



**UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO
COLEGIADO DE ENGENHARIA DA COMPUTAÇÃO**

**ANÍSIO PEREIRA BATISTA FILHO
BRENO ARAUJO CAVALCANTE
ELAYNE RUTE LESSA LEMOS
EZAÚ TERTULIANO DA SILVA
JOÃO HENRIQUE LIMA DUARTE
RAFAEL KLEBSON DOS SANTOS MELO**

RELATÓRIO: Sistema para medição de temperatura e luminosidade

**JUAZEIRO
2020**

**UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO
COLEGIADO DE ENGENHARIA DA COMPUTAÇÃO**

**ANÍSIO PEREIRA BATISTA FILHO
BRENO ARAUJO CAVALCANTE
ELAYNE RUTE LESSA LEMOS
EZAÚ TERTULIANO DA SILVA
JOÃO HENRIQUE LIMA DUARTE
RAFAEL KLEBSON DOS SANTOS MELO**

RELATÓRIO: Sistema para medição de temperatura e luminosidade

Relatório apresentado como requisito parcial para obtenção de aprovação na disciplina de Sistemas Microcontrolados, no curso de Engenharia de Computação, na Universidade Federal do Vale do São Francisco – UNIVASF, Campus Juazeiro.
Orientador: Prof. Dr. Jadsonlee da Silva Sá

**JUAZEIRO
2020**

SUMÁRIO

1. INTRODUÇÃO.....	03
2. FUNDAMENTAÇÃO TEÓRICA.....	04
2.1. MICROCONTROLADOR AVR ATMEGA328P.....	04
2.2. SENSOR DE TEMPERATURA - LM35.....	05
2.3. SENSOR DE LUMINOSIDADE - LDR.....	05
3. METODOLOGIA.....	07
4. RESULTADOS E DISCUSSÕES.....	08
4.1. SIMULAÇÃO DO SISTEMA.....	08
4.2. PROGRAMAÇÃO DO MICROCONTROLADOR.....	09
4.3. FASE DE EXECUÇÃO E TESTES.....	12
5. CONCLUSÃO.....	14
REFERÊNCIAS.....	15

1 INTRODUÇÃO

É notável o quanto tem-se popularizado os sistemas de automação e controle, e uma parte importante deles são os sensores e microcontroladores utilizados para obter leituras do ambiente e permitir a partir disso a tomada de decisão ou mesmo outras ações por parte de possíveis atuadores. Dentre esses sistemas, é amplamente documentado o uso de sensores de temperatura, a exemplo de um sistema que mantém a temperatura de uma estufa em temperatura ótima para as plantas, e de luminosidade, a exemplo de leds programados para acender quando escurece.

Desse modo, o presente trabalho visa a simulação de um sistema capaz de ler e reportar visualmente temperatura e luminosidade ambientes. Aplicações assim não seriam possíveis dispositivos como os microcontroladores, de modo que este permitiu toda a integração e tratamento dos dados coletados. Como requisitos, determinou-se:

- Forma de transmissão: serial para um computador via UART;
- Tensão de alimentação: 5V;
- Taxa de transferência serial: 115200 bps, 8 bits sem paridade e um bit stop;
- Formatação da temperatura: uma casa decimal com unidade em “°C”;
- Formatação da luminosidade: uma casa decimal com unidade percentual (%);
- Frequência de leitura: a cada 15 segundos para temperatura e a cada 10 segundos para luminosidade.

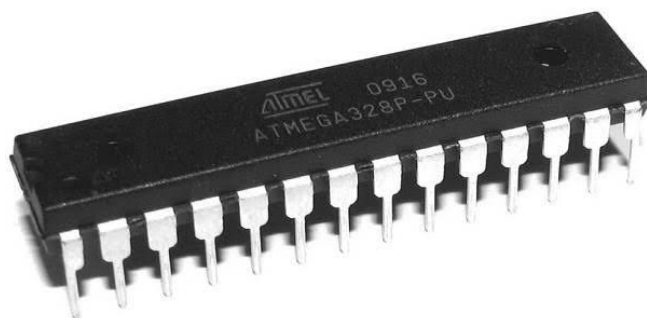
2 FUNDAMENTAÇÃO TEÓRICA

Para o presente trabalho, foram utilizados: o microcontrolador AVR Atmega328p, disponível em uma placa arduino, o sensor de temperatura LM35, o sensor de luminosidade LDR, que serão detalhados a seguir.

