

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO
ENGENHARIA DE COMPUTAÇÃO**

AVALIAÇÃO DA 1ª UNIDADE

Sistemas Digitais

RIJKAARD MELO: 2016018734
TIAGO FELIPE DE SOUZA: 20190153105

**Natal/RN
2021**

RIJKAARD MELO: 2016018734
TIAGO FELIPE DE SOUZA: 20190153105

PROJETO DE SECADOR INDUSTRIAL UTILIZADO PARA SECAGEM DE GRÃOS

Avaliação referente a nota da primeira unidade da disciplina de Sistemas Digitais, semestre 2021.2 do cursode Engenharia de Computação e Automação da Universidade Federal do Rio Grande do Norte.

**Natal/RN
2021**

Lista de figuras

Figura 1: Esquema em diagrama de blocos de um secador industrial utilizado para secagem de grão.

Figura 2: Esquema implementado no *software* Tinkercad da Figura 1.

Resumo

Este relatório tem por objetivo apresentar o esquemático construído e o código desenvolvido em linguagem C dos conceitos aprendidos em sala de aula como entradas e saídas digitais, entradas analógicas, contadores e PWM.

Palavras-chave: digitais, analógicas e PWM.

Sumário

1. Introdução

2. Desenvolvimento

2.1 Código implementado em linguagem C e documentado do secador

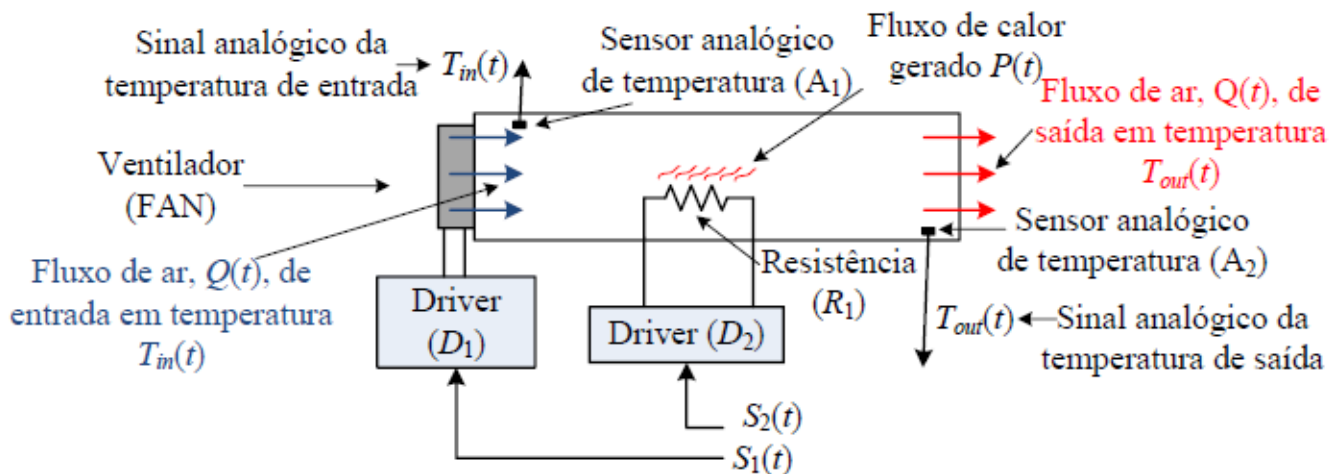
3. Resultados

Referências

1. Introdução

A Figura abaixo, ilustra um esquema em diagrama de blocos de um secador industrial utilizado para secagem de grãos e outros tipos de produtos. O secador já possui circuitos acionadores (drivers) e sensores de temperatura.

Figura 1: Esquema em diagrama de blocos de um secador industrial utilizado para secagem de grão.



