
MICROCONTROLADORES – LABORATÓRIO 04 e 05

Flávio Henrique A. dos Santos

flaviohenriqu@gmail.com

Cidade Universitária Profº. José Aloísio de Campos

Av. Marechal Rodon, s/n Jardim Rosa Elze

CEP - 49100-00

São Cristóvão - SE

Marsol Luz Araújo

marsollaraujo@gmail.com

Cidade Universitária Profº. José Aloísio de Campos

Av. Marechal Rodon, s/n Jardim Rosa Elze

CEP - 49100-00

São Cristóvão - SE

Resumo: Este artigo tem o intuito de demonstrar os resultados obtidos nas experiências realizadas no 4º Laboratório da disciplina Microcontroladores. Para a realização das três experiências deste laboratório foi utilizado um *protoboard* com alguns periféricos associados ao microcontrolador DSPIC30F4011 tais como, *LEDs*, LCD 16x2, teclado e ventoinha; um gravador de PIC, os *softwares MPLAB C30, PICKit2* e o *HyperTerminal*; e um cabo serial. Os conceitos utilizados neste laboratório foram Comunicação Serial e Conversor A/D.

Palavras Chave: Microcontrolador, DSPIC30F4011, comunicação serial, conversor A/D.

Abstract: This article aims to demonstrate the results obtained in experiments on the 4th Microcontrollers Laboratory discipline. For the realization of the three experiments of this lab used a breadboard with some peripherals associated with the dsPIC30F4011 microcontroller such as LEDs, 16x2 LCD, keypad and fan; recorder PIC, MPLAB C30 software, and PICKit2 HyperTerminal, and a serial cable. The concepts used in this lab Serial Communication Converter and A/D.

Keywords: Microcontroller, dsPIC30F4011, serial communication, A/D converter.

1 INTRODUÇÃO

Serial é um protocolo muito comum para comunicação de dispositivos que vem como padrão em quase todo computador. A maioria destes inclui portas seriais baseadas em RS-232, que é uma norma que estabelece o padrão de conector e níveis de tensão. A ideia da comunicação é a seguinte: envia e recebe bytes de informação um bit de cada vez. Embora esta seja mais

lenta que a comunicação paralela, que permite a transmissão de um byte inteiro por vez, ela é mais simples e pode ser utilizada em distâncias maiores.

A porta serial é facilmente encontrada em microcontroladores, o que permite a comunicação com um PC e outros dispositivos. Assim, quando em conjunto com um conversor A/D (também comum em microcontroladores), pode ser aplicado em um sistema de aquisição de dados. A função do conversor A/D é fazer a conversão de uma grandeza analógica em uma informação digital que possa ser processada.

As experiências realizadas neste laboratório misturam os conceitos de comunicação serial e conversor A/D, e serão descritos a seguir:

- Experiência 1: implementar um sistema de aquisição de dados, que terá um menu, mostrado no PC, onde se poderá selecionar o tempo de aquisição e a entrada. A aquisição será manual para as entradas analógicas e serão 16 conversões na frequência de 8 KHz. Deve-se tirar a média das tensões destas para serem mostradas no PC e no LCD. A UART deverá ter um *baud rate* de 19200;
- Experiência 2: escrever um programa onde o conversor A/D terá tanto o início da aquisição quanto a parada controlado pelo teclado do PC. Configurar o conversor A/D no modo de amostragem manual, com 16 conversões a uma frequência de 6 KHz e a média destas deve ser mostrada no PC. A UART deverá ter um *baud rate* de 19200;
- Experiência 3: controlar a velocidade da ventoinha por um potenciômetro ligado ao pino RB8 e controlar a intensidade dos *LEDs* (RB0 a RB6) por um potenciômetro ligado ao pino RB7. Fazer uma função que receba o valor binário da saída do A/D (para cada uma das situações) e a converta em um sinal PWM, onde o valor 0

representa um *duty cycle* de 0% e 1023 representa 100%. Configurar o conversor A/D no modo de amostragem manual. O conversor A/D fará 16 conversões a uma frequência de 8 KHz e a média destas será utilizada como *duty cycle*.

