

Estufa automatizada

Victor Hugo Bezerra Tavares
FGA
Universidade de Brasília
Gama, Brasil
13/0136492
victorhugo.tavares@hotmail.com

Lucas Rocha F. S. e Barros
FGA
Universidade de Brasília
Gama, Brasil
15/0137478
lucas.oct8@gmail.com

Palavras-chave: *MSP430, Estufa, controle.*

nesses sinais será feito o controle de forma devida da estufa.

I. Introdução

É notória a importância da utilização das plantas pelo ser humano. Um exemplo disso é quando seu uso medicinal pode ser mais eficaz que de um antibiótico para o tratamento de doenças ou condições. Uma planta crescendo em ambiente controlado seria sujeita todo o tempo a temperatura, umidade e luminosidade perfeitos para seu desenvolvimento, e isso implicaria em uma planta mais robusta, sem contar o rápido amadurecimento. A ideia do projeto seria, então, criar uma estufa automatizada para poder controlar os fatores citados acima.

Para fazer o controle da estufa tem-se como ideia usar o microcontrolador msp430 que irá receber dados do sensores de temperatura e umidade e baseado

II. Desenvolvimento

A. Descrição de Hardware

Para a realização deste projeto foi utilizada a seguinte lista de materiais:

| ITEM | QUANTIDADE |
|---------------------------------------|------------|
| MSP430G2553 launchpad | 1 |
| Sensor de Temperatura LM35 | 1 |
| Módulo Sensor Umidade do Solo (LM393) | 1 |

| | |
|----------------------------------|---|
| Jumpers | - |
| Protoboard | 1 |
| Lâmpada incandescente | 1 |
| Fonte de Alimentação(12V , 2,5A) | 1 |
| Estrutura em Acrílico | 1 |
| Tubo Garrote (para bomba d'água) | 1 |
| Transistor TIP120 | 1 |
| Resistores | 4 |
| Bomba D'água(RS385 12V - 2A) | 1 |
| AmpOp(Lm741) | 1 |

O hardware consiste em uma estufa que contém um sensor de umidade, um sensor de temperatura, uma bomba d'água e uma lâmpada incandescente.

O sensor de umidade e o amp-op usado para amplificar a tensão vinda da linha de dados do sensor de temperatura foram alimentados pelo VCC e GND do MSP.

Para amplificar esse sinal foi utilizado do o modelo de amplificador não inversor que possui ganho igual a $(1 + R2/R1)$, onde R2 foram dois resistores de 320 Ω em série e R1 um resistor de 320 Ω , ou seja, o ganho foi de 3. A bomba d'água e o sensor de temperatura foram alimentados com uma fonte de 12 V e 2,5 A. A lâmpada foi ligada em 220 V e a conexão era controlado por um relê que se comportava como chave aberta para VCC.

Quando P1.2 é setado há corrente de base no transistor TIP120 e há a passagem de corrente do emissor para o coletor ativando o motor da bomba. E no caso de P1.3 estar em nível baixo o relê irá se comportar como uma chave fechada ligando a lâmpada.

O sensor de umidade do solo consiste em duas partes, uma sonda que entra em contato com o solo, e um pequeno módulo contendo um chip comparador LM393, que vai ler os dados que vêm do sensor e enviá-los para a MSP430. Como saída, temos um pino D0, que fica em nível 0 ou 1 dependendo da umidade, e um pino de

saída analógica (A0), que possibilita monitorar com maior precisão usando uma porta analógica do microcontrolador. Dependendo do valor de umidade, recebido na MSP430 no pino P1.0, o pino P1.2 irá mandar um sinal de 5V ou 0V para o transistor que funcionará como uma chave que liga/desliga a bomba d'água que irá irrigar a planta. A bomba d'água necessita de alimentação externa, pois a MSP430 não possui corrente necessária para o funcionamento do mesmo, por isso o uso de uma fonte de 12V para a alimentação da bomba.

