UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Relatório do Lab 4

Disciplina de Sistemas Embarcados – Prof. Douglas Renaux

Autores: Luis Camilo Jussiani Moreira e João Victor Laskoski

Versão: 25-Sep-2022

1 Introdução

A prática em questão refere-se ao laboratório 4, sendo executado na presente disciplina. Tal laboratório tem o intuito de interagir com diversos periféricos (do processador, da placa Tiva e da placa BoosterPack).

Sendo assim, o atual laboratório tem como objetivo o uso do Joystick, em que movimentando-o, atualiza a cor do LED RGB, como também o uso do sensor de luminosidade.

Portanto, o atual laboratório tem como propósito de aprendizado a utilização de tais periféricos, como LED RGB, Joystick, além de comunicações (I2C e UART) e conversor A/D para confecção do projeto.

2 Planejamento das fases do processo de desenvolvimento

Sendo assim, para tal prática, será dividido nas seguintes etapas para a resolução desse projeto.

Etapa 1:

- Funcionamento do Joystick
- Utilização do Joystick em conjunto com a biblioteca TivaWare
- Funcionamento do LED RGB
- Utilização do LED RGB em conjunto com a biblioteca TivaWare

Etapa 2:

- Uso do conversor A/D, utilizando as funções da biblioteca TivaWare
- Uso de temporização (Timer0 ou SysTick), através da biblioteca TivaWare
- Uso da comunicação UART, como também das funções da biblioteca TivaWare que lidam com essa comunicação
- Uso de PWM através de SysTick, Timer ou da própria API de PWM da TivaWare

Etapa 3:

- ~~Funcionamento do sensor de luminosidade~~
- ~~Utilização do sensor de luminosidade em conjunto com a biblioteca TivaWare~~
- ~~Uso da comunicação I2C, como também das funções da biblioteca TivaWare que lidam com essa comunicação~~

3 Definição do problema a ser resolvido

Sendo assim, para o projeto em questão, deve-se manipular o Joystick e de acordo com sua posição, alterar o valor do led RGB, mapeando a posição atual do Joystick para uma cor em RGB. Além disso, periodicamente deve-se enviar a posição do Joystick para o PC via comunicação UART.

4 Especificação da solução

O diagrama de blocos da Figura 1 ilustra a solução projetada para o laboratório. No diagrama, as linhas contínuas representam ligações entre componentes de diferentes dispositivos (ligação Tiva-PC e ligação Tiva-BoosterPack), enquanto que a ligação em tracejado (Tiva-Timer) indica que o Timer está presente dentro da TM4C1294XL, ou seja, é um periférico interno ao microcontrolador.

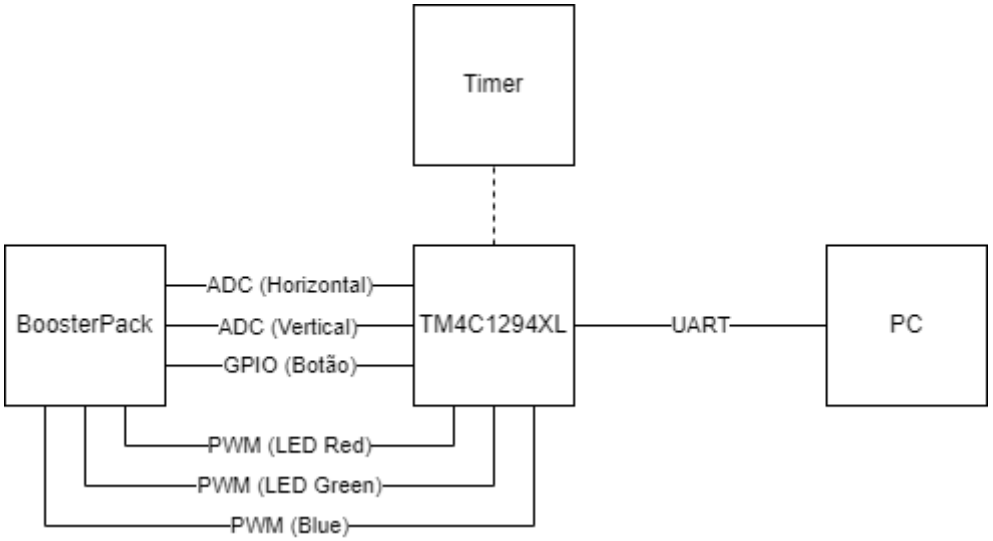

Figura 1: Diagrama de blocos da solução.

Conforme ilustra o diagrama, a ideia é obter os valores do Joystick (horizontal e vertical) por intermédio de um conversor analógico-digital, tornando assim possível o processamento dos dados no microcontrolador. Além disso, o botão do joystick foi configurado programando como um pino de GPIO.

Assim, temos a informação necessária de entrada do sistema (joystick) para o processamento e geração das saídas (informar o valor via UART para o pc e para o LED RGB da BoosterPack). A Figura 2 ilustra o fluxo de dados entre os componentes.

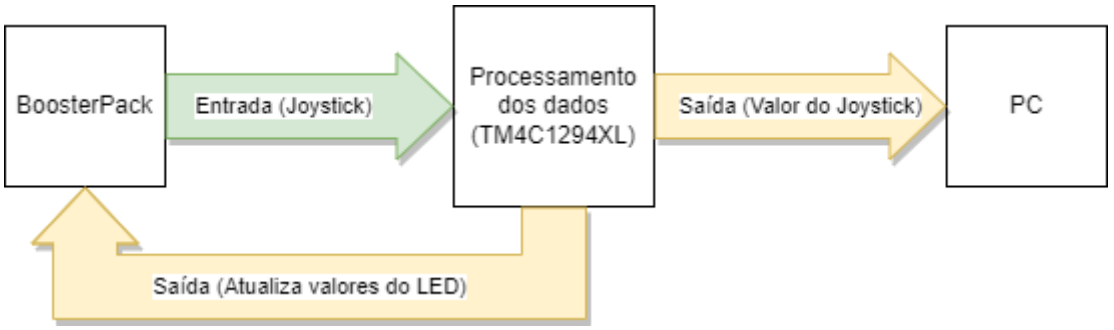

Figura 2: Fluxo de dados entre os dispositivos.