2 COMUNICAÇÃO SERIAL

O módulo UART (*Universal Asynchronous Receiver Transmitter*) é um dos módulos seriais I/O disponíveis na família DSPIC30F. A UART é um sistema assíncrono full – duplex (envia e recebe dados ao mesmo tempo) que pode se comunicar com dispositivos periféricos, tais como PCs e interfaces RS-232 e RS-485. Ele consiste de um gerador de *baud rate*, um transmissor e um receptor assíncrono.

O gerador de *baud rate* é de 16 bits. O registrador UxBRG controla o período de um timer *free running* de 16 bits. As equações (1) e (2) a seguir mostram como se calcula o *baud rate*:

$$\text{Baud rate} = \frac{F_{cy}}{16 * (UxBRG + 1)} \quad (1)$$

$$UxBRG = \frac{F_{cy}}{16 * (\text{Baud rate})} - 1 \quad (2)$$

Em relação ao transmissor, é o registrador de deslocamento UxTSR (*Transmit Shift Register*) que faz o papel principal. Ele obtém seus dados desde o buffer FIFO de transmissão através do registrador UxTXREG, e não é carregado até que o bit de parada tenha sido transmitido desde a carga prévia. Tão pronto como o bit de parada é transmitido, o UxTSR é carregado com novos dados desde o registrador UxTXREG (se disponível). Para configurar a UART como transmissor as etapas abaixo devem ser seguidas:

- Inicializar o registrador UxBRG para o apropriado *baud rate*;
- Configurar o número de bits de dados, número de bits de parada e seleção de paridade escrevendo nos bits PDSEL<1:0> (UxMODE<2:1>) e STSEL (UxMODE<0>);
- Caso sejam desejadas interrupções quando se transmite, configurar o bit de controle UxTXIE no registrador IECx. Especificar a prioridade para a interrupção de transmissão usando os bits de controle UxTXIP<2:0> correspondente ao registrador IPCx. Também selecionar o modo de interrupção de transmissão escrevendo o bit UTXISEL (UxSTA <15>);
- Habilitar o módulo UART setando o bit UARTEN (UxMODE<15>);
- Habilitar a transmissão setando o bit UTXEN (UxSTA<10>), o qual também setará o bit UxTXIF. O bit UxTXIF será zerado na rotina de software que serve à interrupção de transmissão da UART. A operação do bit UxTXIF é controlada pelo bit de controle UTXISEL;

- Carregar o dado para o registrador UxTXREG (começa a transmissão). Se a transmissão de 9 bits tem sido selecionada, carrega uma palavra. Se uma transmissão de 8 bits é usada, carrega um byte. Dados podem ser carregados dentro do buffer até que o bit de status UxTXBF (UxSTA<9>) é setado.

Já no receptor, é o registrador de deslocamento de recepção UxRSR (*Receive Shift Register*) que faz o papel principal. O dado é recebido no pino UxRX e é enviado para o bloco de recuperação de dados. O bloco de recuperação de dados opera a 16 vezes o *baud rate*, enquanto que o deslocador serial de recepção opera ao *baud rate*. Depois da amostragem do pino UxRX pelo bit de parada, o dado recebido em UxRSR é transferido para a FIFO de recepção (se esta está vazia). Para configurar a UART como receptor as etapas abaixo devem ser seguidas:

- Inicializar o registrador UxBRG para o apropriado *baud rate*;
- Configurar o número de bits de dados, número de bits de parada e seleção de paridade escrevendo nos bits PDSEL<1:0> (UxMODE<2:1>) e STSEL (UxMODE<0>);
- Se interrupções são desejadas, então configurar o bit UxRXIE no registrador IECx. Especificar a prioridade para a interrupção usando os bits de controle UxRXIP<2:0> no registrador IPCx. Também, selecionar o modo de interrupção de recepção setando os bits URXISEL<1:0> (UxSTA<7:6>);
- Habilitar o módulo UART setando o bit UARTEN (UxMODE<15>);
- As interrupções de recepção dependerão dos bits de controle URXISEL<1:0>. Se as interrupções de recepção não são habilitadas, o usuário pode ter qualquer valor no bit URXDA. O bit UxRXIF será zerado na rotina de software que serve à interrupção de recepção;
- Ler dados desde o buffer de recepção. Se uma transmissão de 9 bits tem sido selecionada, ler uma palavra. Qualquer outra coisa, ler um byte. O bit de status URXDA (UxSTA<0>) será setado sempre que o dado esteja disponível no buffer.

