

PROTÓTIPO DE TELEMETRIA DESTINADO À FÓRMULA SAE

Julio Cesar Ferreira Lima¹; Melisse Pontes Cabral²; Kaio Vítor Gonçalves de Freitas³;
Jandilson Almeida Bandeira⁴; Rômulo Rodrigues da Silva⁵

1 Universidade Federal do Ceará, Graduando em Engenharia Elétrica,
julio_flima@hotmail.com

2 Universidade Federal de Campina Grande, Graduanda em Ciência da Computação,
melisse.cabral@ccc.edu.br

3 Universidade Federal de Campina Grande, Graduando em Engenharia Elétrica,
kaio.freitas@ee.ufcg.edu.br

4 Universidade Federal de Campina Grande, Graduando em Engenharia Elétrica,
jandilson.bandeira@ee.ufcg.edu.br

5 Universidade Federal de Campina Grande,
romulo@df.ufcg.edu.br

Introdução

No mundo da Formula 1, uma das principais revoluções que a Engenharia Elétrica causou foi a aquisição de dados para análises posteriores, para que assim as equipes pudessem melhorar seus carros com base em dados de teste. Outra grande revolução foi a análise dos valores em tempo real, que deu maior dinâmica no dia da corrida (Technical F1, 2017).

Esse trabalho propõe o desenvolvimento de um protótipo de telemetria no destinado a Fórmula SAE. Usando como plataforma de aquisição de dados, o microcontrolador ATmega368, como o dispositivo de comunicação Xbee-Pro SB2, e um sistema de supervisão desenvolvido em C#. Com o sistema de supervisão, 14 parâmetros referentes ao carro foram capturados e podem ser analisados em tempo real e em testes posteriores, podendo se expandir, por código, para mais quantidades. O raio para receber em campo aberto é de 1,5 km, e podendo ser expandido usando o conceito de malha ou árvore de cluster, design de rede.

Metodologia

A plataforma utilizada nesse projeto é o Arduino que tem como unidade de processamento o ATmega328, possui alguns contemporâneos paradigmas de programação e é codificada em uma adaptação da linguagem C.

Seja usando sensores digitais ou analógicos a adaptabilidade do ATmega328 no campo da instrumentação, torna-o favorito na comunidade mundial de desenvolvimento voltado a hardware. Se não houver compatibilidade lógica entre os sensores e microcontrolador, com apenas alguns componentes, como drivers, o microcontrolador torna-se perfeitamente adaptável uma grande escala de sensores.

1) Sensores:

a) Contador: Um dos parâmetros mais importantes em um carro são a velocidade e o RPM, sendo a primeira medida com a colocação de uma sensor na roda e segunda no eixo de tração. Desse modo que a cada giro é detectado uma rotação, de posse da velocidade angular do motor e da roda, é fácil obter-se a rotação do motor e a velocidade do carro. Esses dois parâmetros podem ser aquisitados por um sensor indutivo, que quando submetido a uma tensão em seus terminais, varia o estado da saída quando submetido a variações de presença e não presença de um metal. Esse tipo de sensor é do tipo discreto e digital, necessitando por vezes de uma interface de conversão, que pode ser feita por um divisor de tensão, acoplamento via regulador de tensão por ICs como LM7805 ou ULN2003.

b) Estados: A posição da marcha é uma variável de estado, que é captada por um botão montado em um dispositivo chamado borboleta da marcha, que assim que pressionado envia um valor nível alto para o microcontrolador que atuará em outro dispositivo, passador de marchas. No entanto, esse valor será guardado e mostrado ao piloto, pois é referente a marcha atual. O tipo de configuração necessária a esse tipo de sensor é um resistor em pull-down, alinhado a um capacitor em paralelo para evitar o efeito de ruídos causados pela chave mecânica.

c) Moderno: O sensor DS18B20 é um sensor de temperatura e possui protocolo 1-wire para comunicação. Neste trabalho ele foi utilizado com encapsulamento imersível de aço inox, ideal para aplicações industriais como também para medição de temperatura do tange de óleo. O DH22 é outro sensor utilizado no carro, que mede a humidade e temperatura do ar. Ele comunica-se via protocolo single-bus, onde necessita também de apenas um fio para barramento de comunicação.

d) Rastreamento de Posição: Alguns sensores tem a função de fazer o rastreamento da posição de um dispositivo, como o volante, o freio, o acelerador, o nível do óleo e o nível da gasolina. Eles podem ser rastreados por potenciômetro, que submetido a uma diferença de potencial em seus extremos, em caso de variação da posição do eixo, o terminal do meio apresentará uma variação de tensão, que pode ser lida em uma entrada analógica.

2) Comunicação: O módulo X-bee precisa de um dispositivo para interfacear seu nível lógico, pois apesar de ser TTL ele trabalha em um nível de tensão de 3.3V, e um conversor serial-USB para interfacear a comunicação com o computador. Apesar de as interfaces serem projetos abertos e passíveis de replicação, nesse trabalho foram utilizadas placas comerciais. Nesse trabalho foi testado com topologia de comunicação do tipo Point-to-Point, bastando setar uma configuração interna do X-Bee para router e o endereço de ID do X-Bee (Digi X-CTU; 2017) que se desejou comunicar-se, para ambos X-Bees. A comunicação com o ATmega328 se dá por meio de RS232, assim o X-Bee espelha os dados recebidos na sua entrada RS232 e os codifica em protocolo Zigbee. Por sua vez, o outro X-Bee faz o processo reverso, enviando no fim bytes na entrada serial do computador para que o supervisor armazene e processe. Buscando o melhor desempenho o *baudrate* adotado foi de 115200.

