EXPERIMENTO 9 - ADC e LPTMR - Implementação

FEEC | EA871

Thiago Maximo Pavão - 247381
Vinícius Esperança Mantovani - 247395

Cálculo do prescaler para TPM

Sabe-se que o período do TPM2 é dada por

$$Periodo = TPMx\_MOD \times 2^{TPMx\_SC\_PS} \times (1 + TPM2\_SC\_CPWMS)/f_{clock}$$

Optamos por contagem progressiva, pois não encontramos a especificação no roteiro, temos então

$$645 \times 309,22us = 0,1994469s = 65535 \times 2^{TPMx\_SC\_PS} \times (1 + 0)/20971520$$

Resolvendo para TPMx_SC_PS, temos $TPMx\_SC\_PS = 5,9963$, o valor deve ser inteiro portanto escolhe-se 6. Portanto a divisão feita é por 64, no prescaler.

Testes de unidade

Com a inicialização, foram feitos os seguintes testes de unidade:

LCD: Foi inserido GPIO_escreveStringLCD(0x0, "Teste"); logo antes do loop infinito for(;;). Executando foi possível ver a palavra escrita no LCD.

Cooler: Inserindo TPM_CH_config_especifica(1, 0, 0b1010, 2000); para ligar um o PWM no PTB0 com aproximadamente 50% de razão de trabalho, vemos que o cooler girou.

LED: Inserindo TPM_CH_config_especifica(2, 0, 0b1010, 32000); vemos que o canal vermelho é ativado, é possível vê lo piscando, já que foi escolhido um pulso de aproximadamente 50% de ciclo de trabalho. Posteriormente, inserindo TPM_CH_config_especifica(2, 1, 0b1010, 32000); nota-se que o LED começa a piscar na cor verde.

Conversão potênciometro: Para aproveitar o função *ftoa*, fizemos o seguinte código, que lê o valor convertido e traz para o intervalo de tensão de 0 a 3,3V. Este valor é então impresso na tela. Anteriormente no código foi selecionado o canal do ADC: 0b01001, para realizarmos medidas no potênciometro.

```
char buffer[15];

for(;;) {
    ftoa(3.3*ADC0_RA/65535, buffer, 2);
    GPIO_escreveStringLCD(0x0, (uint8_t *) buffer);
}
```

Conversão sensor de temperatura: Utilizando a mesma lógica no loop anterior, com cinco casas decimais, foi possível ver que a tensão do sensor sempre se manteve próxima de 0,72V, que indica

uma temperatura de cerca de 25 graus celsius. Ligando o cooler, e direcionando o fluxo de ar na direção da placa controladora foi possível ver que tensão aumenta levemente, o que se encaixa com o esperado.

Função de conversão de tensão para temperatura: Foi utilizado o código, foi possível ver que o valor da temperatura varia ao redor de 24,99 graus celsius. Fazendo novamente o teste com o cooler é possível perceber que a temperatura varia quando o fluxo de ar é colocado na direção da placa.

```
char buffer[15];

for(;;) {
    ftoa(AN3031_Celsius(ADC0_RA), buffer, 5);
    GPIO_escreveStringLCD(0x0, (uint8_t *) buffer);
}
```

Transições de estados

Inicialmente, o estado é AMOSTRA_VOLT, pois não há dados a serem mostrados até que a primeira amostragem dos sensores seja completa. Como foi ativada o número de interrupção de ADC0 no NVIC, foi selecionado trigger por hardware, e o trigger selecionado é o de overflow de TPM02, no momento em que ocorre o primeiro overflow a conversão deve ser feita, isto é feito automaticamente pelo hardware, devido a forma em que os dispositivos foram configurados.

Após a conclusão da conversão, ADC0 gera uma interrupção que será tratada pela rotina de tratamento de interrupção. Nela será feita uma verificação do estado, e no caso de AMOSTRA_VOLT o valor será salvo de acordo em um vetor estático presente no arquivo ISR.c, após isso, a forma de trigger deve ser alterada para por software, para que ela seja iniciada em seguida para realizar a conversão da leitura do sensor de temperatura AN3031. É feita a alteração na forma de trigger e então a seleção do canal do sensor, causando o início da conversão. Por fim o estado é alterado para AMOSTRA_TEMP.

Quando esta segunda conversão é concluída, outra interrupção é gerada e utilizando o estado é possível distinguir que o valor lido deve ser salvo em outra posição do vetor, pois é a leitura de outro canal de ADC0. Após isso, é necessário voltar ao estado para preparar a próxima conversão, retornando para trigger por hardware e selecionando o canal do potenciômetro. Por fim o estado é alterado para ATUALIZACAO.

Neste estado, na *main* é feita a atualização dos periféricos de saída: LED, LCD e Cooler. O estado do LED é configurado via os canais 0 e 1 do TPM2, de acordo com a temperatura lida. O LCD recebe os valores do Duty Cycle e da temperatura, e o cooler recebe o valor lido do potenciômetro diretamente para definir a largura de pulso. Por fim, o estado é alterado para AMOSTRA_VOLT, reiniciando o ciclo da máquina de estados.

Um problema

Ao terminar de implementar a máquina de estados resolvemos alguns problemas que surgiram mas um deles persistiu: Ao colocar o potenciômetro no mínimo percebeu-se que o cooler começava a rodar em sua velocidade máxima. Isto foi resolvido alterando a forma que a largura de pulso do cooler, ao invés de utilizar a função TPM_CH_config_especifica em todas as reconfiguraçôes, como se deseja alterar apenas campo Value do canal foi criada a função TPM_CH_set_V, que altera apenas o campo desejado. O funcionamento dos outros canais (LED) foi mantida.