UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E COMPUTAÇÃO

EA871 - LAB. DE PROGRAMAÇÃO BÁSICA DE SISTEMAS DIGITAIS

EXPERIMENTO 9 – Conversor ADC

Profª Wu Shin-Ting
Aluno: Tiago Eidi Hatta RA: 160388

Proposta do Experimento

Controle de frequência de piscadas de leds através de um potenciômetro, ao utilizar um conversor de sinais analógicos para digitais do MCU da MKL25Z128, via módulo ADC.

Metodologia:

Dados podem ser representados de duas formas: analógica ou digital.

A informação analógica corresponde a uma onda eletromagnética gerada que pode assumir infinitos valores no tempo. Já na informação digital a representação de dados é representada por 1s e 0s.

A conversão de analógico para digital depende da amostragem, que consiste em colher amostras periódicas de um sinal analógico[8].

No caso do MKL25Z128, é integrado um conversor analógico-digital de 16 bits. A técnica implementada é a de aproximação sucessiva com um registrador de aproximação sucessiva de até 16 bits. Os sinais de entrada VIN são amostrados e segurados para serem comparados com os sinais digitais aproximados e convertidos em analógicos pelo circuito DAC. E o comparador realimenta o circuito do registrador SAR com a diferença dos dois sinais e atualiza o conteúdo do SAR com esta diferença. E assim, sucessivamente, até completar todos os bits do SAR e gerar o resultado EOC (end of conversion).

Para este experimento, foi utilizado um potenciômetro, componente eletrônico que possui resistência ajustável [9]. A partir dele, a tensão no pino PTB1 da placa auxiliar pôde ser variada. Esse valor, que é analógico, foi convertido para digital pelo módulo ADC.

Proposta de Solução

Módulos e Registradores Configurados [1]

Módulo SIM:
- SIM_SCGC5 e SIM_SCGC6: responsáveis por habilitar o clock para cada uma das portas relacionadas aos pinos da placa utilizados no experimento.
- SIM_SCGC4 (System Clock Gating Control Register 4), habilita o clock para o módulo UART0.
- SIM_SOPT2 (System Options Register 2), configura a fonte de relógio.

Módulo PORT: Responsável por dar suporte ao controle das portas da placa principal do kit usado. Registradores:

- PCRn: de acordo com a porta utilizada do pino n, configura a multiplexação para GPIO e o modo de disparo de interrupção de acordo com o tipo de borda do sinal.
Foram configuradas:
- Porta A, pinos 4, 5 e 12: push-buttons da placa auxiliar.
- Porta B, pino 18, led vermelho, MUX = 3 (TPM2_CH0)
- Porta B, pino 19: led verde, MUX = 3 (TPM2_CH1)
- Porta D, pino 1: led azul, MUX = 3 (TPM0_CH1)
- Porta C, pinos 0 a 7: Dados GPIO
Os pinos do LCD são ligados ao da porta C da placa.
- Porta C, pinos 8 a 10: RS, E do LCD e LE do latch, respectivamente.

- ISFR: o bit setado corresponde ao pino que detecta (valor lógico 1) ou não (valor lógico 0) uma interrupção configurada.
Foram configuradas:
- Porta A, pinos 4, 5 e 12.

Systick: Timer com contador de 24 bits que permite interrupções periódicas
Registradores configurados:
SYST_CSR: Controla o timer do sistema (habilita ou desabilita o clock do systick) e fornece status do mesmo.
SYST_RVR: Deve ser inicializado com o número do contador a ser reiniciado sempre que o contador chega a zero.
SYST_CVR: Responsável por ler ou limpar o valor do contador.

Módulo GPIO: Comunicação digital do processador do MCU. Registradores:
- PDDR: configura os pinos como saída (lcd) ou entrada (push-buttons) para GPIO.
- PDOR: configura os pinos como saída (lcd) ou entrada (push-buttons) para propósitos gerais.

NVIC (Nested Vectored Interrupt Controller): Controlador de Interrupções
NVIC_ISER: habilita as interrupções de acordo com os pinos configurados
NVIC_ICPR: remove estados de interrupção pendentes
NVIC_IPR4: configura nível de prioridade
NVIC_IPR3: configura nível de prioridade

Módulo UARTx (Universal Asynchornous Receiver/Transmitter): contém um circuito de transmissão e um de recepção. Através destes circuitos os bytes a serem transmitidos são serializados e os bits recebidos são reagrupados em bytes automaticamente, a uma frequência derivada da fonte de clock selecionada.