2.1 MICROCONTROLADOR AVR ATMEGA328P

O Atmega328p (Figura 1) é um microcontrolador de 8-bit com arquitetura RISC da família AVR, fabricante Atmel. Ele possui 32 registradores de propósito geral de 8 bits, 131 instruções, clock interno de 16MHz (chegando até a 20MHz programáveis), operando entre as tensões de 1,8 V a 5,5 V e as temperaturas de -40 °C a 105 °C. O dispositivo foi desenvolvido com a proposta de otimizar o consumo de energia enquanto entrega uma alta velocidade de processamento.

Figura 1 – o microcontrolador AVR Atmega328p da Atmel



Fonte: <<https://atmega32-avr.com/atmega328-datasheet/>>.

Para a presente aplicação utilizou-se em particular dois módulos disponíveis no microcontrolador, sendo o primeiro a interface USART, ou Transmissor/Receptor Universal Síncrono e Assíncrono. A interface permite que os dados lidos possam ser transmitidos em uma saída, nesse caso, a padrão a serem exibidos na tela de um computador de forma assíncrona. Já o segundo é o conversor analógico digital, capaz de, a partir da leitura por meio dos sensores, converter a tensão recebida em um valor de digital, para este caso, pela seguinte função:

$$Resolução = \frac{\Delta V}{2^N - 1}$$

N = número de bits de resolução do conversor ADC

2.2 SENSOR DE TEMPERATURA - LM35

O LM35 (Figura 2) é um sensor de temperatura de circuito integrado em que a tensão é linearmente proporcional à temperatura em graus Celsius (°C), tido como vantajoso se comparado a outros sensores que retornam valores em Kelvin (K), sendo utilizado em diversas aplicações. Ele não necessita de calibragem externa, tendo uma precisão média de $\pm 0,25$ °C em ambientes fechados e $\pm 0,75$ °C. Opera entre as temperaturas de -55 °C e 150 °C e entre as tensões de referência de 4 V a 30 V.

Figura 2 – sensor de temperatura LM35



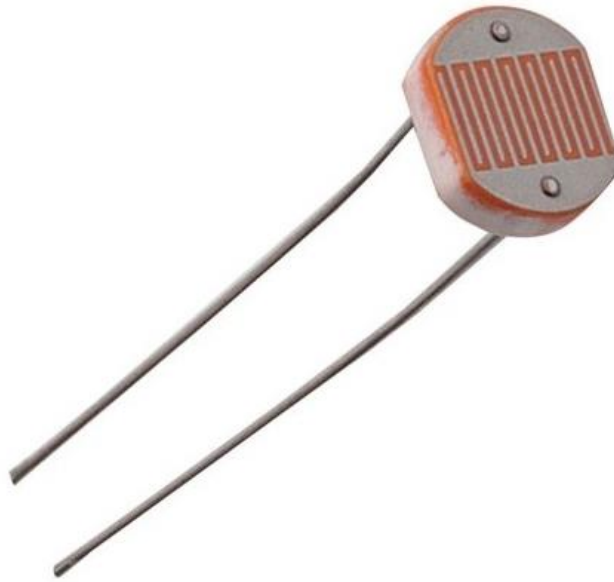
Fonte: <<https://www.indiamart.com/proddetail/lm35-temperature-sensor-18798074573.html>>.

2.3 SENSOR DE LUMINOSIDADE - LDR

Muito considerado para aplicações de detecção luz e escuridão, o LDR (Figura 3) é um sensor de luminosidade que possui resistência variável de acordo com a intensidade da iluminação ambiente. O sensor opera entre as temperaturas de -30 °C e 70 °C, com

resistência no escuro máxima de $1\text{M}\Omega$, e na luz resistência de $10\text{-}20\text{ K}\Omega$, sendo sua tensão máxima de 150 V .

Figura 3 – sensor de luminosidade LDR



Fonte: <<https://www.circuitspedia.com/ldr-sensor-circuit/>>.

