

# Estufa automatizada

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

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

**Palavras-chave:** *MSP430, Hardware, Software.*

## 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 dos sensores de temperatura e umidade e baseado nesses sinais será feito o controle de forma devida da estufa.

## 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
Display (LCD 16x02 - AM/PT)	1

Jumpers	-
Bomba D'água RS385 (12V - 2A)	1
Protoboard	1
Lâmpada incandescente	1
Fonte de Alimentação(12V - 1,5A)	1
Estrutura em Acrílico	1
Tubo Garrote (para bomba d'água)	1
Transistor TIP122	1
Resistor 1K	2

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 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 o pino P1.7 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. O solenóide 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 encontrar 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.

## A. Descrição de Software

Nesse ponto de controle desenvolvemos os códigos que possibilitam a leitura do sensor de temperatura(LM35) e o sensor de umidade(LM393) no software IAR. Foi desenvolvido também o código do sensor de temperatura que iríamos utilizar, o DS18B20, porém não obtivemos êxito na compilação do mesmo, por esse motivo foi feita a troca do sensor. Não conseguimos também elaborar um só código que atendesse nosso sistema, conseguimos compilar apenas os sensores em códigos separados. No próximo ponto de controle, que é a entrega do projeto, pretendemos unificar os códigos e atingir o objetivo inicialmente proposto pelo grupo em conjunto com o professor.

será desenvolvido os códigos para os acionadores (lâmpada incandescente caso a temperatura esteja muito baixa e uma válvula solenóide que irá irrigar a planta caso a umidade do solo esteja muito baixa).

Neste código (anexo 2) é lido o valor analogico de umidade na porta P1.5 e feito a conversão analogico-digital. Se a umidade estiver menor que o valor estabelecido a msp irá mandar um sinal de nível lógico alto pela porta P1.7 para o transistor que funciona como uma chave que irá habilitar a passagem de corrente da fonte para a bomba d'água.

O código do anexo 3 relaciona-se ao sensor de temperatura. Ele é bem parecido com o código anterior pois os dois pegam o valor analogico recebido e fazem a conversao para digital, porém no caso do circuito que tange o sensor de temperatura, o sinal irá acionar um led que posteriormente se substituído por uma lâmpada incandescente.

Anexo 1:

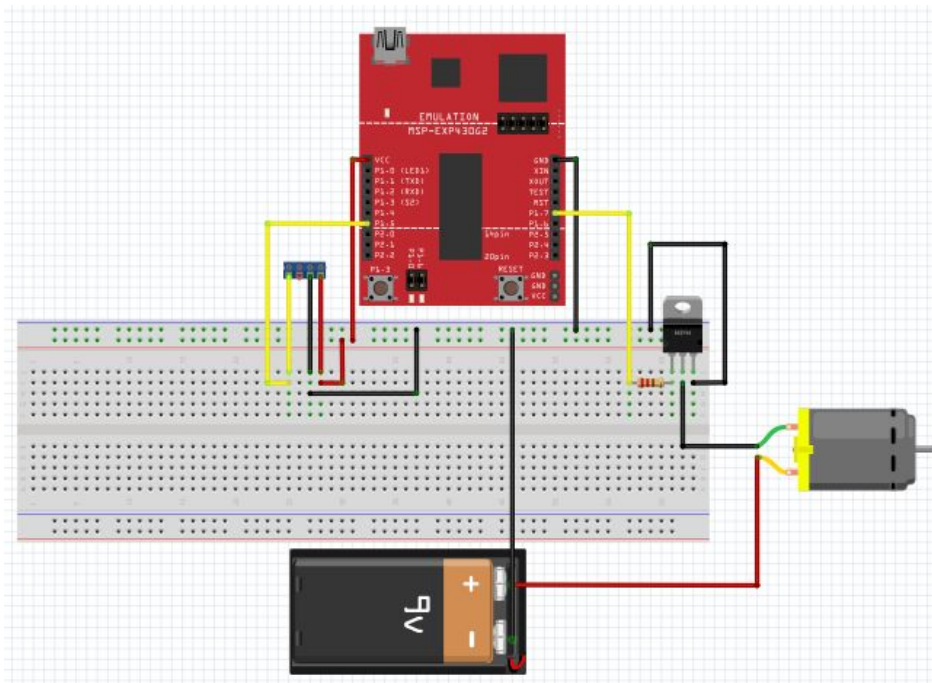


Imagem 1(Esquema de pinagem sensor de umidade)