UART0_C1: Configura diversos recursos de UART0. Deve ser configurado com transmissor e receptor desabilitados.
UART0_C2: Habilita e desabilita transmissor e receptor de UART0.
UART0_C4: Configura a taxa de amostragem, setando o valor de OSR (Over Sampling Ratio).
UART0_C5: Configura o Both Edge Sampling, que permite receber dados em ambas bordas de subida e descida do clock.
UART0_BDH: juntamente com UART BDL, controla o divisor para a taxa de transmissão UART (UART baud rate).
UART0_BDL: similar a UART0_BDH.
UART0_S1: Configura o transmissor.

Módulo TPM: Responsável por configurar cada pino segundo uma determinada função: Input Compare, Output Compare e PWM.

TPMx_SC: indica status de overflow, através de uma flag e controla os bits usados para configurar o enable de interrupções, a configuração do módulo e o fator pré-escala.
TPMx_MOD: contém o valor do módulo do contador do LPTPM. Quando o contador alcança o valor de modulo e incrementa, a flag overflow é setada.
TPMx_CNT: contém o valor do contador LPTPM. Reset limpa o registrador do contador. Escrevendo qualquer valor nos 15 bits menos significativos também limpa o contador.
TPMx_CnSC: contém a flag de estado/interrupção do canal correspodente e controla os bits usados para configurar a habilitação de interrupção, configuração geral do canal e também a função do pino (IC, OC, PWM).
TPMx_CnV: contém o valor capturado do contador LPTPM para modo input ou um valor de comparação para modo output.

Módulo ADC: Responsável pela conversão de sinais analógicos em digitais.

ADC0_CFG1: seleciona o modo de operação, fonte de relógio, divisor de clock e configura ADC para baixo consumo de energia e o período de amostragem (curto ou longo).
ADC0_CFG2: seleciona as configurações especiais para altas velocidades de conversões e seçecopma a duração do período longo de amostragem.
ADC0_SC2: mostra se há uma conversão acontecendo, além de selecionar o tipo de trigger (hardware/software), habilitar a função de comparação e selecionar a tensão de referência para o módulo ADC.
ADC0_SC3: controla a calibração, a conversão contínua, a habilitação e configuração da função de média de amostragem pelo hardware.
ADC0_SC1A: controla o modo de operação (uni ou bipolar) nos canais do conversor do módulo ADC.

Abaixo estão os pseudo-códigos das duas partes do experimento, bem como a explicação da solução baseada nos códigos fontes em C.

Pseudo-Código

```
// Funcao main
MAIN:
INÍCIO
        CONFIGURA PINOS (GPIO, INICIA LCD, UART, TPM, NVIC)

        INICIALIZA SYSTICK
        CONFIGURA ADC
        CALIBRA ADC

        CRIA CARACTERE ê (chama criaCaractere)

        ALOCA BUFFER TRANSMISSOR
        ALOCA BUFFER RECEPTOR

        PARA LOOP VERDADEIRO
                Inicio:                 // rotulo de inicio

                SE estado_pta5 OU estado_pta12 OU estado_terminal OU estado_pot iguais a 1
                        SE estado_pta5 ou estado_terminal igual a 1
                                Seleciona a cor de acordo com o tempo apertado da botoeira ou do
        lido pelo terminal
```

FIM_SE
SE estado_pta12 ou estado_terminal igual a 1
Seleciona frequencia de acordo com aperto da botoeira ou do lido no terminal
SE estado_terminal = 1
Manipula caracteres para selecionar cor ou frequência segundo entrada do terminal

CONVERTE PARA ASCII O VALOR FLOAT DA FREQUENCIA DE PISCADA

FIM_SE

SE estado_pot igual a 1
// houve mudanca no potenciometro, usuario alterou frequencia
Seleciona frequencia de acordo com o potenciômetro

MANDA STRING PARA O VISOR: “Cor: “
MANDA STRING PARA O VISOR: NOME DA COR SELECIONADA DO LED
MANDA STRING PARA O VISOR: “Frequ”
MANDA STRING PARA O VISOR: “ê”
// (criado anteriomente e armazenado em CGRAM)
MANDA STRING PARA O VISOR: “ncia: ”
MANDA STRING PARA O VISOR: FREQUENCIA CONVERTIDA EM ASCII

MANDA STRING PARA O TERMINAL: “Cor: ”
MANDA STRING PARA O TERMINAL: NOME DA COR SELECIONADA DO LED
MANDA STRING PARA O TERMINAL: “Frequência: ”
MANDA STRING PARA O TERMINAL: FREQUENCIA CONVERTIDA EM ASCII
FIM_SE