3 METODOLOGIA

Foi definido que se faria uso de uma placa arduino uno, plataforma de prototipação eletrônica open source, que possui o microcontrolador AVR Atmega328p, para esta aplicação. Assim, seria montado o sistema de acordo com simulação feita primeiramente no aplicativo tinkercad, de simulação e prototipação de circuitos com microcontroladores, a fim de garantir a montagem correta para não danificar os componentes.

Na programação do microcontrolador, além das funções auxiliares, seriam definidas: funções de configuração do conversor analógico digital e UART, função para impressão no formato específico dos dados de cada sensor, prezando pelo princípio da modularização. Na repetição, deveria ser usada uma variável de controle e manipular o tempo de leitura a partir disso.

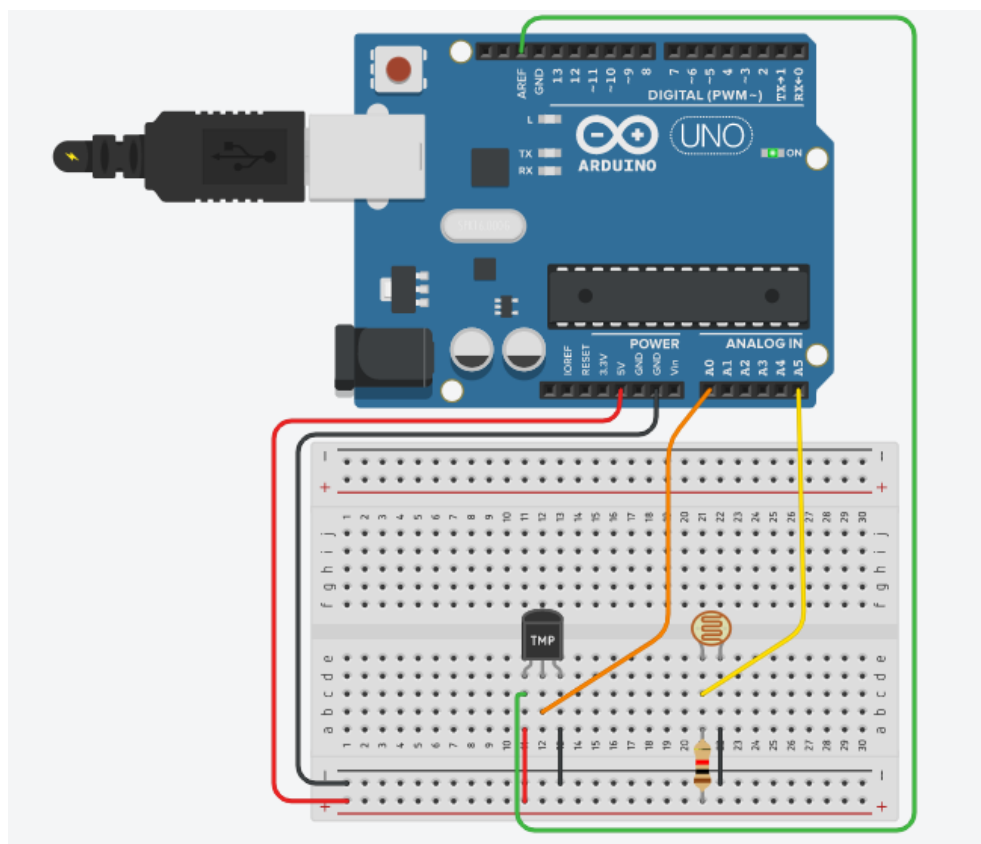
4 RESULTADOS E DISCUSSÕES

Definido como o sistema seria feito, foi possível dar seguimento ao projeto, em que se fez simulação, programação do microcontrolador e execução do sistema, conforme se discorre em seguida.

4.1 SIMULAÇÃO DO SISTEMA

Como determinado anteriormente, o primeiro passo foi o da simulação da montagem do sistema no aplicativo online Tinkercad (Figura 4). O fio laranja representa a conexão do conversor com o terminal do sensor de temperatura, já o fio amarelo representa a conexão com o sensor de luminosidade. O fio verde representa a conexão do circuito com a tensão de referência para o sistema e ajuda a dar estabilidade à leitura analógica.

Figura 4 – Simulação da montagem do sistema no Tinkercad



Fonte: Autores.