A UART usa um formato padronizado de um bit de *start*, oito ou nove bits de dados, e um ou dois bits de *stop*. A paridade é suportada pelo hardware, e poderia ser configurado pelo usuário como par, ímpar ou sem paridade. O número de bits de dados, bits de parada e a paridade, são especificados nos bits PDSEL<1:0> (UxMODE<2:1>) e STSEL (UxMODE<0>). A UART transmite e recebe primeiro o bit menos significativo (LSB). O transmissor e o receptor da UART são

funcionalmente independentes, mas usam o mesmo formato de dados e *baud rate*.

Para habilitar a UART é necessário setar o bit UARTEN (UxMODE<15>) e o bit UTXEN (UxSTA<10>). Isso configura os pinos UxTX e UxRX como uma saída e uma entrada respectivamente, sem importar a configuração nos registradores TRIS e PORT/LATCH. Para desabilitar, deve – se zerar o bit UARTEN. Zerando o bit UARTEN enquanto a UART está ativa abortará todas as transmissões e recepções pendentes e resetará o módulo como definido anteriormente. Habilitando novamente a UART esta recomeçará na mesma configuração.

3 CONVERSOR A/D

O módulo conversor A/D do DSPIC16F4011 permite conversões de um sinal de entrada analógico a um número digital de 10 bits. Ele apresenta nove entradas analógicas, as quais são multiplexadas em quatro amplificadores de amostragem e manutenção (*sample and hold* – S/H). A saída do S/H é a entrada do conversor o qual gera o resultado.

Este módulo apresenta seis registradores de controle e status de 16 bits: ADCON1, ADCON2 e ADCON3, que controlam a operação do módulo A/D; ADCHS, seleciona os pinos de entrada a serem conectados aos amplificadores S/H; ADPCFG, que configura os pinos de entrada analógica como entradas analógicas ou como entrada/saída digital; e ADCSSL, que seleciona as entradas a serem sequencialmente escaneadas.

Para configurar o conversor A/D as etapas abaixo devem ser seguidas:

- Selecionar os pinos da porta como entradas analógicas, ADPCFG<15:0>;
- Selecionar a tensão de referência para acertar a faixa das entradas analógicas, ADCON2<15:13>;
- Selecionar o *clock* de conversão analógica para acertar a relação da data desejada com *clock* do processador, ADCON3<5:0>;
- Determinar quantos canais S/H serão usados, ADCON2<9:8> e ADPCFG<15:0>;
- Determinar quantas amostragens acontecerão, ADCON1<3> e ADCSSL <15:0>;
- Determinar quantas entradas serão alocadas para os canais S/H, ADCHS<15:0>;
- Selecionar a apropriada sequência amostragem/conversão, ADCON1<7:0> e ADCON3<12:8>;
- Selecionar quantos resultados de conversão serão apresentados no buffer, ADCON1<9:8>;
- Selecionar a relação de interrupção, ADCON2<5:9>;
- Ligar o módulo A/D ADCON1<15>.

Feita a configuração, é necessário agora configurar a interrupção A/D (se requerido): zerar o bit ADIF e selecionar a prioridade da interrupção. Logo após, se inicia

a amostragem, espera o tempo de aquisição requerido, finaliza o disparo da aquisição e começa a conversão. Terminada a conversão total, ou ocorrerá a interrupção ou setará o bit DONE. Em seguido o resultado é armazenado no BUFFER.

O conversor A/D tem uma relação máxima na qual as conversões serão completadas. Um *clock* do módulo analógico, TAD, controla a temporização da conversão. A conversão A/D requer de 12 períodos de *clock* (12 TAD). O *clock* A/D é derivado desde o *clock* de instrução (TCY) do dispositivo ou do *clock* RC interno. O período do *clock*

de conversão A/D é selecionado por software usando um contador de seis bits. Existem 64 possíveis opções para o TAD, especificado pelos bits ADCS<5:0> (ADCON3<5:0>). A equação (3) dá o valor a TAD como uma função dos bits de controle ADCS e do TCY.

$$TAD = \frac{Tcy(ADCS+1)}{2} \quad (3)$$

$$ADCS = \frac{2TAD}{Tcy} - 1 \quad (4)$$

4 CONCEPÇÃO DAS EXPERIÊNCIAS

Neste laboratório foram implementadas três experiências que utilizam os conceitos de comunicação serial e conversor A/D.

