

## **AQUISIÇÃO E TRANSFERÊNCIA DE DADOS VIA ETHERNET DE VARIÁVEIS ANALÓGICAS**

GUSTAVO WREGE GONÇALVES<sup>1</sup>; RUHAN ÁVILA DA CONCEIÇÃO<sup>2</sup>;  
MARCELO ESPOSITO<sup>3</sup>.

<sup>1</sup>Universidade Federal de Pelotas – [gustavo.wrege@gmail.com](mailto:gustavo.wrege@gmail.com)

<sup>2</sup>Universidade Federal de Pelotas – [ruhanconceicao@gmail.com](mailto:ruhanconceicao@gmail.com)

<sup>3</sup>Universidade Federal de Pelotas – [marcelo.esposito@ufpel.edu.br](mailto:marcelo.esposito@ufpel.edu.br)

### **1. INTRODUÇÃO**

Os sistemas de controle são parte integrante da sociedade moderna. Inúmeras aplicações estão à nossa volta: foguetes orbitando a terra, processos industriais, automobilismo, etc. Um sistema de controle consiste em subsistemas e processos construídos com o objetivo de se ter uma saída desejada com um desempenho desejado, dada uma entrada que representa a saída desejada. (NISE, 2009).

Variáveis de entrada geralmente são obtidas pelo sistema de controle por meio da aquisição de dados provenientes de sensores (sensor de temperatura, sensor de luminosidade, potenciômetro, acelerômetro, entre outros). Soma-se a isso, a variável de saída que é convertida para o mundo externo utilizando-se atuadores, como por exemplo: motores, sistema de aquecimento e arrefecimento, lâmpadas, etc.

Além da comunicação com o mundo externo, destaca-se também a comunicação interna de um sistema de controle, a qual visa realizar a comunicação entre os subsistemas e processos que constituem o mesmo. Entre estes, podemos destacar a comunicação serial e paralela, comunicação via rádio, comunicação ethernet, etc.

Este trabalho apresenta o desenvolvimento de um sistema de aquisição de dados utilizando um dispositivo embarcado, que é capaz de trabalhar simultaneamente com três sensores e um atuador. Os dados obtidos pelo sistema foram transferidos para uma aplicação em *Java* hospedada em um microcomputador através da comunicação ethernet.

### **2. METODOLOGIA**

A fim de testar o sistema desenvolvido, como atuador utilizou-se um motor de passo e como sensor um *encoder* magnético, o qual retorna a posição angular do atuador. Uma vez que o sistema conta com mais duas variáveis de entrada, estas serão ligadas a potenciômetros utilizados como transdutor de posição angular e transdutor de movimento linear.

Tanto os sensores quanto o atuador foram conectados a uma plataforma de desenvolvimento, a chipKIT<sup>TM</sup> MAX32<sup>TM</sup> Board (DIGILENT, 2011), que usa o microcontrolador PIC32MX795F512L (MICROCHIP, 2008). Este microcontrolador tem uma frequência de operação de 80 MHz, com 83 pinos de entrada e saída, sendo 16 analógicos, além de contar com diversas funções periféricas como, UART, SPI, I2C e saídas PWM.

O *encoder* magnético utilizado no projeto foi o AM4096 (RLS, 2010) da RLS. Este *encoder* é baseado no efeito Hall, ou seja, os sensores detectam a componente perpendicular do campo magnético. Este *encoder* possui alta

precisão, podendo aferir até  $1/4096^\circ$  de deslocamento angular e conta com diversas formas de saída: incremental, serial, comunicação UVW, senoidal e tensão linear. Neste projeto, a saída utilizada foi à tensão linear, onde para cada posição angular do motor de passo, o *encoder* retorna um nível de tensão entre 0 e 3,3 VCC (corrente contínua).

O motor de passo utilizado foi o 42HS03 da Leadshine (LEADSHINE, 2014), com o *driver* de potência DM164130-7 da Microchip (MICROCHIP, 2012) alimentado com uma fonte CC de 9 VCC e corrente máxima de 1 ACC. O motor de passo com *angle step* de  $1,8^\circ$  operou com esquema de ligação unipolar.

### 3. RESULTADOS E DISCUSSÃO

Visando efetuar a comunicação entre o MAX32 e o *encoder* magnético programou-se o *encoder* para retornar os dados lidos na forma de tensão com resposta linear, além disso, configuraram-se alguns outros parâmetros como o período por volta e também o sinal usado para a indicação do sentido de rotação. A programação deste dispositivo foi realizada através da comunicação serial I2C (NXP, 2014), também conhecida como TWI. Ao se realizar a comunicação I2C é necessário saber *a priori* o endereço do dispositivo. Este procedimento foi realizado com auxílio de um analisador lógico (Logic 8-channel USB Logic Analyzer Saleae). Ao fazer a requisição do endereço do *encoder*, este retornou um sinal de aceito (ACK - *acknowledge*) no endereço 0x08 (Figura 1).

