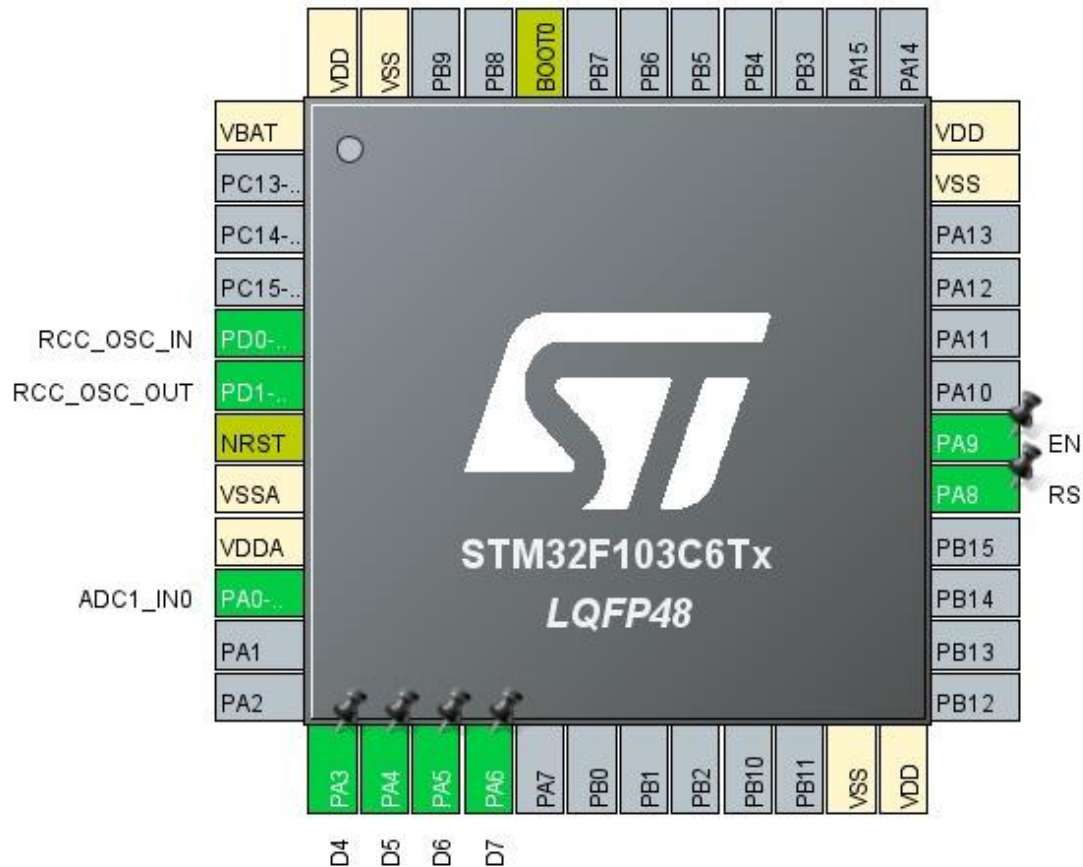
LCD Pin No	LCD Pin Name
1	Ground (Gnd)
2	VCC
3	VEE
4	Register Select (RS)
5	Read/Write (RW)
6	Enable (EN)
7	Data Bit 0 (DB0)
8	Data Bit 1 (DB1)
9	Data Bit 2 (DB2)
10	Data Bit 3 (DB3)
11	Data Bit 4 (DB4)
12	Data Bit 5 (DB5)
13	Data Bit 6 (DB6)
14	Data Bit 7 (DB7)
15	LED Positive
16	LED Negative

4. Esquematisações

4.1 STM32CubeIDE

Essa ferramenta, desenvolvida pela própria STMicroelectronics, possibilita que seja feita uma configuração prévia dos pinos do microcontrolador a fim de gerar automaticamente o código fonte base para a aplicação desejada.

Primeiramente, para a implementação do display LCD, utilizamos as portas PA3 a PA6 do microcontrolador para serem ligadas aos pinos de entrada de dados do display (D4, D5, D6 e D7). A porta PA8 para o pino de Register Select (RS) e PA9 para o pino Enable (EN). Em seguida, PA0 foi configurada como ADC para receber as informações do sensor (pino VOUT). É ativado também o RCC nas portas PD0 e PD1.