Ficou assim determinado também que se atribuiria o pino A0, equivalente ao bit ADC0 do microcontrolador, ao sensor de temperatura, enquanto o sensor de luminosidade seria conectado ao pino A5, equivalente ao bit ADC5.

4.2 PROGRAMAÇÃO DO MICROCONTROLADOR

Apesar de o código fonte desenvolvido ser disponibilizado com comentários no Anexo 1, de nome “sistema_temp_lumi.c”, aqui se comenta as partes que o compõem e a lógica utilizada. Assim, tem-se inicialmente a importação de bibliotecas, definições e declarações de variáveis, conforme pode se verificar no código abaixo.

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#define SENSOR_TEMP 0x00
#define SENSOR_LUMI 0x05
uint8_t datal, datah, check=0;
float tensao;
uint16_t data;
```

As funções de configuração foram definidas para a UART e para modificação do conversor para cada sensor, como se vê abaixo.

```
void adcTempSetup() {
    ADMUX = SENSOR_TEMP;
    ADCSRA = 0x87;
    ADCSRB = 0x00;
    DIDR0 = 0x01;
}
void adcLumiSetup() {
    ADMUX = SENSOR_LUMI;
    ADCSRA = 0x87;
    ADCSRB = 0x00;
    DIDR0 = 0x20;
}
void adcLeitura() {
    ADCSRA = (ADCSRA) | (0x40);
    while((ADCSRA & 0x10) != 0x10) {}
    datal = ADCL;
    datah = ADCH;
    ADCSRA = ADCSRA | 0x10;
    data = (datah << 8 | datal);
}
```

```

void usartSetup() {

    UBRROH = 0x00;
    UBRROL = 0x08;

    UCSROA = 0x40;
    UCSROB = 0x08;
    UCSROC = 0x06;

}

```

Para a transferência via serial foram definidas três funções auxiliares, a primeira envia cada byte de informação (um carácter) pela serial, a segunda permite enviar textos e a terceira transforma os números para que possam ser enviados também, como podem ser vistas a seguir.

```

void transfer(uint8_t byte) {
    while( (UCSR0A & 0x20) != 0x20) {}
    UDR0 = byte;
}
void transferString(char* texto) {
    int i;
    for(i=0; texto[i]!='\0'; i++) {
        transfer(texto[i]);
    }
}
void transferNumb(int valor) {
    char texto[20];
    int i=18;
    do{
        texto[i] = ((char)valor%10)+'0';
        valor/=10;
        i--;
    }while(valor && (i>=2));
    texto[19] = 0;

    transferString(texto+i+1);
}

```

As funções de transferência de temperatura e de luminosidade, bem como a função que se utiliza delas para enviar os dados em um instante são exibidas a seguir.

```

void transferTemp() {

```

```

    adcTempSetup();
    adcLeitura();

    tensao = 4.88*data;

    transferNumb(tensao/10);
    transferString(",");
    transferNumb((int)tensao%10);
    transferString(" °C \n");
}
void transferLumi() {

    adcLumiSetup();
    adcLeitura();

    tensao = (1024-(data*0.9765625));

    transferNumb(tensao/10);
    transferString(",");
    transferNumb((int)tensao%10);
    transferString("% \n");

}
void serialPrint(uint8_t sensor) {
    if(sensor==SENSOR_TEMP) {
        transferTemp();
    }
    else if(sensor==SENSOR_LUMI) {
        transferLumi();
    }
    else{
        transferString("ERROR");
    }
}
}

```

Na main(), é chamada a função de configuração da USART e no laço de repetição define-se os estados possíveis e os atrasos utilizados por meio de uma variável de controle, como pode ser visto abaixo.