Fonte: Material didático para prova da 1ª unidade

2. Desenvolvimento

2.1 Código implementado em linguagem C e documentado do secador

```
//Variavel do sensor de temperatura de entrada
float temp_in;
//Variavel do sensor de temperatura de saída
float temp_out;
//Variavel do sensor de umidade
float umidade;
//Variavel do fluxo de calor
int P = 0;
//Variavel de velocidade do fluxo
int Q = 0;

void setup()
{
    //Habilita as portas 7,6 e 5 como saída e o restante como entrada
    DDRD = 0b1110000;
    //Habilita todas as portas como entrada
    DDRC = 0b00000000;
    //Habilitar a porta 9 como saída
    DDRB = 0b00000010;

    //Fast PWM
    TCCR1A = 0b10000011;
    //Prescaler 256
    TCCR1B = 0b00000100;
}

void loop()
{
    //Verifica se a chave CH1 está ligado
    if(PIND & 0b00000100){

        //Vcc como referencia(01) e ADC0 como entrada
        ADMUX = 0b01000000;
        //Habilita conversor e Inicia a conversão;
        ADCSRA = 0b11000000;

        //Este bit é definido quando uma conversão ADC é concluída e os registros de
        dados são atualizados
        while(!(ADCSRA & 0b00010000));

        //Aproxima o valor registrado para variação de 0 a 200 °c
        temp_in = (ADC*0.59)-11.40;

        //Vcc como referencia(01) e ADC1 como entrada
        ADMUX = 0b01000001;
        //Habilita conversor e Inicia a conversão;
```

```
ADCSRA = 0b11000000;
```

//Este bit é definido quando uma conversão ADC é concluída e os registros de dados são atualizados

```
while(!(ADCSRA & 0b00010000));
```

//Aproxima o valor registrado para variação de 0 a 200 °C

```
temp_out = (ADC*0.59)-11.40;
```

//Verifica se o sistema está funcionando

// O sistema deve funcionar se a temperatura de entrada for maior ou igual a 20

// E saída menor ou igual a 120

```
if(temp_in >= 20 && temp_out <= 120){
```

```
    //Sistema Funcionando
```

//Vcc como referencia(01) e ADC2 como entrada

```
ADMUX = 0b01000010;
```

//Habilita conversor e Inicia a conversão

```
ADCSRA = 0b11000000;
```

//Este bit é definido quando uma conversão ADC é concluída e os registros de dados são atualizados

```
while(!(ADCSRA & 0b00010000));
```

//Aproxima o valor registrado para variação de 0 a 100%

```
umidade = (ADC*0.19608);
```

//Verifica se o valor da umidade é igual a 100

```
if(umidade >= 100.00){
```

```
    P = 100;
```

```
    Q = 25;
```

//No Prescaler

```
TCCR1B = 0b00000001;
```

//Duty 50%

```
OCR1A = 512;
```

//Verifica se o valor da umidade é igual ou maior do que 50

//E umidade menor do que 100

```
}else if(umidade >= 50 && umidade < 100){
```

```
    P = 50;
```

```
    Q = 50;
```

//Prescaler 8

```
TCCR1B = 0b00000010;
```

//Duty 50%

```
OCR1A = 512;
```

//Verifica se o valor da umidade é igual ou maior do que 25

//E umidade menor do que 50

```
}else if(umidade >= 25 && umidade < 50){
```

```
    P = 25;
```

```
    Q = 100;
```

