

Software para Controle de Temperatura em Estufas

ANTÔNIO CARLOS ALVES SILVA
EDNA MIE KANAZAWA
KARINA DUTRA DE CARVALHO
VANESSA GODOY KINOSHITA
JOAQUIM QUINTEIRO UCHÔA
WILIAN SOARES LACERDA
BRUNO DE OLIVEIRA SCHNEIDER

UFLA – Universidade Federal de Lavras
DCC – Departamento de Ciência da Computação
tony@comp.ufla.br
kanazawa@comp.ufla.br
dutra@comp.ufla.br
vakino@comp.ufla.br
joukim@comp.ufla.br
lacerda@ufla.br
bruno@comp.ufla.br

Resumo: Neste trabalho é apresentado o projeto desenvolvido para controle de temperatura em estufas, denominado CTE, utilizando uma Interface de Aquisição e Controle de Dados [2]. Todo o controle é feito via software, onde pode-se optar por dois tipos de controle: o controle automático, onde é possível definir-se a temperatura mínima e máxima; ou o controle manual, onde o usuário poderá optar por acionar periféricos de aquecimento, resfriamento e desligamento da estufa. Foi desenvolvida também uma interface gráfica, para facilitar a integração do usuário com o sistema.

Palavras Chaves: controle, temperatura, estufa, interface, aquisição de dados .

1 Introdução

O CTE, Controle de Temperaturas em Estufas tem como objetivo a automação do controle de temperaturas em estufas.

A automação torna-se importante nesta área, uma vez que permite um melhor desenvolvimento no cultivo de plantas que não se desenvolveriam por motivos climáticos.

No projeto para controle de temperatura temos dois itens importantes: o software para controle da temperatura, a interface e o hardware representado pela Interface de Aquisição e Controle de Dados [2].

O software de controle recebe os dados, que correspondem a níveis de tensões que são convertidos para temperaturas correspondentes e decide quais dispositivos devem ser acionados se a temperatura estiver fora do limite.

O software de controle permite ter dois tipos de controle: automático e manual.

- Controle Automático: o usuário define um intervalo de temperatura e o software se encarrega de acionar os dispositivos responsáveis em manter a temperatura dentro do limite estabelecido pelo usuário.
- Controle Manual: o usuário define qual dispositivo deseja acionar para aumento ou diminuição da temperatura.

Na Figura 1 apresenta-se uma visão geral do CTE, em diagrama de blocos.

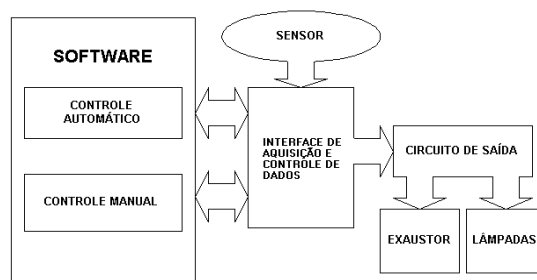


Figura 1: Diagrama de blocos do CTE

O software além de disponibilizar dois tipos de controle envolve também a interface gráfica, para facilitar a integração do usuário com o sistema.

A Interface de Aquisição e Controle de Dados tornou-se um item importante para o desenvolvimento do sistema, pois é possível converter um sinal analógico para digital, além de outros fatores como a amplificação do sinal proveniente do sensor para obtendo-se um sinal com maior precisão.

O sensor de temperatura (NTC – Negative Temperature Coeficient) conectado à Interface de Aquisição de Dados responsabiliza-se por colher dados da estufa e enviá-los ao software de controle da temperatura, para verificar se esta temperatura está dentro do limite fornecido pelo usuário.

Todo o programa de controle foi desenvolvido utilizando a linguagem C++ [1, 4, 6], o compilador mingw32 [3] e para desenvolvimento da interface gráfica utilizou-se a biblioteca wxWindows [5].

A linguagem C++ é freqüentemente chamada de linguagem de médio nível, isto permitiu combinar linguagens de alto nível com a funcionalidade da linguagem assembly. A importância da linguagem C no projeto, permitindo a manipulação de bits e endereços, freqüentemente utilizada, pois temos ativação de periféricos via portas endereçáveis [4].

O wxWindows é uma estrutura produzida para GUI (Graphical User Interface), por ser multiplataforma vários recursos podem ser associados a interface gráfica desenvolvida.

