

TRANSMISSOR DE TEMPERATURA MICROCONTROLADO E SISTEMA DE SUPERVISÃO

LOPES, Danien Leyes¹; PAIM, Guilherme Pereira¹; RADEL, Felipe Vesenick²; SCHMITZ, Gabriel Figueiredo²; ESPOSITO, Marcelo³.

¹ Graduação em Engenharia Eletrônica; ² Graduação em Engenharia de Controle e Automação;

³ Universidade Federal de Pelotas, Centro de Engenharias. marcelo.esposito@ufpel.edu.br

1 INTRODUÇÃO

Embora se tenha bastante intimidade com a temperatura, um conceito claro sobre ela não é muito trivial. Por exemplo, ao se utilizar o corpo humano como referência, tem-se apenas o conceito de quente e frio e mesmo assim não há um padrão para esse tipo de medição, visto que cada corpo humano apresenta sensações distintas, o que em termos físicos caracteriza apenas inferência. Todas as substâncias são constituídas por uma grande quantidade de moléculas, que se encontram em contínuo movimento. Quanto mais rápido for o movimento das moléculas, mais quente se apresenta o corpo e quanto mais lento o movimento, mais frio está o corpo. Esta condição pode ser descrita como um potencial térmico ou como uma energia efetiva da substância. A temperatura é o número dado a este atributo; a medição da energia em movimento (ECIL, 2010).

No presente trabalho é apresentada uma forma de monitoramento da temperatura por meio de um sistema de medição, composto por um *hardware* baseado em uma plataforma microcontrolada e um sistema SCADA (*Supervisory Control And Data Acquisition*). A interface foi desenvolvida utilizando-se a ferramenta Eclipse E3 (Eclipse, 2012). Sistema ou *software* supervisor é uma ferramenta de desenvolvimento de aplicativos que se presta a fazer a comunicação entre um computador e uma rede de automação e que traz ferramentas padronizadas para a construção de interfaces entre o operador e o processo. É parte fundamental de praticamente qualquer sistema de automação de processos e permite realizar as seguintes funções: monitorar variáveis do processo em tempo real (temperatura, pressão, nível, etc.), diagnosticar falhas ou condições indevidas através de alarmes e eventos, ajustar parâmetros do processo (*set-points*, ganhos de controladores, alarmes, etc.) e enviar comandos para o sistema (Radel *et al.*, 2012).

Para a medição da temperatura foi utilizada a plataforma de desenvolvimento livre Arduino UNO e um sensor de temperatura LM35. Ao comparar o valor medido por um sensor independente com o indicado em um *display* montado *in loco* e o valor indicado na tela do supervisor, constatou-se boa aproximação.

2 MATERIAIS E MÉTODOS

A unidade experimental foi estruturada sobre a plataforma de desenvolvimento livre Arduino UNO fabricado pela Smart Projects. Esta plataforma utiliza o circuito integrado (CI) microcontrolador ATmega328P, marca ATMEL. Em conjunto foi utilizado o sensor de temperatura LM35, marca National Semiconductor. Uma interface homem-máquina (IHM) foi criada a partir de um *display* LCD marca HITACHI. A comunicação foi realizada utilizando-se o padrão de troca de dados

binários via interface RS-232, para tanto foi necessário o uso do circuito integrado MAX232, marca Texas Instruments. O sistema supervisor foi desenvolvido utilizando-se a ferramenta Eclipse E3 em modo de Demonstração.

Os circuitos foram alimentados por uma fonte de alimentação de 9 Vcc, ajustada em 5 Vcc (nível TTL - *Transistor-Transistor Logic*), encontrada no próprio *kit* Arduino. Esta configuração foi usada para evitar problemas de referencial de tensão e porque o somatório das correntes era inferior aos 500 mA suportados pelo *kit*, segundo o manual do fabricante.

O sensor de temperatura LM35 é um semicondutor com precisão de $\pm 0,75^\circ\text{C}$ na sua faixa de operação de -55°C à 150°C . O sensor LM35 pode ser alimentado com uma tensão entre 4 e 30 Vcc e o consumo de corrente nominal é de aproximadamente $60\ \mu\text{A}$. Outra característica é a sua baixa impedância de saída, possibilitando a conexão direta com o microcontrolador, sem *buffers* ou amplificadores. Possui saída linear e proporcional em tensão com relação à temperatura em graus centígrados. A função de transferência estabelece a relação de $10\ \text{mV}/^\circ\text{C}$. O sensor de temperatura foi protegido por um invólucro de resina epóxi para que fosse possível realizar medições de temperatura em líquidos sem que houvesse a condução de energia entre os seus terminais.