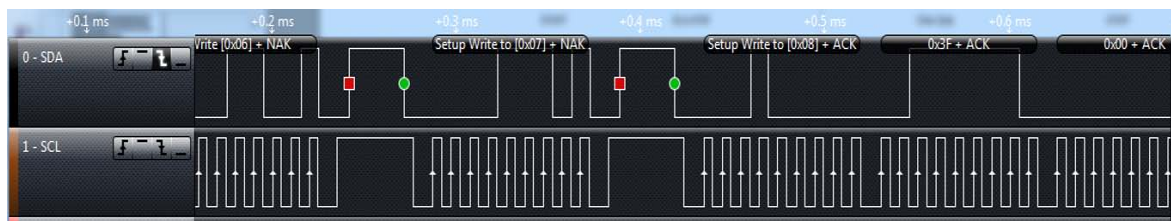


Figura 1. Endereço I2C do *encoder* magnético.

Para a ligação do motor de passo ao sistema de controle (MAX32) utilizou-se a biblioteca AccelStepper (MCCAULEY, 2010), que possui um desempenho razoável, capaz de suportar diversos tipos de motores de passo e de *drivers*, simultaneamente. Após alguns testes, foi definido que a velocidade máxima que poderia ser obtida com o motor seria de 400 passos por segundo (duas voltas por segundo) ou 120 rpm.

A operação do motor de passo pode ocorrer na forma de um degrau de posição, ou de velocidade ou de forma senoidal. A posição é dada em passos por segundo, mas pode ser convertida em graus, dado que cada passo do motor equivale a  $1,8^\circ$ .

O hardware utilizado para a comunicação ethernet foi a *shield* chipKIT™ Network Shield (DIGILENT, 2012), a qual agrega uma porta ethernet ao MAX32. Em termos de *software*, a comunicação ethernet foi viabilizada através do desenvolvimento de um servidor *HTTP* no MAX32, assim a aplicação que acessou o MAX32 a fez como sendo um cliente web.

O cliente web é uma aplicação desenvolvida em Java que plota os gráficos dos valores obtidos por meio dos *encoders* e também a velocidade estabelecida para o motor de passo. A comunicação entre o cliente e o servidor é efetuada periodicamente a cada segundo. Este tempo de acesso é controlado pelo cliente, que requisita ao servidor hospedado no MAX32 os dados do processo. A

periodicidade é realizada por meio da classe *Timer*, a qual faz parte da API padrão do Java.

A cada requisição efetuada pelo cliente, são transmitidas 25 amostras de cada *encoder* e do motor, ou seja, uma taxa de amostragem de 40 ms. O tempo de envio das amostras e a amostragem foram definidos em função de que quando um conjunto de amostras é enviado pelo canal ethernet uma operação bloqueante é executada e caso muitos dados sejam transmitidos o motor pode perder passos.

Na Figura 2 as linhas pontilhadas representam a comunicação efetuada via ethernet. Os blocos *ENC* e *M* representam o *encoder* e o motor de passo, respectivamente. Ao ligar o sistema, o motor é direcionado até a posição zero, aguardando então uma requisição HTTP para dar início ao modo de operação do motor. Na primeira requisição identificada pelo sistema, um código de aceitação é enviado para o cliente web e o motor é iniciado no modo de operação definido. A cada requisição posterior, é enviado o conjunto de 75 amostras dos três *encoders* (25 de cada) e 25 amostras da velocidade de atuação do motor. O cliente então interpreta os dados recebidos e cria os gráficos. Durante a execução, a leitura das amostras é realizada por uma função de interrupção, controlada por um *timer* do MAX32, que adquire a cada 40 ms uma amostra de cada um dos canais analógicos de leitura ou de escrita.

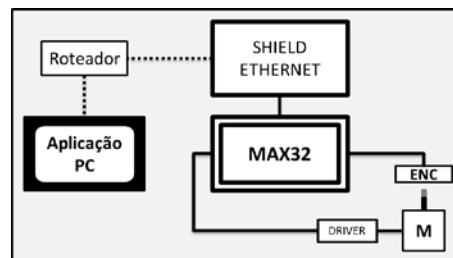


Figura 2. Diagrama de blocos do sistema de aquisição e transferência de dados via ethernet.

Através da abordagem de transferência de dados via ethernet, o usuário pode obter informações do sistema remotamente, desde que o MAX32 e o microcomputador que hospeda a aplicação estejam na mesma rede ethernet.

A Figura 3 apresenta os gráficos gerados pela aplicação desenvolvida. O motor promove um deslocamento de forma senoidal a carga acoplada ao seu eixo e a leitura da velocidade angular é realizada pelo *encoder* magnético (*Encoder 1*).