Para verificação da funcionalidade do sistema desenvolveu-se um protótipo. Na Figura 2 apresenta-se a integração do sistema.

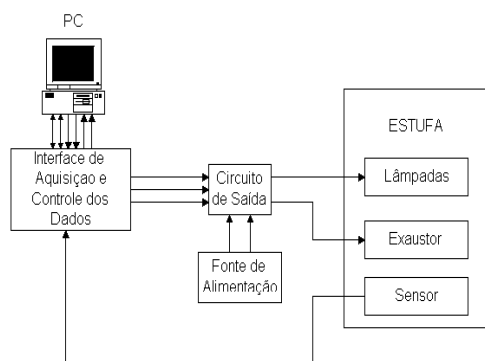


Figura 2: Integração do sistema

2 Hardware

O hardware constitui-se de quatro partes:

- Interface de Aquisição e Controle de Dados [2];

- Sensor de temperatura (NTC);
- Circuito de saída;
- Periféricos (lâmpadas e exaustor).

O sensor presente na estufa constitui-se de uma resistência que varia com a temperatura (NTC). Como a variação de tensão destes sensores é pequena, realiza-se o condicionamento deste sinal possibilitando sua interpretação pelo conversor analógico para digital. O conversor ADC 0804 responsabiliza-se pela conversão do sinal analógico produzido pelo sensor em correspondente sinal digital, tornando possível o tratamento e a interpretação do sinal via software. Toda a parte de aquisição e conversão faz parte da Interface de Aquisição e Controle de Dados.

O circuito de saída também acoplado à Interface de Aquisição e Controle de Dados responsabiliza-se no acionamento de três dispositivos acoplados a ele para controle da temperatura, ou seja tem-se duas lâmpadas para aumento da temperatura e um exaustor para diminuir a temperatura se esta estiver acima do limite.

Na Figura 3 apresenta-se o circuito de saída, que faz parte do sistema CTE.

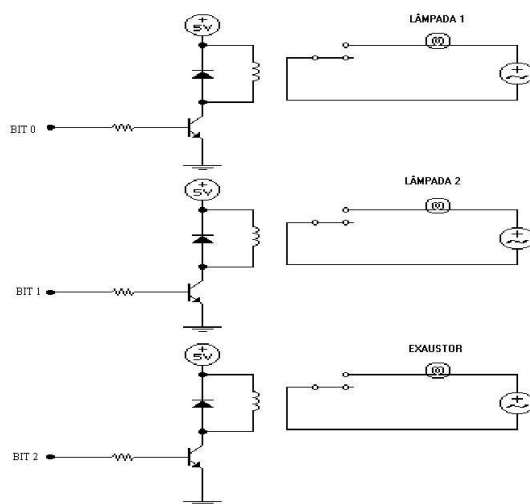


Figura 3: Circuito de Saída

Observa-se pela Figura 3 a existência de duas lâmpadas, que atuam em conjunto e têm o objetivo de aquecer a estufa, tem-se também um exaustor utilizado para resfriamento.

A comunicação software-hardware é feita pelo barramento ISA do PC, onde a Interface de Aquisição e Controle de Dados é acoplada. O endereço para comunicação inicia-se a partir de 300H, mas nem todos os recursos da Interface são utilizados.

3 Software

Desenvolveu-se o software utilizando-se os recursos da orientação a objetos, ou seja, o programa de controle constitui-se de cinco classes: CTE, Dados, Saída, Help, Controle Automático e Controle Manual, sendo cada uma descrita nos itens seguintes.

3.1 Classe CTE

A classe CTE inicia o sistema chamando outras classes ou finalizando o sistema. Esta classe constitui-se de uma tela principal com as opções para os tipos de controle e ajuda ao usuário.

Após a escolha de uma das opções, a classe CTE determina qual objeto utilizar. Ao fim de cada processo (controle automático, controle manual ou ajuda ao usuário), o usuário pode retornar à tela principal, onde também há a opção de finalizar o sistema.

Tanto na inicialização do sistema quanto na finalização os periféricos serão desligados via software.

A interface gráfica simplifica e facilita a comunicação com o usuário.

Na Figura 4 apresenta-se a tela principal do sistema.