```

int main(void) {

    usartSetup();

    while(1) {
        switch(check) {
            case 0:
                _delay_ms(10000);

```

```

        serialPrint(SENSOR_TEMP);
        _delay_ms(5000);
        serialPrint(SENSOR_LUMI);
        check = 1; break;
    case 1:
        _delay_ms(5000);
        serialPrint(SENSOR_TEMP);
        check = 2; break;
    case 2:
        _delay_ms(10000);
        serialPrint(SENSOR_TEMP);
        serialPrint(SENSOR_LUMI);
        check = 0; break;

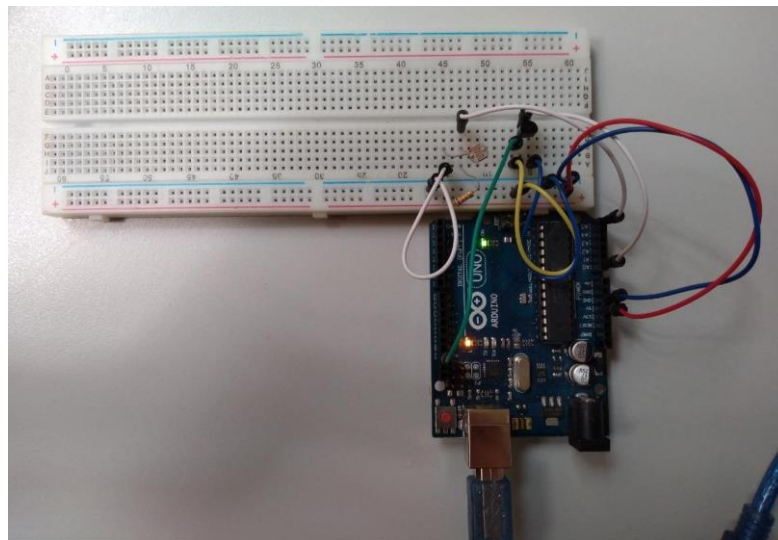
    default:
        serialPrint(10);
}
}
}

```

4.3 FASE DE EXECUÇÃO E TESTES

Após passar pela simulação sem apresentar problemas, foi possível fazer a montagem do sistema fisicamente (Figura 5). De modo que, ao longo de toda a programação do microcontrolador, a cada alteração de código, verificava-se as saídas obtidas.

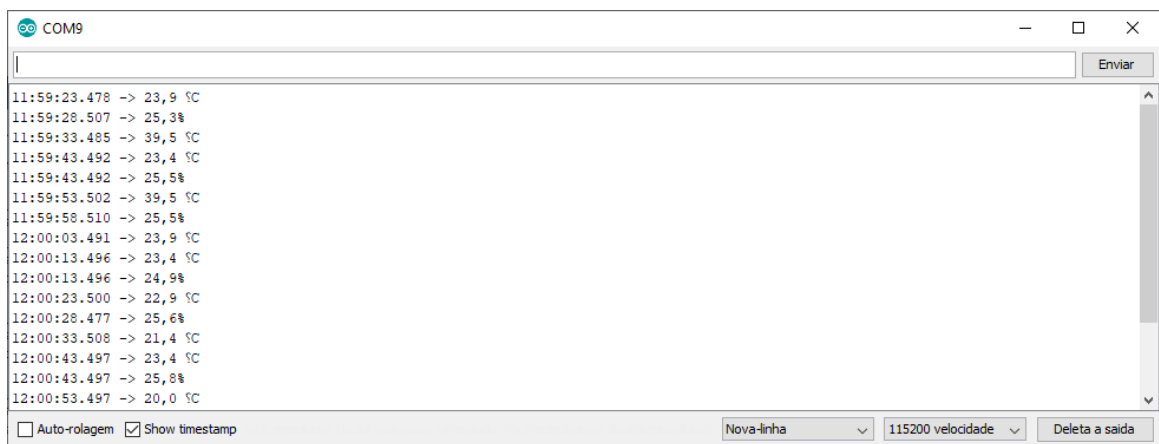
Figura 5 – Montagem do sistema com os sensores LM35 e LDR



Fonte: Autores.

Para facilitar a verificação da duração dos ciclos de cada sensor por meio do timestamp, utilizou-se o monitor serial presente na Arduino IDE, como mostra a Figura 6. Pode-se perceber que a temperatura foi impressa a cada 10 segundos, já a luminosidade, a cada 15 segundos.

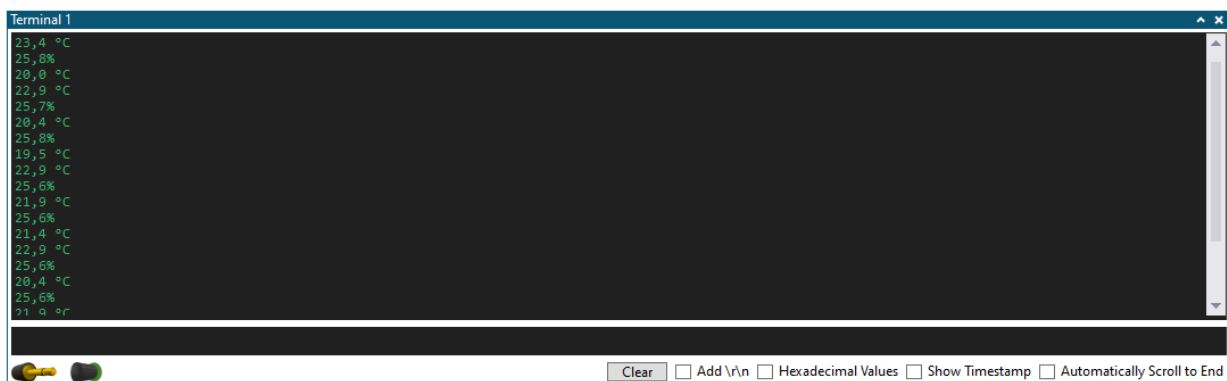
Figura 6 – Saída do sistema em funcionamento na Arduino IDE



Fonte: Autores.

Porém, como também foi possível observar na figura e validar na literatura, a Arduino IDE usa a tabela ASCII como referência, de modo que o carácter “°” não é reconhecido. Assim, foi necessário usar o Atmel studio (Figura 7), para que fossem exibidos os dados corretamente formatados, pois essa possui a tabela ASCII estendida.

Figura 7 – Saída do sistema em funcionamento no Atmel Studio



Fonte: Autores.

5 CONCLUSÃO