O sensor de temperatura que primeiramente escolhemos é uma versão à prova de água do sensor DS18B20 que opera entre -55°C até +125°C e com precisão de $\pm 0,5^\circ\text{C}$ se estiver trabalhando dentro da faixa de -10°C até +85°C. Porém devido a dificuldade de comunicação com o sensor, devido aos seus protocolos, foi feito a mudança para o sensor de temperatura LM35. Ele é um sensor de precisão, que apresenta uma saída de tensão linear relativa à temperatura em que ele se encontra no momento em que for alimentado por uma tensão de 4-20Vdc e GND, tendo em sua saída um sinal de 10mV para cada Grau Celsius de temperatura, sendo assim, apresenta uma boa vantagem com relação aos demais sensores de temperatura calibrados em "KELVIN", não necessitando nenhuma subtração de variáveis para que se obtenha uma escala de temperatura em Graus Celsius. O sensor irá monitorar a temperatura ambiente e de acordo com a faixa estabelecida irá controlar a luminosidade incidente na planta com a lâmpada incandescente, que também irá necessitar de alimentação externa devido a limitação de corrente do microcontrolador.

As conexões da parte eletrônica do projeto foram feitas no fritzing e podem ser visualizadas no anexo 1.

B. Descrição de Software

Utilizou-se os pinos P1.0 E P1.1 como entrada e P1.2 e P1.3 como saída. As flags *ADC10ON* e *ADC10IE* setadas no registrador *ADC10CTL0* habilitam a conversão AD e a interrupção do vetor *ADC10*, respectivamente. Não foi a toa que os pinos de entrada foram escolhidos de tal forma, a rotina de conversão múltipla acontece de forma crescente a partir do pino P1.0, sendo os parâmetros configurados para isso o *INCH_1* (habilita o A0 e o A1) e *CONSEQ_1* (habilita a conversão em modo sequencial), ambos no registrador *ADC10CTL1*.

A conversão nesses dois pinos foi habilitada colocando o valor do registrador *ADC10AE0 = BIT0 + BIT1* ; configuramos para *ADC10DTC1 = 0x2* determinando duas conversões para cada bloco (já que faremos duas conversões sequencialmente). A múltipla conversão AD foi habilitada setando a flag correspondente a *MSC*, configurando para dois canais habilitando a flag *ADC10SHT_2*, ambas no registrador *ADC10CTL0*. A linha *while (ADC10CTL1 & BUSY)*; indica que haverá operação de conversão; as flags *ENC + ADC10SC* habilitam a conversão e já dá o início, respectivamente. O valor convertido é armazenado no registrador *ADC10SA* e passado para o vetor inteiro sem

sinal *res* de duas posições. Posteriormente é habilitada a interrupção no vetor ADC10

```
#pragma vector=ADC10_VECTOR  
e sua rotina é  
interrupt void ADC10_ISR(void)  
{  
  __bic_SR_register_on_exit(CPUOFF);  
}
```

liberando os bits especificados na máscara CPUOFF para a próxima interrupção, saindo do modo de baixo consumo.

No pino P1.0 será lido o valor analógico vindo do sensor umidade, (*res[0]*), que responde com um valor alto de tensão (1023 de referência) para um solo seco, e mais baixo para um solo húmido. Fazendo testes e comparações com diversos solos e quantidades de água neles chegamos a um valor de referência de 650 lido.

E no pino P1.1 é lido o valor analógico do sensor de temperatura. Observando seu datasheet e confirmando com testes é notado que o sensor varia 10 mV de tensão para cada °C e é uma relação diretamente proporcional. Como tivemos que amplificar o sinal de dados deste sensor por conta de ser muito inferior ao de referência do MSP essa variação mudou para aproximadamente 118 mV por °C. Saber disso foi importante para poder ter a possibilidade de configurar o sistema de acordo com a necessidade de cada planta. Porém para o nosso caso a temperatura deveria ser de 31

°C. Observando o valor de tensão na variável *res[1]* e confirmando com o cálculo agora pouco citado chegamos num valor de 360.

Então, os valores no vetor *res* controlarão as ações nos pinos P1.2 e P1.3 que controlarão a bomba d'água e a lâmpada respectivamente. Se *res[0]* > 650 significa que o solo está seco, então o BIT2 vai para alto acionando o motor, caso contrário ele não é setado. E se *res[1]* > 360 significa que a temperatura já passou de 31 °C, então o BIT3 é setado e a lâmpada apaga (relé funcionando como chave fechada para GND).

III. Resultados

O composto hardware e software foram avaliados e verificados para se obter a resolução final. Com a estrutura da estufa concluída e todos os dispositivos devidamente posicionados foi possível verificar, após a calibração dos sensores, o funcionamento correto da estufa, respondendo com a precisão esperada para cada um dos sensores.