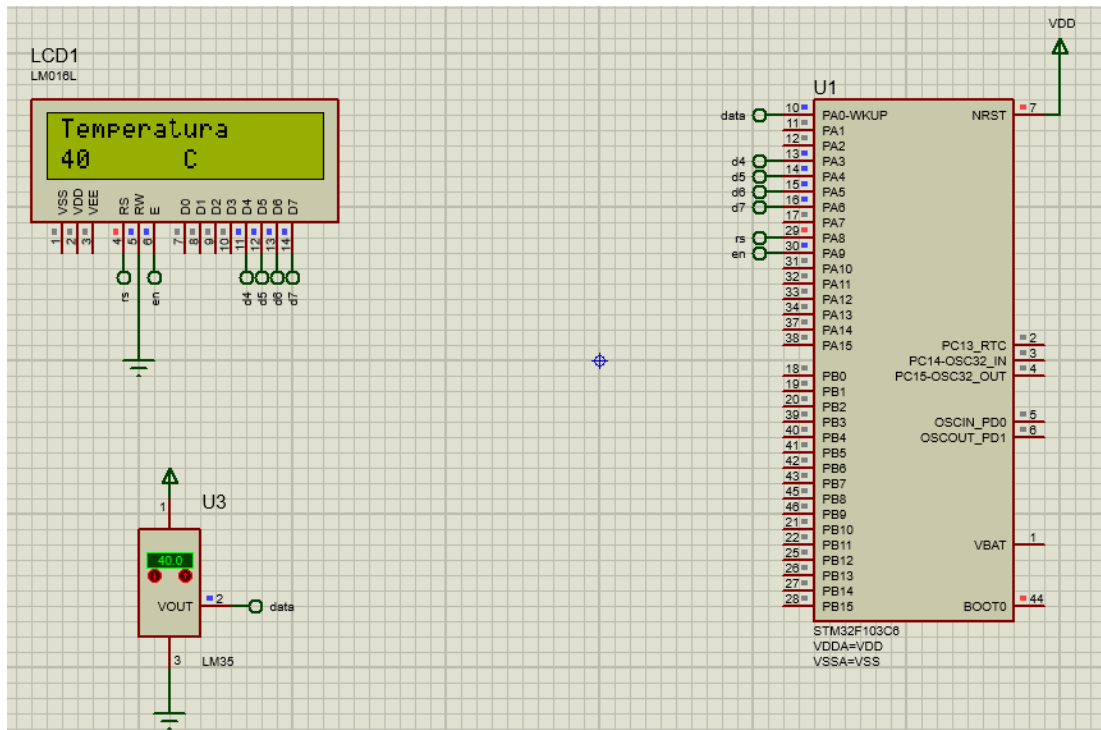


4.1.2 Comunicação do microcontrolador com o LCD

- Conexões no LCD (Todos os pinos foram configurados como OUTPUT no STM32CubeIDE) :
 - 1 → Vss
 - 2 → Vcc
 - 3 → Vee
 - 4 → RS → PA8

4.2 Resumo das Conexões

Os pinos foram devidamente conectados como se segue no esquemático:



5. Algoritmos

Este é o loop principal do algoritmo, aqui iniciamos a porta com a função “HAL_ADC_START”, para que as configurações de porta, pino e ADC possam ser setadas. Após isso é verificado o valor da função “HAL_ADC_PollForConversion”, quando obtido o valor indicado no código é quando é possível obter o valor da porta onde o sensor está conectado. Para obtenção desse dado, usamos função “HAL_ADC_Get_Value”, ela nos retorna o valor que está sendo enviado para a porta no exato momento e, como esse valor é em binário, é necessário realizar a conversão para obtermos o valor em graus celsius, com o fator indicado abaixo. Após isso, armazenamos o valor em uma variável auxiliar e enviamos para o display através das funções da biblioteca Lcd.h, implementada e disponibilizada nos arquivos do projeto.

Note que fazemos a verificação para que caso a temperatura ultrapasse de 43 °C, o que é suficiente para gerar queimaduras, é mostrado um alerta de temperatura elevada.

```

while (1)
{
    /* USER CODE END WHILE */
    HAL_ADC_Start(&hadc1);
    if((HAL_ADC_PollForConversion(&hadc1, 1000000)) == HAL_OK){
        adcval = HAL_ADC_GetValue(&hadc1);
        temp = ((adcval)*3.3);
        aux = (int)temp/26.8;
        if(aux>43){
            adcval = HAL_ADC_GetValue(&hadc1);
            temp = ((adcval)*3.3);
            aux = (int)temp/26.7;
            Lcd_cursor(&lcd, 0, 0);
            Lcd_string(&lcd, "Cuidado!!! ");
            HAL_Delay(200);
            Lcd_cursor(&lcd, 1, 8);
            Lcd_string(&lcd, "C");
            Lcd_cursor(&lcd, 1, 0);
            Lcd_int(&lcd, aux);
        }
        Lcd_cursor(&lcd, 0, 0);
        Lcd_string(&lcd, "Temperatura");
        HAL_Delay(10);
        Lcd_cursor(&lcd, 1, 8);
        Lcd_string(&lcd, "C");
        Lcd_cursor(&lcd, 1, 0);
        Lcd_int(&lcd, aux);
        HAL_Delay(10);
    }
}

```