A plataforma Arduino pode ser alimentada por uma fonte de corrente contínua ou uma bateria externa com tensão entre 6 e 20 V, possui entradas analógicas e digitais, comunicação serial e também conversor A/D (Analógico/Digital) de 10 *bits* de resolução. O programa foi desenvolvido em linguagem própria com compilador disponibilizado pelo fabricante.

O sensor de temperatura foi conectado a uma das entradas analógicas do Arduino permitindo assim a leitura dos dados através do conversor A/D. A tensão utilizada como referência para o conversor A/D foi de 1,23 Vcc, sendo esta ajustada com um *trimpot* (*Trimmer Potentiometer*) de 1 k Ω . Como a tensão de referência é proporcional ao valor amostrado, a calibração entre o sensor LM35 e o termômetro digital tipo espeto (sensor independente ao sistema) foi realizada variando-se este valor. A conversão do sinal analógico em digital resulta em um número inteiro entre 0 e 1023 que é armazenado como um número real. Este então é dividido por 10 para formar a casa decimal referente à medição da temperatura, assim a máxima temperatura a ser medida é de $102,3^\circ\text{C}$.

Utilizou-se um *display* LCD 16 x 2 (colunas x linhas) com luz de fundo para a construção da Interface Homem-Máquina. O LCD pode ser alimentado por uma tensão entre 2,7 e 5 Vcc e necessita de uma corrente de aproximadamente $135\ \mu\text{A}$ com o *backlight* ligado. Este *display* possui um controlador interno próprio e recebe os dados de forma paralela, cada *byte* precisa ser dividido em 2 *nibbles* sendo que apenas um 1 *nibble* deve ser enviado por vez pois são utilizadas 4 vias de barramento de dados. Os pinos de seleção são: RS, RW e E. O *bit* do pino RS é utilizado para a seleção do envio de dado (nível 1) ou de comando (nível 0). O *bit* do pino RW (*Read* ou *Write*) foi fixado em nível 0, ou seja, só permite escrever, porque não foi utilizada a memória do controlador interno do *display*. E o *bit* do pino E (*Enable*) é pulsado após o *nibble* estar disponível na sua entrada paralela, para apresentação do dado ou realização do comando desejado.

Através da biblioteca *LiquidCrystal.h* pré-programada pelo fabricante do Arduino, foi possível configurar a comunicação paralela de forma ágil, por meio de um parâmetro informa-se ao Arduino os pinos selecionados como saída para o LCD e com outro a configuração de envio de dados.

Todo o sistema comunica-se com o microcomputador através do padrão RS-232, porém para que fosse possível a utilização deste padrão com o Arduino foi necessário adequar os níveis de tensão de RS-232 para TTL e vice-versa. A solução para este problema foi encontrada com o uso do circuito integrado MAX232. Efetuou-se a montagem típica que consta na folha de dados deste CI e que consiste na ligação de quatro capacitores de 1 μF e um de 10 μF diretamente nos terminais do mesmo. Este CI foi alimentado com uma tensão de 5 Vcc e absorve uma corrente aproximada de 10 μA . A comunicação serial do Arduino foi realizada por meio de uma biblioteca que está implícita no compilador da IDE (*Integrated Development Environment*).

Esta troca de informações entre o microcomputador e a plataforma Arduino possibilitou a utilização do sistema supervisorio Elipse E3. O *layout* do supervisorio foi definido utilizando o programa Fritzing Beta para a criação do plano de fundo. Um *driver* ASCII para comunicação RS-232 foi gentilmente fornecido pela Elipse. O mesmo foi configurado de forma a interpretar os dados enviados, sendo que todas as informações necessárias para a configuração do *driver* foram obtidas a partir do material de ajuda disponibilizado pelo mesmo pacote da Elipse.

3 RESULTADOS E DISCUSSÃO

A interface principal do supervisorio é mostrada nas figuras 1 e 2. O supervisorio mostra uma imagem do *hardware* como plano de fundo e também um *display* no qual a temperatura medida é mostrada. Toda a prototipação foi realizada com uma matriz de contatos (*protoboard*), o que dificultou o processo devido ao mau contato. Para solucionar o problema *jumpers* foram confeccionados com cabos flexíveis, pinos estampados, soldados com estanho e isolados com tubo termocontratil. Percebe-se que a calibração e a validação experimental ocorreram com o sensor LM35 imerso em água. Uma vez que a medição expondo os sensores apenas ao ambiente não garante a homogeneidade das medidas e prejudica consideravelmente o tempo de resposta dos sensores. Embora a medição com o sensor LM35 foi dificultada pelo isolamento térmico e elétrico com resina, identificaram-se pequenas discrepâncias ($\pm 1^\circ\text{C}$) entre os três pontos de indicação da temperatura, na variação em estágios entre 20 e 80 $^\circ\text{C}$.