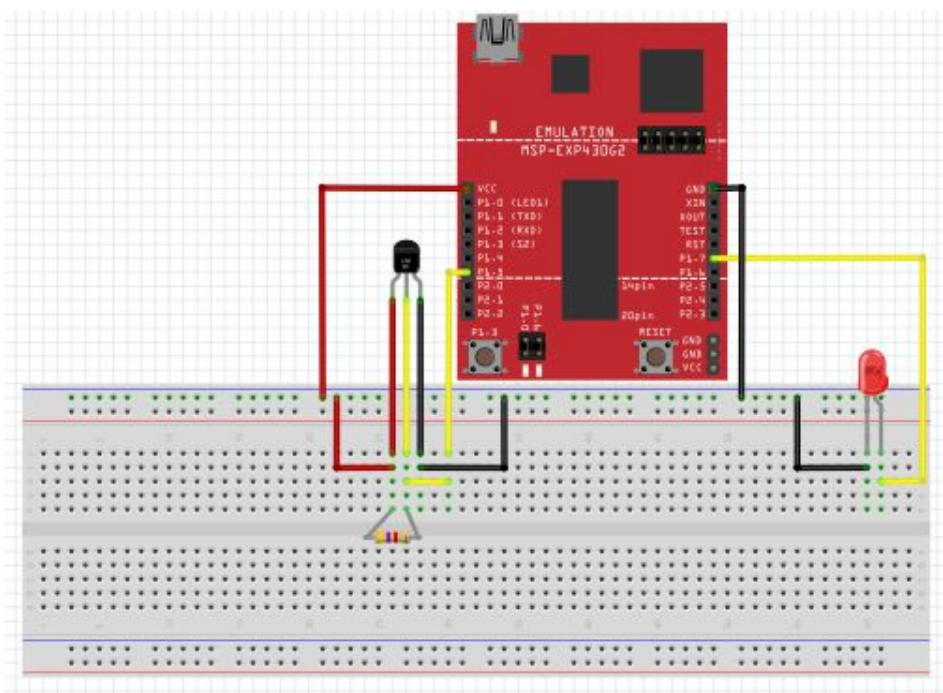


Imagem 2(Esquema de pinagem sensor de temperatura)

## Anexo 2

```
void main(void)
{
    {
        int ValorRecebido;
        WDTCTL = WDTPW +
        WDTHOLD;
        ADC10CTL1 = INCH_5 +
        ADC10DIV_3;
        ADC10CTL0 = SREF_0 +
        ADC10SHT_3 + ADC10ON;
        ADC10AE0 |= BIT5;
        P1SEL |= BIT5;
        BCSCTL1 =
        CALBC1_1MHZ;
        DCOCTL =
        CALDCO_1MHZ;
        P1OUT = 0;
        P1DIR |= BIT7;
        while(1) {
            __delay_cycles(1000);
            ADC10CTL0 |= ENC +
            ADC10SC;
            ValorRecebido =
            ADC10MEM;
            if(ValorRecebido <=
            500){
                P1OUT = BIT7;
            }
            else P1OUT = 0;
        }
    }
}
```

## Anexo 3

```
#include <msp430g2553.h>

void main(void)
{
    {
        unsigned int sample[1];
        WDTCTL = WDTPW +
        WDTHOLD;
        ADC10CTL1 = INCH_5 +
        INCH_7 + CONSEQ_3 + ADC10DIV_3
        +ADC10CT + ADC10SSEL_2;
        ADC10CTL0 = SREF_0 +
        ADC10SHT_1 + ADC10ON;
        ADC10AE0 |= BIT5 + BIT7;
        P1SEL |= BIT5;
        //BCSCTL1 =
        CALBC1_1MHZ;
        //DCOCTL =
        CALDCO_1MHZ;
        P1OUT = 0;
        P1DIR |= BIT7;

        while(1)
        {
            __delay_cycles(1000);
            ADC10CTL0 |= ENC +
            ADC10SC;
            sample[1] = ADC10MEM;

            if((sample[1]*0.35) >=
            1000){
                P1OUT = BIT7;
            }
            else
                P1OUT &= 0;
        }
    }
}
```

```

    }
  }
}

```

<[https://www.pjrc.com/teensy/td\\_libs\\_OneWire.html](https://www.pjrc.com/teensy/td_libs_OneWire.html)>

```

}

```

<<http://www.smallbulb.net/2012/238-1-wire-and-msp430>>

## I. Referências

Hackster Io, Medidor de temperatura e pressão. Disponível em:

<<https://www.hackster.io/55877/temperature-and-humidity-meter-iot-887cba>>

Instruction Tables, Estufa Automatizada. Disponível em:

<<http://www.instructables.com/id/Automated-Greenhouse/>>

<<https://hackaday.io/project/2375-gardenautomationandsensornetwork#menu-description>>

<<http://ieeexplore.ieee.org/document/7967388/>>

<<https://github.com/fisherinnovation/FI-Automated-Greenhouse>>

<<https://www.linkedin.com/pulse/agriculture-projects-irrigation-based-8051-avr-msp430-prakash/>>

<[https://www.ripublication.com/ijtam17/ijtamv12n4\\_11.pdf](https://www.ripublication.com/ijtam17/ijtamv12n4_11.pdf)>

<<https://link.springer.com/article/10.1007/s11277-009-9881-2>>

<<https://github.com/odd13/greenHouse>>

<<https://www.filipeflop.com/blog/monitore-sua-planta-usando-arduino/>>

<<https://e2e.ti.com/support/microcontrollers/msp430/f/166/t/490177>>