//Prescaler 64


```

        TCCR1B = 0b00000011;
        //Dutty 50%
        OCR1A = 512;

//Verifica se valor da umidade é menor do que 25
}else{
    P = 0;
    Q = 0;
    //Prescaler 256
    TCCR1B = 0b00000100;
    //Dutty 50%
    OCR1A = 512;
}

//Habilita a porta 7 como HIGH
PORTD = 0b10000000;
}else{
    if(temp_in <= 20 && temp_out >= 120){
        //Led laranja e verde
        //Habilita a porta 5 e 6 como HIGH
        PORTD = 0b01100000;
    }else{
        if(temp_in < 20){
            //LED laranja
            //Habilita a porta 5 como HIGH
            PORTD = 0b00100000;
        }else{
            //LED verde
            //Habilita a porta 6 como HIGH
            PORTD = 0b01000000;
        }
    }
}

}else{

    //Habilita todas as portas como LOW
    PORTD = 0b00000000;
    //Desabilitar o conversor
    ADCSRA = 0b00000000;
}
}

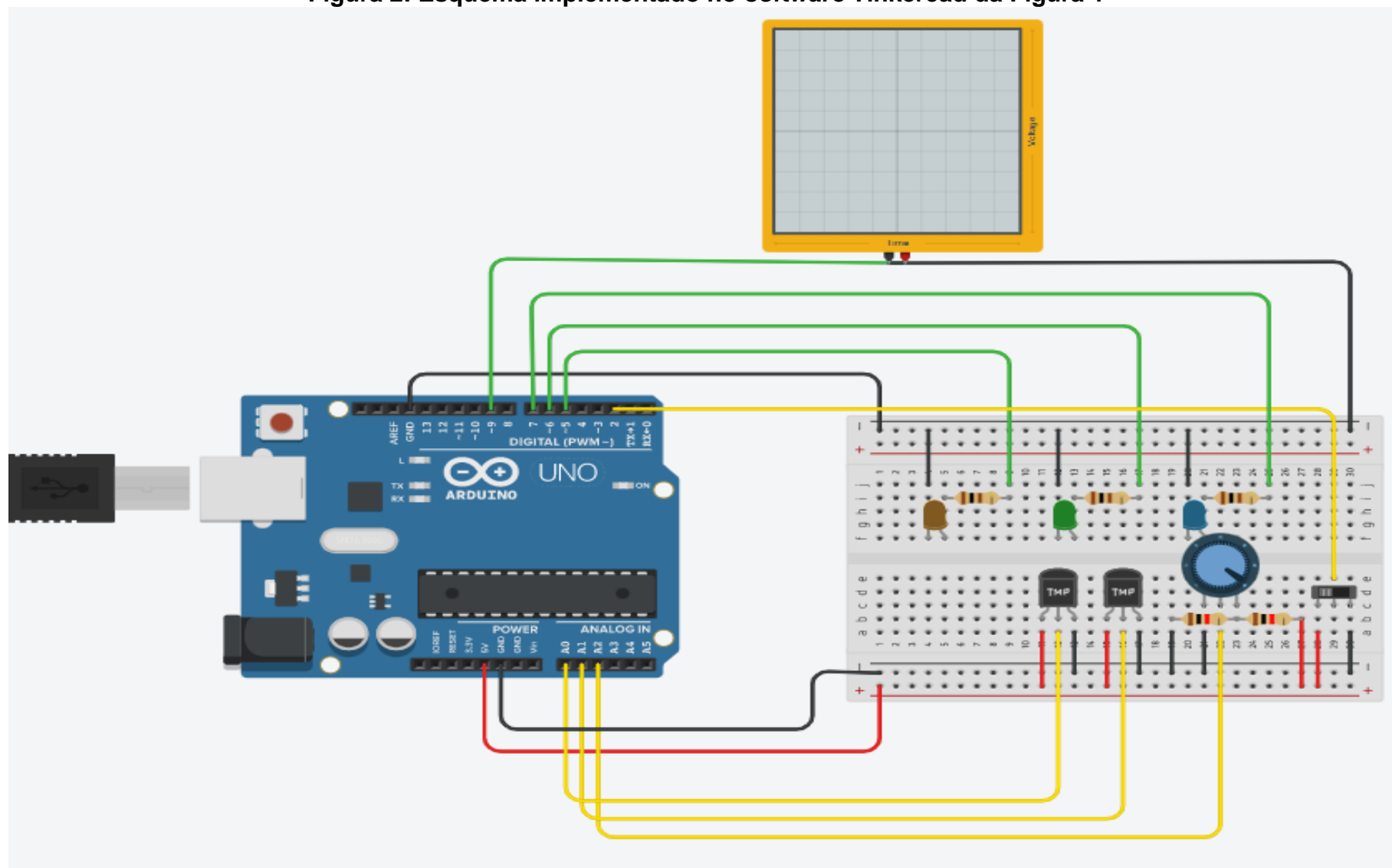
```

***** O link com o código e o esquemático está disponível em:**

<https://www.tinkercad.com/things/gdVbUFBXEi0-super-gaaris/editel?sharecode=d-dOVLiH4yK883oPtYSCzOsFt6lTrn2TzuuJHNF9D0g>

3. Resultados

Figura 2: Esquema implementado no *software* Tinkercad da Figura 1



Foi utilizado dois sensores de temperatura para simular as medidas de temperatura e um potenciômetro para simular o sensor de umidade.

Referências

Aulas síncronas, assíncronas e slides do Professor Marcelo Fernandes constantes no SIGAA.

Material complementar dos monitores, com os vídeos de entradas e saídas digitais, entradas analógicas, contadores e PWM, interrupção e comunicação serial, constantes no SIGAA.

Datasheet do Atmega 328.