SE estado_pta5 e estado_pta12 e estado_terminal e estado_pot igual a 0
// botoeiras soltas e sem mensagem recebida no terminal

ACENDE LED DE ACORDO COM O VALOR DE INDICE ATRAVÉS DE MODULO TPM

TEMPO: INVERSO DA FREQUENCIA SELECIONADA POR NIVEL

SE estado_pta5 e estado_pta12 e estado_terminal iguais a 1
// aperto de botoeira ou mensagem no terminal detectados
// delay é função demorada, por isso a verificação
VÁ PARA inicio
// retorna para rotulo inicio

DELAY10US passado como parametro TEMPO determinado de delay

APAGA LED

SE estado_pta5 e estado_pta12 e estado_terminal iguais a 1
// aperto de botoeira ou mensagem no terminal detectados
// delay é função demorada, por isso a verificação
VÁ PARA inicio
// retorna para rotulo inicio

DELAY10US passado como parametro TEMPO determinado de delay

FIM_SE

FIM PARA
FIM

Código Fonte
Enviado como anexo ao relatório pelo Ensino Aberto. Documentado de acordo com o especificado pela Doxygen.

Testes

Para os testes, foram realizadas as seguintes etapas:

1) Teste da leitura do MCU com o sensor de temperatura integrado ao MKL25Z128 e conversão pelo módulo ADC;
2) Teste da leitura dos valores pela variação do potenciômetro e conversão do sinal analógico em digital pelo módulo ADC, usando o recurso *debug* do Code Warrior;
3) Após a integração do programa com outros módulos com inclusão da interface de controle dos leds, botoeiras, lcd e terminal, houve o teste com a mudança da frequência de piscadas pela variação do potenciômetro;
4) Teste de todas as outras funções do programa, variar cor e frequência dos leds pelas botoeiras da placa auxiliar e pelo terminal, bem como a mudança instantânea do texto exibido no lcd e no terminal para qualquer alteração realizada.

Conclusão

A maioria dos sensores produzem sinais analógicos. Pode-se observar isso, ao se programar o MCU para valores do sensor de temperatura acoplado ao MKL25Z128 (Cap.28.4.8 em [2]), bem como neste experimento, ao usar um potenciômetro.

Para a resolução do problema, percebeu-se que o periférico é passivo. Ou seja, o potenciômetro mede os valores continuamente sem “avisar” quando iniciou uma nova amostragem. O único aviso possível é no momento em que foi concluida uma amostragem, após o programa dar o gatilho de "iniciar uma amostragem". Para resolver este problema, foi utilizado o timer Systick, que permite que o núcleo gere interrupções periódicas independentes do controle de NVIC. Portanto, a cada 0.25s, o programa iniciava uma amostragem e a partir disso, poderia modificar a frequência de piscadas dos leds, caso houvesse alteração pelo potenciômetro, ao ler o registrador ADCx_RA.

Para este experimento, uma observação é a de que a frequência de piscadas foi proporcionalmente linear aos valores controlados pelo potenciômetro. Assim, para valores próximos como 0.5Hz e 1.0Hz, o usuário precisava girar bem pouco o eixo do dispositivo em comparação aos valores entre 50Hz e 60Hz, por exemplo.

Além disso, outra constatação é a de que os leds tem as cores propostas pelo sistema RGB, diferentemente do desejado. Isso ocorre pois os três leds estão ligados em

módulos distintos (TPM0 (B) e TPM2 (R e G)). Como os dois módulos tem contadores diferentes, sem uma sincronização dos seus sinais, o B pode estar fora de fase de R e G. Isso gera efeitos visuais distintos do esperado.

Referências

[1] FRDM-KL25Z User's Manual.

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/FRDMKL25Z.pdf

[2] KL25 Sub-Family Reference Manual

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KL25P80M48SF0RM.pdf

[3] ARMv6-M Architecture Reference Manual – ARM Limited.

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/ARMv6-M.pdf

[4] EA871 – Descrição do Hardware da Placa Auxiliar.

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/DescricaoDoHardware.pdf

[5] LCD Tutorial for Interfacing with Microcontrollers. Rickey's World.

http://www.8051projects.net/lcd-interfacing/index.php

[6] Datasheet do display LCD.

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea079/datasheet/AC162A.pdf

[7] EA871 - Descrição do Hardware da Placa Auxiliar

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/DescricaoDoHardware.pdf

[8] Analógico e Digital

http://br.ccm.net/faq/2962-analogico-e-digital

[9] Potenciômetro

https://pt.wikipedia.org/wiki/Potenci%C3%B4metro