



I Congresso Nacional das Engenharias da Mobilidade

Engenharia Mecatrônica – Modelagem, análise e simulação

Preparando o microcontrolador MSP-430 para operar em uma rede de sensores.

Eugênia Giancoli Jabour¹, dijabour@gmail.com

Frederico José Dias Möller¹, fredericomollerper@gmail.com

Luiza Bartels de Oliveira¹, boliveira.luiza@gmail.com

¹: Instituto Federal de educação ciência e tecnologia do sudeste de Minas Gerais, Campus Juiz de Fora, Rua Bernardo Mascarenhas, 1283 – Bairro Fábrica, Juiz de Fora – Minas Gerais, CEP: 36080-001. e-mail: secretaria.jf@ifsudestemg.edu.br

Resumo: No presente trabalho, uma série de implementações foram realizadas para possibilitar que microcontroladores MSP-430 funcionassem de modo a simular uma rede de sensores. Inicialmente foram implementadas duas funções: uma de forma a decompor valores inteiros em um vetor de caracteres; e outra de forma a recompor o valor inteiro. Foi criado um programa que recebe caracteres diretamente de uma porta COM, aplicando a função de recomposição neles. Em seguida, foi projetada uma placa que liga diretamente o MSP-430 como o módulo XBee.

Palavras-chave: comunicação serial, msp430, xBee, redes de sensores.

1. Sobre o trabalho científico

Devido a necessidade de montar uma rede de sensores para testar um protocolo de transporte, tivemos que preparar o microcontrolador MSP-430 para trabalhar com o módulo xBee, um módulo de comunicação sem fio baseado no protocolo IEEE 802.15.4 enviando dados para uma base escoadora.

Para a comunicação UART tal microcontrolador, no entanto, exige que seu programa quebre os dados a serem enviados em valores do tipo char e os armazene em uma variável de buffer de transmissão, o que não é interessante, uma vez que sensores transmitem dados do tipo int e float.

Para sobrepor esse obstáculo, a princípio, foi implementada uma função itoa(), baseada em função de mesmo nome usada na linguagem c++. Essa função varre o valor inteiro transformando cada dígito do número em um valor char correspondente ao dígito na tabela ASCII e o armazena em um vetor de char. Em seguida, uma outra função recebia esse vetor e transmitia cada elemento, transmitindo por fim o valor 13, que na tabela ASCII corresponde ao comando “carriage return”, ou enter, separando assim o último valor transmitido do próximo a ser transmitido. No entanto, essa função exige um número considerável de comandos e em conjunto com outras rotinas poderia acabar se tornando um peso para o microcontrolador. Além disso, é possível que alguns nós da rede tenham que manipular os dados recebidos de outros nós e isso não poderia ser feito facilmente sem uma função tipo atoi(), que revertesse o processo e que seria tão custosa para o microcontrolador quanto a itoa().

Por isso foram implementadas, duas novas funções para transformar valores int em um vetor de quatro valores char e outra para reagrupar esse vetor em um valor tipo int. A vantagem no uso de tais funções, ao invés de transformar cada dígito do valor int em uma informação char é que o valor matemático do número não é perdido, além disso a quantidade de valores char a ser enviada e sempre quatro, o que não seria possível se fosse transmitido cada dígito como um valor char.

No entanto, se a base escoadora estiver usando um programa telnet comum para receber as informações da rede de sensores, os valores char recebidos poderão ser convertidos para seus caracteres correspondentes da tabela ASCII antes que se tenha a possibilidade de manipulá-los. Por isso, neste trabalho também, foi escrito um programa em c++ que recebe os dados por uma porta COM, agrupa-os e os transforma em um valor tipo int.

2. Funções Implementadas

2.1. Função decompor()

O objetivo desta função é transmitir valores inteiros poucas operações. Ela parte da premissa que, se um valor do tipo int é composto de quatro bytes e um valor char é composto por um byte, então é possível quebrar o valor inteiro em quatro valores char.

Inicialmente a função convertia o valor int para um número de base 256, onde cada casa numérica era representada por um valor char. Mas, de acordo com o fabricante do microcontrolador, operações de divisão são pouco recomendadas e por isso optamos por usar operações bit a bit, realocando grupos de oito bits do valor inteiro a ser transmitido para a posição inicial do mesmo, e os armazenando em variáveis tipo char.

2.2. Função recompor()

A função recompor() inicialmente realizava a conversão dos valores de um vetor tipo char de tamanho 4 para um valor int, tratando cada posição do vetor como um dígito de um número de base 256. Com a mudança na função decompor a função recompor foi reescrita, de forma a ler o endereço de memória do vetor de valores char recebidos e converter para um valor inteiro.