3) Processo do Protocolo

Usou-se apenas quatro dígitos para apresentar os valores de cada sensor. Partindo do pressuposto que zeros a esquerda em nada acrescentam em grandeza matemática e que cada variável possui 4 caracteres numéricos que se multiplicados cada um pela sua contribuição em unidades, dezenas, centenas, milhares e ao fim somados, compreenderão um número na base decimal e não mais apenas um caractere. Dessa forma foi desenvolvido um padrão de arranjo dos dados em uma *string* de caracteres. Em meio a *string*, o caractere “?” foi colocado para representar o início da *string*, o caractere “!” para representar o fim e o caractere “&” foi usado como espaçador entre as variáveis.

4) Sistema Supervisor

No desenvolvimento do software a linguagem escolhida foi C# que permite o desenvolvimento rápido de aplicativos executados no .NET Framework e mantém a expressividade de C. A estrutura desenvolvida, foi dividida em três microambientes: Resources, Program, Form.

Há apenas dois *namespaces* responsáveis pelo funcionamento completo do sistema. O *namespace* nomeado de “Cockpick” é responsável por realizar acesso direto ao “Database”.

A recepção de dados pela porta serial é feita por uma biblioteca nativa no C# a SerialPort (Microsoft; 2016). A utilização de métodos foi protegida contra múltiplos acessos a mesma porta por meio de um bloco “try/catch”. O tratamento de dados usado, trata-se da conversão dos bytes recebidos via porta serial em números mensuráveis e prontificados para realizar comparações e cálculos.

O maior benefício do projeto é a análise posterior dos dados para que seja possível identificar falhas e pontos possíveis de otimização de performance. Pensando nisso foi visto que o melhor modo para salvar esses dados seria em arquivos que pudessem ser lidos em softwares de edição e visualização de planilhas, devido à alta difusão e fácil usabilidade.

Assim foi decidido adotar armazenamento em frames de dados, utilizando XML. Desse modo, sendo possível que cada arquivo gerado seja interpretado por softwares que tratem de planilhas de frames como o Excel. Dada a necessidade de proteger o sistema contra travamentos e conseqüentemente a perda de dados foi inserido um sistema de salvamento de dados automático que não causa interferência na usabilidade.

Resultados e Discussão

Pela impossibilidade de concepção de um carro inteiro para implementação e teste de sensores, devido ao custo e tempo onerosos, foi desenvolvido um protótipo adaptado de um joystick destinado inicialmente a video-games. Todo o hardware de interface foi adaptado na própria placa do X-Bee devido a sua simplicidade. Foram adicionadas duas antenas de 5dB para melhorar a qualidade de transmissão e captação do sinal.

Objeto deste trabalho é apenas um protótipo, portanto, não houve a preocupação com a durabilidade referente a interferências mecânicas futuramente provocadas pela vibração natural do carro. Essa é uma preocupação séria, que em caso de desenvolvimento real, deverá ser sanada utilizando amortecedores de borracha para o case de sustentação dos dispositivos elétricos, junto com a eliminação de contatos via conectores, necessitando que todos eles sejam devidamente soldados em uma placa mãe.

Conclusões

O protótipo deve e será melhorado quanto a sua latência em resposta em relação ao computador, porém tal latência dado o tempo de análise lento, característica intrínseca do ser humano, não compromete em nada sua análise. Também seria interessante programar o código relativo ao Atmega328 na IDE do Arduino, usando o paradigma da programação orientada a objetos (OOP), visando deixar o código mais simples de entender.

Já que esse tipo de tecnologia é utilizado pelo esporte mais caro do mundo, então possui elevada valorização dentro do mercado, o que torna o custo-benefício um diferencial.

Apesar do projeto se tratar apenas de um protótipo, ele atendeu todos os objetivos propostos. Com advento desse, dado pelo o árduo processo até então, a implementação de futuros sensores terá sua dificuldade muito reduzida, se limitando basicamente a apenas a uma implementação de sensores no microcontrolador. Mesmo por tratar-se de um hardware simples, de baixo custo, foi possível notar uma imensa robustez, pois passados meses da realização do projeto, ele ainda funciona e inclusive já fora objeto de implementação de outro projeto, baseado em seu código fonte.

Palavras-Chave: Telemetria; ; Formula SAE; Arduino; X-Bee; C#.

Referências

- Arduino, Language Reference. 2015. Disponível em: <<https://www.arduino.cc/en/Reference/HomePage>>. Acesso em 22 de Março de 2017.
- Digi X-CTU, 2017. Disponível em: <www.digi.com>. Acesso em 22 de Março de 2017.
- Microsoft, “Visual Studio Documentation”, 2016. Disponível em: <<https://docs.microsoft.com/en-us/visualstudio>>. Acesso em 22 de Março de 2017.
- Technical F1, Formula 1 Dictionary, “Telemetry”. 2017. Disponível em: <<http://www.formula1-dictionary.net/telemetry.html>>. Acesso em 26 de Fevereiro de 2017.