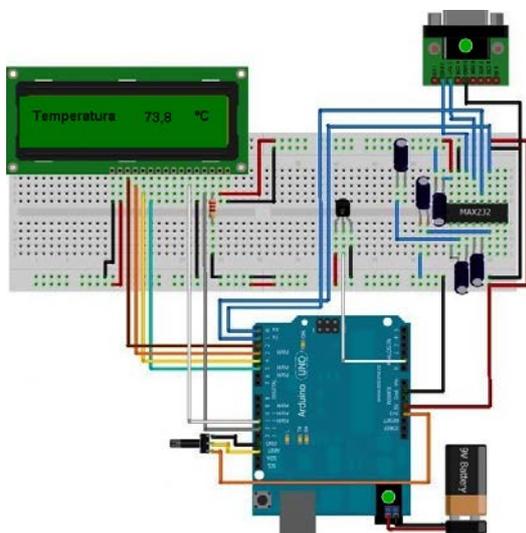


Figura 1 – Interface principal do supervisorio

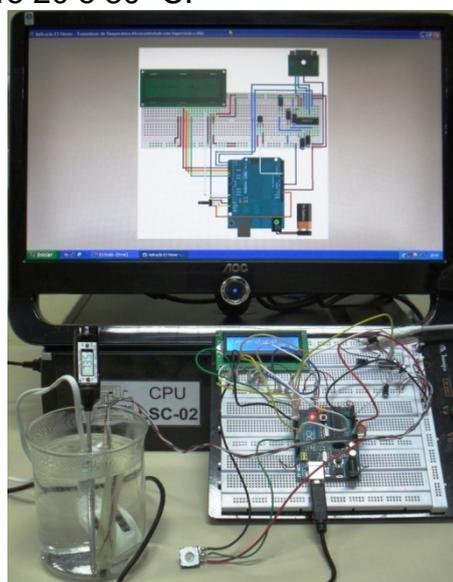


Figura 2 – Transmissor de temperatura.

4 CONCLUSÃO

O ambiente acadêmico proporciona sempre grandes desafios, entretanto tal atmosfera muitas vezes não prima pelo desenvolvimento de percepções que são sempre bem vindas no mercado de trabalho. Neste trabalho oportunizou-se o aprendizado de inúmeras habilidades como visão industrial, detecção de problemas e carências, busca e aprimoramento de soluções. Ao analisar as possibilidades advindas, percebeu-se que não bastam apenas boas soluções de engenharia, mas é necessária a habilidade construtiva e financeira que possibilite a implementação de um projeto de baixo custo e robusto.

Utilizando o conhecimento de programação em linguagem C, desenvolvido em disciplinas do ciclo básico do curso de engenharia e auxiliados pelas ferramentas e bibliotecas disponibilizadas pela plataforma Arduino foi possível acelerar o desenvolvimento do *hardware* e minimizar as possibilidades de falhas.

5 REFERÊNCIAS

ARDUINO, **Manual, software e hardware Arduino**. Disponível em: <<http://www.arduino.cc>>. Acesso em: 10 de Julho de 2012.

ATMEL CORPORATION, **Datasheet do microcontrolador ATmega328P**. Disponível em: <<http://www.atmel.com/devices/atmega328p.aspx>>. Acesso em: 13 de julho de 2012.

ELIPSE. **Instalador do Elipse E3**. Disponível em: <<http://www.elipse.com.br/>>. Acesso em: 17 de julho de 2012.

ECIL. **Sensores de temperatura – Princípios e aplicações**. 2010.

FRITIZING, **Instalador Fritizing Beta**. Disponível em: <<http://www.fritizing.org>>. Acesso em: 10 de julho de 2012.

HITACHI SEMICONDUCTOR, **Datasheet do display LCD 16x2**. Disponível em: <<http://pdf1.alldatasheet.com/datasheet-pdf/view/63673/HITACHI/HD44780.html>>. Acesso em: 13 de julho de 2012.

RADEL, Felipe Vesenick; SCHMITZ, Gabriel Figueiredo; LOPES, Danien Leyes; PAIM, Guilherme Pereira; ESPOSITO, Marcelo. Aplicação de controlador lógico programável e sistema de supervisão no controle de semáforos. In: **XXI CONGRESSO DE INICIAÇÃO CIENTÍFICA (CIC) e IV MOSTRA CIENTÍFICA (MC) – UFPel**. Pelotas, 8 a 11 de outubro de 2012. Anais do XXI CIC 2012.

TEXAS INSTRUMENTS, **Datasheet do MAX232**. Disponível em: <<http://www.ti.com/lit/ds/symlink/max232.pdf>>. Acesso em: 13 de julho de 2012.