3. Programa de base

O programa da base escoadora, tem por finalidade receber os valores transmitidos pela rede de sensores e gravá-los em um arquivo. Para redes de MSP-430 que convertem valores tipo int e float em vetores de valores char, convertendo dígito a dígito em seu correspondente da tabela ASCII, programas telnet como o hyperterminal (windows) e xterm (linux) servem bem ao propósito.

No entanto, para redes operando com a função decompor(), o uso desses programas é um problema, uma vez que a conversão imediata dos valores tipo char recebidos pela base não é interessante. Tal conversão pode gerar perda de informação, pois alguns valores, quando convertidos em ASCII geram comando que não são gravados no arquivo final. O valor inteiro 13, por exemplo, quando convertido em um vetor de char pela função decompor, se torna o vetor (13,0,0,0), o programa hyperterminal interpretará ele como um comando “carriage return” (enter) e três “null”.

Para substituir o hyperterminal, foi criado um programa que usa a biblioteca windows.h e suas funções CreateFile() e ReadFile(), para abrir e ler uma porta COM do computador. Os dados são salvos em um arquivo de texto, que possibilita o acompanhamento das medições realizadas.

4. Shield

Para efetuar a ligação física do microcontrolador com o módulo xBee de forma simples, foi projetado um shield. A configuração de ligações mínimas, previstas no datasheet do xBee foi escolhida como base para

as ligações efetuadas no shield. Com essa configuração, os pinos de VCC, GND, 1.1 e 1.2 do MSP-430 são ligados aos pinos 1,10,2,3 (VCC,GND,Data Output,Data Input)do xBee respectivamente. Os pinos 1.1 e 1.2 do MSP-430, em modo de comunicação UART, representam respectivamente sua entrada e saída de dados (Data Input e Data Output).



Fig. (4) – Protótipo do shield MSP-430 / xBee.

5. Considerações futuras

Como a função decompor transforma um dado tipo int em um vetor de tipo char com tamanho 4, o microcontrolador que estiver usando ela, sempre vai precisar transmitir quatro bytes para enviar um valor int completo. Isso pode ser vantajoso em relação a uma função que transforme cada dígito do valor int em um valor char e envie, no caso de números com mais de quatro dígitos, pois a função decompor enviará menos pacotes e com isso haverá uma chance menor de perda de informação.

No entanto, a perda de um byte enviado pela função decompor, terá um impacto maior sobre a informação do que um byte enviado por uma função que converte cada dígito em seu correspondente da tabela ASCII. Como a informação é enviada e reagrupada sequencialmente com as funções decompor e recompor, um byte perdido acarretará em erro em todo o resto da informação a ser transmitida, enquanto uma informação transmitida da outra forma (dígitos convertidos em valores char), um byte perdido só acarretará problema no valor int ao qual ele pertence, ou no máximo, vai deixar de separar dois valores tipo int que tenham vindo em sequência. Um futuro teste, comparando o desempenho do conjunto função decompor() e recompor() (no programa de base), contra uma função itoa() e o programa hyperterminal poderá determinar qual função valerá a pena para cada situação.

6. Referências

- 1) DIGI INTERNATIONAL INC (Estados Unidos). **XBee®/XBee-PRO® RF Modules**. Disponível em: <<https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>>. Acesso em: 23 set. 2009.
- 2) DENVER, Allen. **Serial Communications in Win32**. Disponível em: <<http://msdn.microsoft.com/en-us/library/ms810467.aspx>>. Acesso em: 11 dez. 1995.



**I Congresso Nacional das
Engenharias da Mobilidade**

**Mechatronics Engineering - modeling, analysis and simulation
Preparing the MSP-430 microcontroller to operate on a sensors
network**

**Eugênia Giancoli Jabour¹, dijabour@gmail.com
Frederico José Dias Möller¹, fredericomollerper@gmail.com
Luiza Bartels de Oliveira¹, boliveira.luiza@gmail.com**

¹: Instituto Federal de educação ciência e tecnologia do sudeste de Minas Gerais, Campus Juiz de Fora, Rua Bernardo Marcarenhas , 1283 – Bairro Fábrica, Juiz de Fora – Minas Gerais, CEP: 36080-001. e-mail: secretaria.jf@ifsudestemg.edu.br .

Abstract: At the present study, a range of implementations have been realized to enable the MSP-430 microcontrollers work with a purpose of simulating a network sensor. At first two functions have been implemented: one in order to decompose integer values in a vector of characters; and other in order to rebuild the integer value. A program, which receives characters directly from a serial port, have been written, applying the rebuilt function. Then, a shield was projected. It directly connects the MSP-430 with the XBee module.

Key-words: *serial communication, msp430,xBee, sensors network.*