4.1 Experiência 1

A primeira experiência consistia em implementar um sistema de aquisição de dados, no qual o usuário selecionava o tempo de aquisição (a cada 0,5 seg, 1 seg, 10 seg, 1 minuto ou 1 hora) e o canal de entrada a ser varrido (pino RB7 ou RB8). O conversor A/D deveria ser configurado para realizar uma amostragem manual e 16 conversões a uma frequência de 8 kHz. O módulo UART deveria ter um *baud rate* de 19200.

Inicialmente, foi necessário calcular, através da equação (3), o TAD (*clock* de conversão) apropriado para a amostragem desejada. Com o Tcy igual a 62.5 ns e os bits de controle ADCS<5:0> (ADCS) setados com 16 Tcy, o TAD escolhido foi de 1 us, que é maior que o mínimo para o DSPIC30F4011 (153,85 ns).

O tempo de amostragem utilizado foi de 32 TADs, SAMC<4:0> (ADCON3<12:8>). Como o tempo de conversão requerido é de 12 TADs e o período de conversão total é de 125 us (inverso de 8 kHz), restam 81 us que são utilizados como atraso para o início de uma nova conversão.

Concluído os cálculos, restava configurar os registradores conforme solicitado.

- ADCON1 foi configurado com: ADC desligado, formato de saída inteiro e conversão automática;

- ADCON2 com: tensão de referência AVDD e AVSS, com varredura, interrupção após 16 amostras e buffer como palavra de 16 bits;
- ADCON3 com Tad de 16 Tcy;
- ADCSSL com varredura ativada para coletar RB7 e RB8.

Finalizada a configuração dos registradores, deu-se início ao algoritmo capaz de realizar as tarefas solicitadas. A ideia consiste basicamente em alternar entre dois diferentes menus: para tempo de aquisição e para canais de varredura. Em cada menu, as opções foram dispostas com letras distintas, de modo que o usuário precisa apenas observar qual letra corresponde a qual comando. Por exemplo, a letra “a” seleciona a aquisição a cada 0,5 segundos; enquanto o dígito 7 seleciona o canal 7.

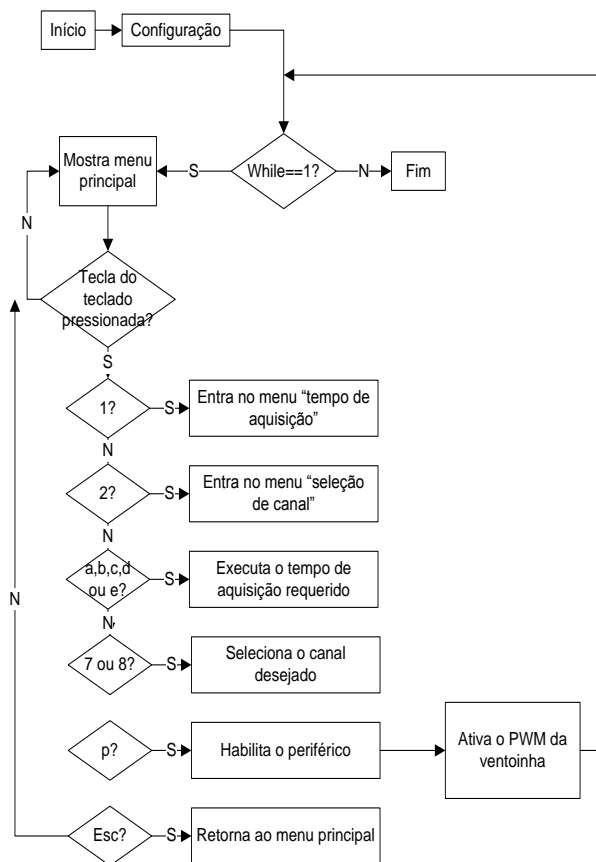


Figura 4.1 – Fluxograma da experiência 1.