IV. Conclusão

Em suma, a partir do exposto, pode-se constatar que o projeto concluiu seus requisitos iniciais, com algumas ressalvas durante o decorrer do semestre, porém nada que prejudicou a conclusão da proposta. Foi possível aprofundar bem o

conhecimento adquirido em sala de aula sobre o funcionamento do microcontrolador, em específico referente a lógica dos registradores e a comunicação da placa com outros dispositivos. O conjunto final funcionou de maneira adequada certificando assim o sucesso do projeto.

ANEXOS

Anexo 1

```
#include <msp430g2553.h>
```

```
void main (void)
```

```
{
```

```
    unsigned int res[2] = 0;
```

```
    WDTCTL = WDTPW + WDTHOLD;
```

```
    ADC10CTL1 = INCH_1 +  
    CONSEQ_1;
```

```
    ADC10CTL0 = ADC10SHT_2 + MSC  
+ ADC10ON + ADC10IE;
```

```
    ADC10AE0 = BIT0 + BIT1;
```

```
    ADC10DTC1 = 0x2;
```

```
    P1DIR |= BIT2 + BIT3;
```

```
    P1OUT = 0;
```

```
    for (;;)
    {
```

```
        ADC10CTL0 &= ~ENC;
```

```
        while (ADC10CTL1 & BUSY);
```

```
        ADC10SA = res;
```

```
        ADC10CTL0 |= ENC + ADC10SC;
```

```
        __bis_SR_register(CPUOFF + GIE);
```

```
        if (res[0]>650){
```

```
            P1OUT |=BIT2;
```

```
        }
```

```
        else{
```

```
            P1OUT &= ~BIT2;
```

```
        }
```

```
        if(res[1]>360){
```

```
            P1OUT |= BIT3;
```

```
        }
```

```
        else{
```

```
            P1OUT &= ~BIT3;
```

```
        }
```

```
    }
```

```
}
```

```
#pragma vector=ADC10_VECTOR
```

```
__interrupt void ADC10_ISR(void)
```

```
{
```

```
    __bic_SR_register_on_exit(CPUOFF);
```

```
}
```

Anexo 2:

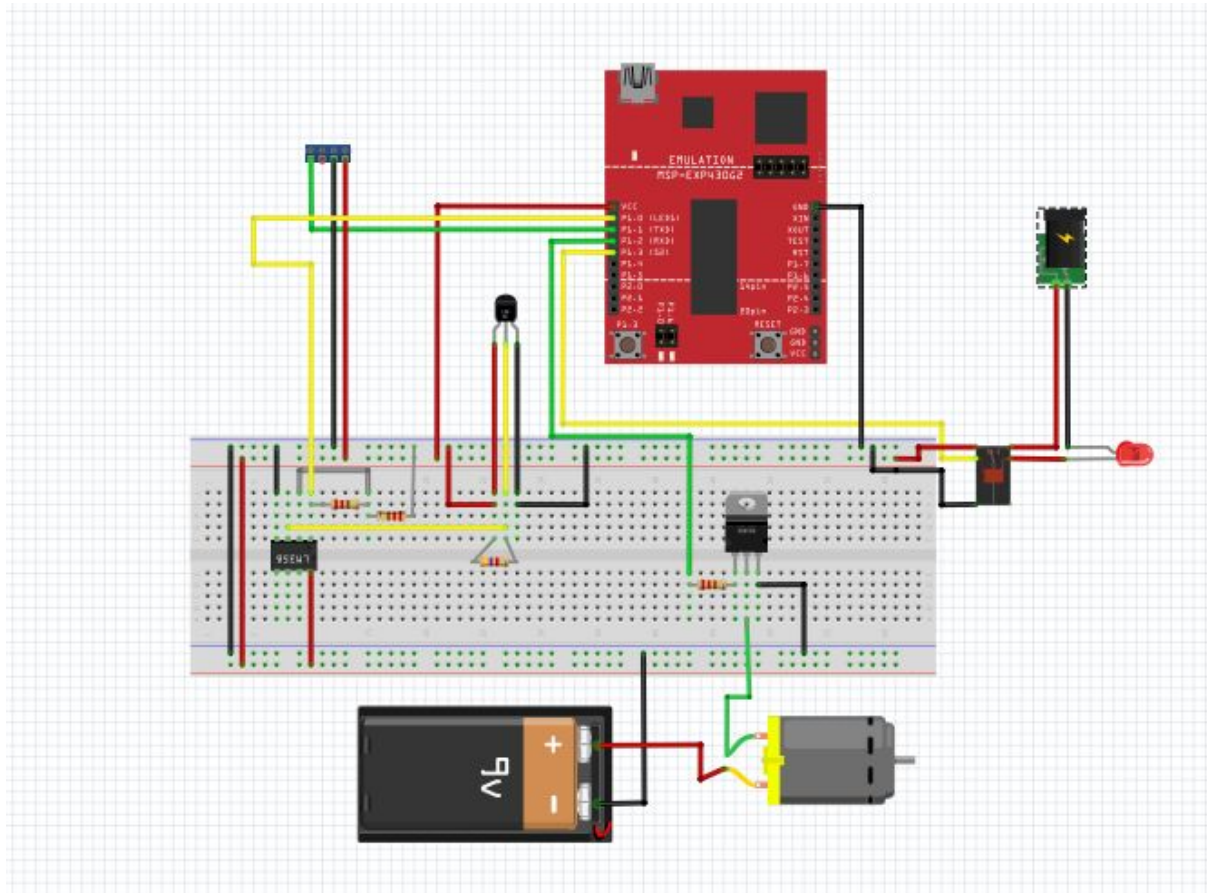


Imagem 1(Esquemático representativo de pinagem projeto final)

Instruction Tables, Estufa
Automatizada. Disponível em:
<<http://www.instructables.com/id/Automated-Greenhouse>>. Acessado em:

V. Referências

Hackster Io, Medidor de temperatura e pressão. Disponível em:
<<https://www.hackster.io/55877/temperature-and-humidity-meter-iot-887cba>>
. Acessado em: 26/09/2017.

Hackster Io, Garden automation and sensor network. Disponível em:<<https://hackaday.io/project/2375-gardenautomationandsensornetwork#menu-description>>. Acessado em: 05/10/2017.

IEEE Xplore Digital Library, Greenhouse monitoring system design based on MSP430 and king view. Disponível em: <<http://ieeexplore.ieee.org/document/7967388/>> Acessado em: 09/10/2017.

Github Repository, The Fisher Innovation Automated Greenhouse is a experimental prototype contained hydroponic greenhouse with automated climate control for use in small spaces. Disponível em: <<https://github.com/fisherinnovation/FI-Automated-Greenhouse>>. Acessado em: 16/10/2017.

Linkedin, Agriculture Projects & Irrigation Projects - IRRIGATION BASED PROJECTS (8051 / AVR / MSP430 / ARM7 / ARM Cortex-M3). Disponível em: <<https://www.linkedin.com/pulse/agriculture-projects-irrigation-based-8051-avr-msp430-prakash/>>. Acessado em: 07/11/2017.

International Journal of Theoretical and Applied Mechanics. ISSN 0973-6085 Volume 12, Number 4 (2017) pp. 821-827. Design And Analysis of Solar Powered Automated Green House. Disponível em: <https://www.ripublication.com/ijtam17/ijtamv12n4_11.pdf>. Acessado em: 07/10/2017.

Springer Link, A Study on Greenhouse Automatic Control System Based on Wireless Sensor Network. Disponível em: <<https://link.springer.com/article/10.10>

[07/s11277-009-9881-2](https://doi.org/10.1007/s11277-009-9881-2)>. Acessado em: 09/10/2017.

Github Repository, Arduino driven green house. Automated to water the plants and open vents at either times, temperatures or humidity levels. Disponível em: <<https://github.com/odd13/greenHouse>>. Acessado em: 10/10/2017.

PJRC, Eletronic Projects Components Available Worldwide - OneWire Library. Disponível em: <https://www.pjrc.com/teensy/td_libs_OneWire.html>. Acessado em: 25/10/2017.

Smallbulb, Welcome To The World Of 1-Wire And MSP430. Disponível em: <<http://www.smallbulb.net/2012/238-1-wire-and-msp430>>. Acessado em: 03/11/2017.

Filipeflop, Monitore sua planta usando Arduino. Disponível em: <<https://www.filipeflop.com/blog/monitore-sua-planta-usando-arduino/>>. Acessado em: 03/11/2017.

Texas Instruments, DHT11 1-Wire Interfacing With MSP430G2553 Using bit banging. Disponível em: <<https://e2e.ti.com/support/microcontrollers/msp430/f/166/t/490177>>. Acessado em: 09/11/2017.