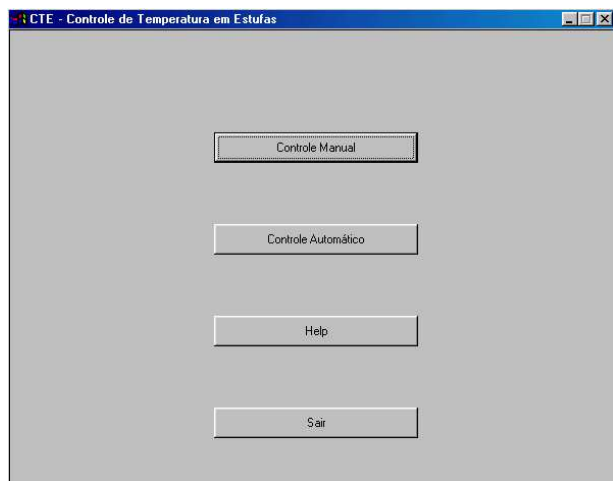


Figura 4: Tela principal do sistema

3.2 Classe Dados

Na classe Dados determina-se a temperatura da estufa. Para coleta da temperatura na estufa são utilizados três endereços para comunicação software-hardware descritos a seguir:

- 307H – para ativar a conversão analógico/digital;
- 306H – para leitura do sinal digital proveniente do conversor;

- 304H – para controle dos periféricos, do ganho amplificador e controle do sample/hold.

Na Figura 5 apresenta-se o fluxograma da classe Dados.

Observe pelo fluxograma que o ganho do amplificador é ajustado até que se tenha uma leitura precisa da temperatura.

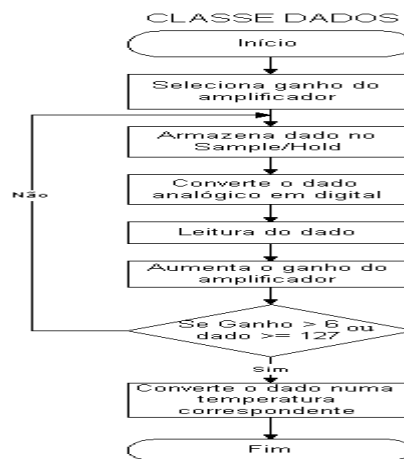


Figura 5: Fluxograma da Classe Dados

Para ajustar o ganho do amplificador e o sample/hold da Interface de Aquisição e Controle de Dados os bits D4 a D7 da porta 304H são manipulados a cada intervalo de leitura da temperatura.

A seguir é apresentada a interface da classe Dados.

```
#ifndef DADOS_H
#define DADOS_H
class Dados{

public:
    Dados();
    int gettemp( );
    void registrador(int ganho,int samplehold, int bitsaida);

private:
    inline int inportb(int port);
};
#endif
```

A função `gettemp ()` segue o fluxograma da Figura 5. A função `registrador` tem o objetivo de ajustar a palavra de controle da porta 304H.

Na Tabela 1 são apresentados os ganhos que o amplificador programável pode assumir na manipulação dos bits de D4 a D6. O bit D7 é utilizado para ativação do sample/hold.

Tabela 1: Ganhos do amplificador programável

D6	D5	D4	Ganho do amplificador
0	0	0	2
0	0	1	4
0	1	0	6
0	1	1	8
1	0	0	10
1	0	1	12
1	1	0	14

Como a palavra de controle da porta 304H utilizada para diferentes funções e também utilizada em classes diferentes, a equação 1 ajusta os bits D4 a D7, que são utilizados na classe Dados e mantêm o estado dos bits de D0 a D3 (bitsaida), que são utilizados pela classe Saida para ativação dos periféricos.

```
palavra = (samplehold * 128 + ganho * 8 + bitsaida); (1)
```

Após a leitura do dado convertido, feita pela função inportb desenvolvida em assembly, se o dado estiver adequado, ou seja, bem amplificado o dado é convertido para uma temperatura correspondente.

O valor 0,0196 da equação 2 corresponde a variação de um bit do sinal digital. A variável *resultado* corresponde ao sinal analógico proveniente do NTC. A equação 3 representa a curva característica do NTC, e com ela é possível determinar a temperatura correspondente ao sinal analógico.

```
resultado = entanalogica * 0.0196/ganho[i-1]; (2)
```

```
temperatura = (int)(resultado * (-30) + 99); (3)
```

3.3 Classe Saída

Através de um objeto dessa classe, o software controla os periféricos, que estão conectados ao circuito de saída.

Abaixo temos descrito os bits utilizados para controle dos periféricos associados a eles.