Analisando o fluxograma na Figura 4.1, nota-se que o programa aguarda uma tecla do usuário e executa a conversão de acordo com os parâmetros selecionados, mostrando os resultados no PC e no LCD.

A disposição dos menus possui a seguinte ordem:

- Teclas 1 e 2: selecionam um dos dois menus (tempo de aquisição e amostragem individual);
- Teclas a, b, c, d ou e: dentro do menu 1, selecionam o tempo de aquisição;

- Teclas 7 ou 8: dentro do menu 2, selecionam o canal a ser exibido;
- Tecla “Esc”: retorna ao menu principal.

A tecla “P” foi adicionada como criatividade. Quando pressionada, ela habilitava o periférico ventoinha.

4.2 Experiência 2

A segunda experiência consistia em controlar o conversor A/D pelo PC. Caso o usuário pressionasse a tecla “A” (aquisição), uma conversão de uma das entradas dos potenciômetros seria iniciada. Caso se pressionasse a letra “P”, a conversão seria interrompida e a mensagem “A conversão parou, pressionar A para começar de novo” seria exibida no PC. A amostragem deveria ser configurada como manual e a conversor deveria fazer 16 conversões a uma frequência de 6 kHz. O conversor AD pegaria 16 amostras e faria a média destas. O baud rate para a comunicação serial deveria ser de 19200.

Analogamente à primeira experiência, foi calculado o Tad e o tempo de amostragem necessária para que a conversão resultasse num tempo total de 166,66 μs (aproximadamente 6 KHz). A partir da equação (1) foi escolhido novamente um Tad de 375 ns, isto é, o registrador ADCS foi setado com o valor de 11.

Como seriam 16 amostras, temos um tempo de conversão e amostragem de

$$\frac{166,66}{16} \cong 10,41 \mu s / amostra$$

Além disso,

$$\frac{10,41 \mu s}{0,375 \mu s} \cong 28$$

Ou seja, para um TAD de 0,375 μs, seriam necessários 28 Tad para se atingir o tempo de conversão total de uma amostra. Como 12 TAD são gastos com a conversão, sobriam 16 TAD para a amostragem. Dessa forma, devemos configurar os bits SAMC<4:0> (ADCON3<12:8>) com o valor de 16.

$$T_{amostragem} + T_{conversão} = 16TAD + 12TAD = 28TAD$$

$$28TAD = 28 * 0,375 = 10,5 \mu s / amostra$$

Para 16 amostras:

$$10,5 \frac{\mu s}{amostra} * 16 amostras = 168 \mu s \rightarrow 5,9 kHz$$

Configurando agora os registradores:

- ADCON1 foi configurado com: ADC desligado, formato de saída inteiro e conversão automática;

- ADCON2 com: tensão de referência AVDD e AVSS, sem varredura, interrupção após 16 amostras e buffer como palavra de 16 bits;
- ADCON3 com TAD de 6 Tcy e amostragem de 16 TAD;
- ADCHS: seleciona o canal CH0, configura entrada analógica AN8 (RB8);
- ADCSSL com varredura desativada.

Analisando agora o fluxograma da Figura 4.2, onde é mostrado as etapas do algoritmo da experiência 2, nota-se que o programa aguarda a entrada de um comando pelo usuário. Caso ele pressione A, a conversão inicia; caso pressione P, a conversão para.

Quando a conversão está ativa, o conversor A/D faz a amostragem e conversão de 16 amostras antes de gerar uma interrupção. Assim, após esta, o buffer já conterá todas as 16 amostras convertidas. Tais amostras serão passadas para a variável “valor” de modo acumulado, isto é, valor guardará a soma das 16 amostras. Após o fim do laço (i de 1 a 16), uma variável do tipo *float* calculará o valor médio e fará a escala correspondente aos 5 volts (multiplicando por 5 e dividindo por 1023).

O resultado da conversão de cada amostrada é mostrado continuamente no PC e o valor médio, após as 16 amostras, é mostrado tanto no PC como no LCD.