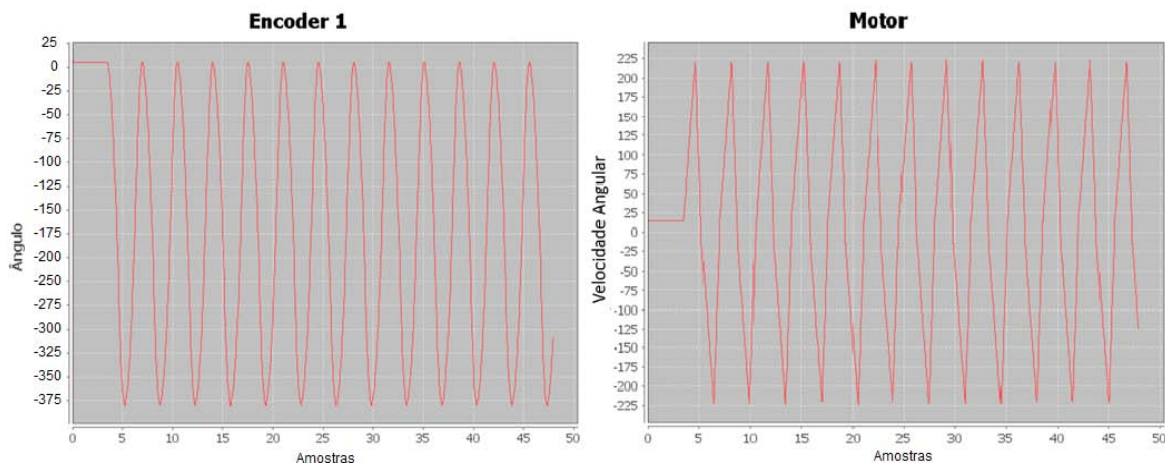


Figura 3. Gráficos gerados pela aplicação em Java.

## 4. CONCLUSÕES

Este trabalho apresentou o desenvolvimento de um sistema de aquisição e transferência de dados via ethernet com capacidade para três canais analógicos de leitura e um canal de escrita. Os testes e simulações realizadas demonstraram que o sistema supervisorio projetado possui características importantes como flexibilidade e robustez. A leitura da posição angular e do movimento linear (encoder) e a escrita do número de passos ou da velocidade angular (motor) podem ser plenamente utilizadas no controle do movimento oscilatório característico de um pêndulo invertido. Do ponto de vista da instrumentação de processos dois pontos são relevantes no presente trabalho, o uso de um *encoder* programável, com sistema de medição sem contato e o uso da tecnologia ethernet. Em ambos os casos o valor monetário de dispositivos semelhantes para uso industrial são altos, seja pela alta resolução, ou pela necessidade de mão de obra qualificada para o desenvolvimento de algoritmos complexos e bem estruturados.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

- DIGILENT. **ChipKit MAX32 Board Reference Manual**. 25 jul. 2011. Revision. Acessado em 14 jul. 2014. Online. Disponível em:  
[http://ww1.microchip.com/downloads/en/DeviceDoc/chipKIT%20Max32\\_rm.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/chipKIT%20Max32_rm.pdf)
- DIGILENT. **chipKIT™ Network Shield Board Reference Manual**. 27 jun. 2012. Revision. Acessado em 14 jul. 2014. Online. Disponível em:  
[http://www.digilentinc.com/Data/Products/CHIPKIT-NETWORK-SHIELD/chipKIT Network Shield\\_rm.pdf](http://www.digilentinc.com/Data/Products/CHIPKIT-NETWORK-SHIELD/chipKIT Network Shield_rm.pdf)
- LEADSHINE. **42HS Series Hybrid Stepping Motors**. Acessado em 14 jul. 2014. Online. Disponível em:  
<http://ww1.microchip.com/downloads/en/DeviceDoc/Leedshine%2042HS03%20Stepper%20Motor%20Datasheet.pdf>
- MCCAULEY, M. **AccelStepper library**. 2010. Acessado em 14 jul. 2014. Online. Disponível em:  
<http://www.airspayce.com/mikem/arduino/AccelStepper/>
- MICROCHIP. **PIC32MX Family Reference Manual**. 2008. Acessado em 14 jul. 2014. Online. Disponível em:  
[http://hades.mech.northwestern.edu/images/2/21/61132B\\_PIC32ReferenceManual.pdf](http://hades.mech.northwestern.edu/images/2/21/61132B_PIC32ReferenceManual.pdf)
- MICROCHIP. **F1 LV Evaluation Platform Motor Control Add-On User's Guide**. 2012. Acessado em 14 jul. 2014. Online. Disponível em:  
<http://ww1.microchip.com/downloads/en/DeviceDoc/41629A.pdf>
- NISE, N. S. **Engenharia de Sistemas de Controle 3ª Edição**. LTC, 2009.
- NXP. **I<sup>2</sup>C-bus specification and user manual**. 4 abr. 2014. Revision 6. Acessado em 16 jul. 2014. Online. Disponível em:  
[http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)
- RLS. **AM4096 – Angular magnetic encoder IC**. 10 mar. 2010. Acessado em 14 jul. 2014. Online. Disponível em:  
[www.rls.si/media/pdf/en/95a93b667afa49ae98ddb16157d82bee.pdf](http://www.rls.si/media/pdf/en/95a93b667afa49ae98ddb16157d82bee.pdf)