- D0 para acionamento da primeira lâmpada
- D1 para acionamento da segunda lâmpada
- D2 para acionamento do exaustor

A seguir é apresentado o protótipo da classe Saída.

```
#ifndef SAIDA_H
#define SAIDA_H

class Saida{
```

```
public:
    Saida();
    void ligarVentilador();
    void ligarLampadas();
    void desligar();
    inline void outportb(unsigned short int port,unsigned char val);

private:

};
#endif
```

Os D0 e D1 são manipulados conjuntamente.

Para ativar um periférico é utilizado a função outportb, desenvolvida em assembly. Com esta função é possível enviar uma palavra de controle para uma porta endereçável. A seguir é apresentado um exemplo utilizando a função outportb.

```
outportb(0x304, 0x04);
```

No exemplo acima o endereço da porta é 304H e a palavra de controle é 4, onde no caso estaremos manipulando o bit D2, que é responsável em acionar o exaustor.

3.4 Classe Help

Para ajudar o usuário a entender o funcionamento do software, há uma explicação geral de como funciona cada tipo de controle, além de explicações de como manusear o sistema. Há também uma parte ilustrativa onde temos diagramas em blocos do hardware e do software. Na Figura 6 é mostrado a tela principal da classe Help.

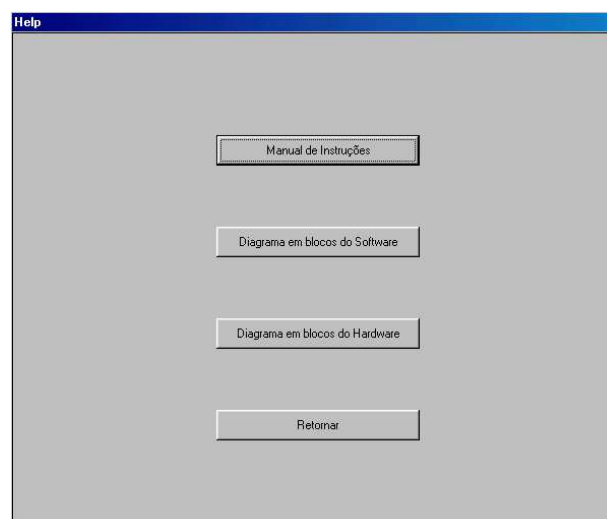


Figura 6: Tela principal da classe Help

3.5 Classe Controle Automático

Esta classe tem por objetivo acionar o circuito de saída automaticamente de acordo com a temperatura

retornada pela classe Dados. No início do processo, o usuário determina um intervalo de tempo para a leitura da temperatura na estufa, em segundos e a temperatura mínima e máxima desejada. O programa só aceita valores entre 0°C e 70°C, sendo a mínima menor que a máxima. Caso uma dessas exigências não seja seguida é emitido um aviso para que as correções sejam realizadas pelo usuário.

Após o acionamento do sistema, toma-se uma decisão a cada leitura. É possível visualizar a temperatura atual do sistema e o estado dos periféricos na estufa. O controle possui dois graus de liberdade, ou seja, aciona-se as lâmpadas a partir da temperatura mínima mais dois graus e aciona-se o exaustor a partir da temperatura máxima menos dois graus.

A seguir é apresentada a função que realiza o controle da temperatura.

```
comparacao(int temp,int maiorTempNormal, int
menorTempNormal){
//temp corresponde a temperatura atual
//maiorTempNormal corresponde a temperatura
//definida pelo usuário
//menorTempNormal corresponde a temperatura
//mínima definida pelo usuário
//O controle possui dois graus de liberdade

//Verifica se temperatura na estufa está
//maior que o desejado pelo usuário
//Se estiver, liga-se o ventilador
if (temp > (maiorTempNormal - 2)) return 1;

//Verifica se temperatura na estufa está
//menor que o desejado pelo usuário
//e estiver, ligam-se as lâmpadas
else if (temp<(menorTempNormal+2)) return 2;

//Caso contrário, a temperatura está entre o
//intervalo desejado e os periféricos não
//são ativados
else return 0;
}
```

Para modificação do intervalo de leitura e temperaturas máximas e mínimas o usuário deve primeiro desligar o sistema.

Na Figura 7 é mostrado a tela do controle automático. O termômetro é um recurso ilustrativo, que auxilia o usuário na leitura e visualização da variação da temperatura. Este recurso é apresentado também no controle manual.