Dado os relatos anteriores, conclui-se que foi possível atender aos requisitos estipulados e chegar a um resultado satisfatório. Além disso, é notável como ao longo da execução do presente trabalho foi possível reforçar na consciência dos integrantes da equipe a ampla gama de aplicações para os dispositivos utilizados, em virtude da oportunidade de desenvolver um trabalho prático altamente com vínculo claro às situações reais. Fomentando também a conclusão da importância de processos assim para consolidar o aprendizado na disciplina de Sistemas Microcontrolados.

Ressalta-se como ponto de atenção para sistemas similares que em algumas medições foram obtidas variações mais acentuadas entre uma leitura e outra de temperatura (em até 12 °C de diferença) e de luminosidade (em até de 5%) sem que houvesse variação aparente no ambiente, a isso atribuímos a conexões feitas em protoboard e instabilidade na tensão fornecida. Assim, para trabalhos futuros, pensa-se em estudar e confirmar suposições da variação para possivelmente utilizar de dispositivos que deem maior estabilidade e confiabilidade ao sistema.

REFERÊNCIA

ATMEGA32-AVR.COM, **ATmega328 Datasheet**. Disponível em: <<https://atmega32-avr.com/atmega328-datasheet/>>. Acesso em: 31 jan 2020.

AUTODESK, **Tinkercad | Create 3D digital designs with online CAD**. Disponível em: <<https://www.tinkercad.com>>. Acesso em: 31 jan 2020.

CIRCUITSPEDIA.COM, **Dark Sensor Circuit**. Disponível em: <<https://www.circuitspedia.com/ldr-sensor-circuit/>>. Acesso em: 31 jan 2020.
Disponível em: <<https://www.filipeflop.com/produto/sensor-de-luminosidade-ldr-5mm/>>. Acesso em: 31 jan 2020.

COMPONENTS101.COM, **LDR Datasheet**. Disponível em: <https://components101.com/sites/default/files/component_datasheet/LDR%20Datasheet.pdf>. Acesso em: 31 jan 2020.

INDIAMART.COM, **LM35 Temperature Sensor**. Disponível em: <<https://www.indiamart.com/proddetail/lm35-temperature-sensor-18798074573.html>>. Acesso em: 31 jan 2020.

TEXAS INSTRUMENTS, **LM35 Datasheet**. Disponível em: <<http://www.ti.com/lit/ds/symlink/lm35.pdf>>. Acesso em: 31 jan 2020.