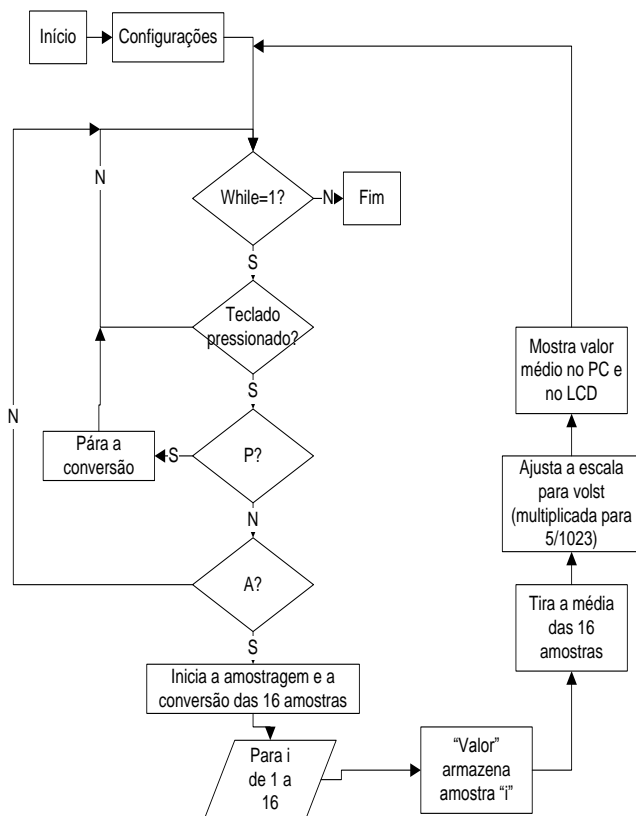


Figura 4.2 - Fluxograma da experiência 2.

Como criatividade, o usuário tinha a opção de escolher a escala da tensão, ou 5 ou 12. Pressionando a tecla “2” era mostrada, no PC e LCD, a média das 16 amostras na escala de 12. Pressionando a tecla “1”, retornava a escala de 5.

4.3 Experiência 3

A terceira experiência consistia utilizar os dois potenciômetros do kit para controlar a intensidade dos LEDs B<6:0> e da ventoinha, de modo que os *duty cycles* fossem correspondentes aos valores analógicos convertidos, isto é, 0 corresponderia a 0% e 1023 a 100%. Assim, se a conversão resultasse em um número (de 10 bits) x, o *duty cycle* seria:

$$Duty\ cycle = x * 100 / 1023 \quad (5)$$

Mais uma vez utilizando a equação (3) para o cálculo do TAD (*clock* de conversão) apropriado para a amostragem desejada. Com o Tcy igual a 62.5 ns e os bits de controle ADCON3<5:0> (ADCS) setados com 16 Tcy, o TAD escolhido foi de 1 us, que é maior que o mínimo para o DSPIC30F4011 (153,85 ns).

O tempo de amostragem utilizado foi de 32 TADs, SAMC<4:0> (ADCON3<12:8>). Como o tempo de conversão requerido é de 12 TADs e o período de conversão total é de 125 us (inverso de 8 kHz), restam 81 us que são utilizados como atraso para o início de uma nova conversão.

Como a varredura das entradas AN7 e AN8 foram escolhidas com 16 amostras antes da interrupção ADIF. Os registradores ficaram da seguinte forma:

- ADCON1 foi configurado com: ADC desligado, formato de saída inteiro e conversão automática;
- ADCON2 com: tensão de referência VDD e VSS, com varredura, interrupção após 16 amostras e buffer como palavra de 16 bits;
- ADCON3 com TAD de 16 Tcy e amostragem de 32 TADs;
- ADCSSL com varredura ativada para AN7 e AN8.

Finalizada a configuração dos registradores, deu-se início ao algoritmo para realização das tarefas designadas. O fluxograma da Figura 4.3 descreve as etapas seguintes:

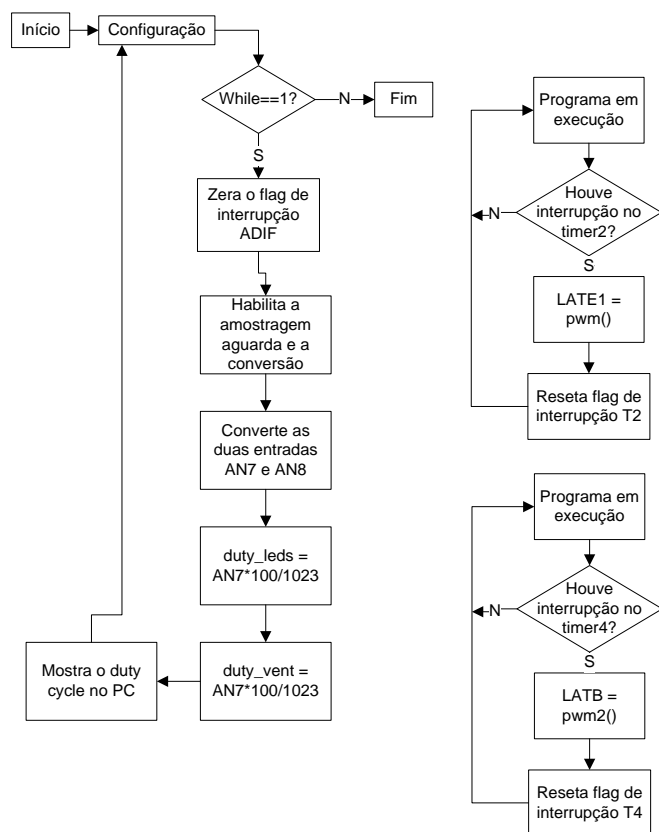


Figura 4.3 - Fluxograma da experiência 3.

O programa consiste basicamente em fazer a varredura dos dois canais (AN7 e AN8), configurados no registrador ADCSSL. Com os valores das entradas convertidos para digital, disponíveis a cada interrupção, bastaria fazer uma escala baseando-se na equação (5), enviando o resultado do AN7 para o *duty cycle* dos LEDs e o do AN8 para o *duty cycle* da ventoinha. Com os valores de DC, restava utilizar a função `pwm()` com um dos timers (um para cada periférico), método já empregado em laboratórios anteriores. Na Figura 3.3, nota-se que foram utilizados o timer 2 para a ventoinha e o timer 4 para os LEDs. A função `pwm()` retorna 1 para a ventoinha em ALTO e 0 pelo tempo programado em BAIXO. A `pwm2()`, idêntica à anterior, difere apenas no retorno, a saber, 127. Com este valor, acendem-se os sete leds RB<6:0>.

5 CONCLUSÃO

A utilização do módulo serial (UART) possibilita uma interação poderosa com o microcontrolador, permitindo o controle de diversos periféricos a partir do PC.

Embora a configuração dos registradores para a comunicação serial seja extensa, a criação da biblioteca UART.h (disponibilizada com os códigos das

experiências) facilitou a implementação, uma vez que poucos parâmetros precisam ser efetivamente modificados. Compreendendo o mecanismo de recepção e envio de caracteres e de posse da tabela ASCII, foi possível, então, realizar diversas tarefas com o microcontrolador de forma simples e prática. Assim, as experiências puderam ser realizadas de forma mais organizada, com criação de menus, saídas de leitura, além do controle via teclado do PC.

Assim como a UART, o módulo conversor A/D é uma importante ferramenta do microcontrolador. Ela permite a interação da máquina com o meio físico externo.

O módulo ADC do DSPIC30F4011, com seus 10 bits de resolução e a utilização da técnica de aproximações sucessivas, mostrou-se eficiente e com boa velocidade. A capacidade de varrer até 16 entradas, utilizar multiplexação de canais e definir tempo de amostragem, torna o módulo altamente ajustável aos interesses do projetista. No entanto, as possibilidades de configuração acabam requerendo bastante atenção aos vários registradores.

No mais, as experiências funcionaram conforme solicitado e este laboratório gerou motivação para outras aplicações, nas quais o microcontrolador poderá interagir com mais periféricos analógicos.

6 REFERÊNCIA BIBLIOGRÁFICA

[Hidalgo, 2013.1] Hidalgo, A. R. 2013.1, Apostilas da disciplina Microcontroladores da Universidade Federal de Sergipe ministrada pelo Prof. Dr. Antônio Ramirez Hidalgo.

[Souza, 2008] Souza, V. A. Programação em C para o DSPIC: Fundamentos – São Paulo: Ensino Profissional, 2008.