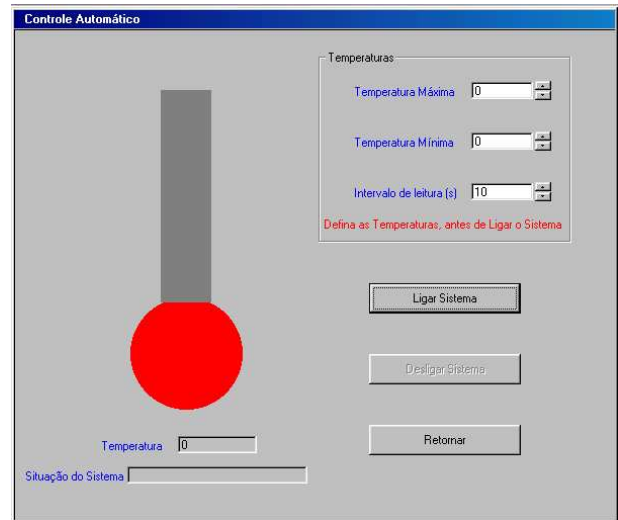


Figura 7: Tela do Controle Automático

3.6 Classe Controle Manual

Neste controle o usuário tem a opção de selecionar o periférico que deseja acionar. No início do processo, o usuário deve determinar um intervalo de tempo para a leitura da temperatura na estufa, em segundos. O sistema atualiza a temperatura a cada intervalo de tempo determinado pelo usuário. O usuário pode selecionar qual periférico deseja acionar e o software, através de um objeto da classe Saída, envia uma palavra através do porta 304H, responsável pelo acionamento dos periféricos de aquecimento, resfriamento e desligamento. Através desse menu, o usuário pode sair do controle manual e voltar para a tela principal, onde novas opções podem ser selecionadas.

Na Figura 8 é apresentada a tela do Controle Manual.

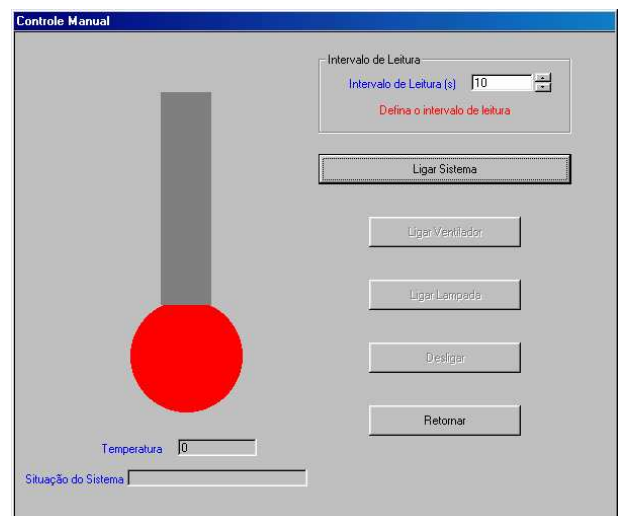


Figura 8: Tela do Controle Manual

4 Conclusão

Um dos principais motivos para desenvolvimento do CTE foi a verificação da integração software/hardware, e a geração de um software de controle eficiente para o sistema integrado.

Num projeto mais amplo pode-se acoplar outros tipos de sensores, que são necessários para o cultivo de plantas, que sofrem com a variações não só de temperatura, mas também de umidade e outros fatores ambientais.

Novos recursos gráficos, como o gráfico da variação da temperatura, poderiam ser desenvolvidos e associados ao sistema.

A geração de relatórios também seria uma implementação futura.

5 Referências

- [1] DEITEL, H. M. & DEITEL, P. J., *C++ How to program*. Upper Sandle Rive, New Jersey: 1998. 2ª ed. 1130 p.
- [2] KANAZAWA, M. Edna & LACERDA, S. Wilian *Desenvolvimento de uma Interface de Aquisição e Controle de Dados*. Lavras: InfoComp,V.1, 1999 20-25p.
- [3] KHAN, Mumit. *MingGw: Minimalist GNU For Windows*. www.mingw.org. 2000.
- [4] SCHILDT, H. C, *Completo e Total*, São Paulo: Makron Books, 1990. 889p.
- [5] SMART, Julian. *WxWindows, cross-plataform development for UNIX/Windows/MacOS*. www.wxwindows.org. 2000.
- [6] STROUSTRUP, Bjarne. *C++ a Linguagem de Programação*. Porto Alegre: Bookmam. 3ª ed. 823p.