5 Estudo da plataforma de HW

Para a implementação da solução planejada, diversos módulos devem ser estudados, entre eles: ~~Timer, Interrupção, Conversor ADC,~~ PWM ~~UART~~ e configuração de GPIO.

5.1 Considerações a respeito do PWM

A TivaWare™ possui funções para programar PWM, sendo assim possível modificar a intensidade de brilho do LED da BoosterPack ao programar 3 PWMs distintos. Assim, a fim de alterar corretamente os valores nos pinos, a Figura 3 deve ser consultada.

Figura 3: Relação entre os pinos da TM4C129XL e o LED da BoosterPack.

5.5 Configurações a respeito do uso de GPIO

Para o acesso ao botão presente no Joystick e ao pino relacionado ao LED azul foi necessário a programação de GPIO, uma vez que são sinais digitais de entrada e saída (Leitura do botão do Joystick, onde 0 representa o botão pressionado e a ativação da cor azul ao LED, sendo 1 ativo e 0 desligado). A Figura 4 representa a relação entre os periféricos citados e os pinos.

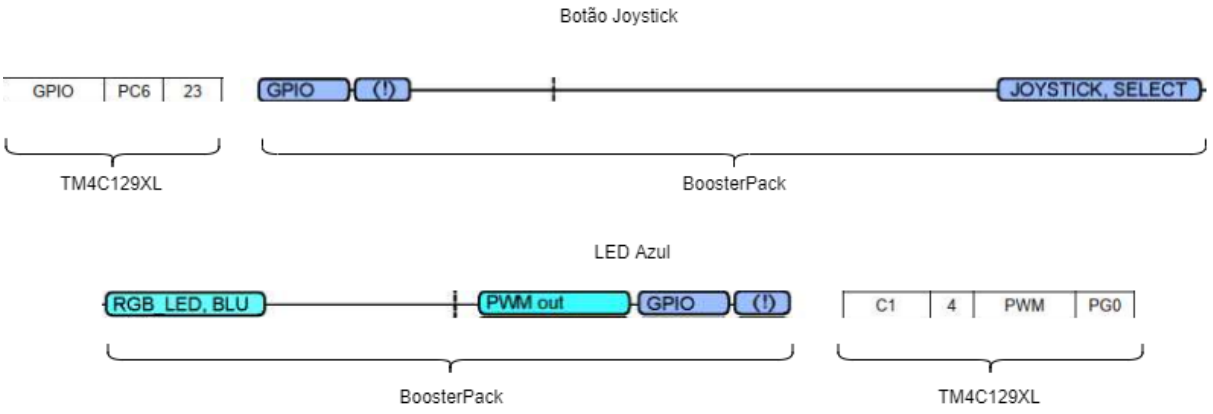

Figura 4: Relação entre os pinos da TM4C129XL com o LED azul e o botão da BoosterPack.

6 Estudo da plataforma de SW

Para a implementação da solução planejada, diversos módulos devem ser estudados para serem programados corretamente, entre eles: Timer, Interrupção, Conversor ADC e UART.

6.1 Considerações a respeito do Timer

Como o TM4C1294XL possui dois timers como periféricos, é possível assim implementá-los. A ideia do Timer é realizar uma contagem de 200 ms, podendo assim disparar uma interrupção ao término da contagem e reiniciá-la. Portanto, no estudo da documentação da Tiva essas questões foram levadas em consideração.

Sendo assim, as função da TivaWare™ úteis na solução são:

- TimerConfigure();
- TimerLoadSet();
- TimerControlTrigger();
- TimerEnable();
- IntEnable();
- TimerIntEnable();

6.2 Considerações a respeito da Interrupção e do Conversor ADC

A interrupção trabalha em conjunto com o timer, portanto é necessário ter conhecimento a respeito da ligação entre eles.

Ademais, um dado do conversor ADC só será lido após o disparo de interrupção provocado pelo término da contagem do timer. Assim, ao consultar a documentação da TivaWare™ é possível verificar que a função que relaciona ao ADC o término de timer como gatilho é TimerControlTrigger(). Além disso, para a solução do problema projeta-se o uso das seguintes funções:

- GPIOPinTypeADC();
- ADCSequenceConfigure():
- ADCSequenceStepConfigure();
- ADCSequenceEnable();
- ADCIntEnable();
- IntEnable();
- IntMasterEnable();
- IntPrioritySet();
- ADCIntClear();
- ADCSequenceDataGet();

6.3 Considerações a respeito da comunicação UART

A comunicação UART será feita para conectar dois dispositivos de hardwares distintos (TM4C129XL e PC). Como foi exigido a taxa de transmissão de 115.200 bps e o padrão de

transmissão 8N1, a pesquisa na documentação é necessária para a correta configuração da comunicação, resultando assim na seleção de funções descritas abaixo.

- GPIOPinTypeUART();
- UARTConfigSetExpClk();

  Nessa função é configurada o tamanho da palavra, stop bit e bit de paridade, podendo assim configurar conforme o padrão 8N1.

- UARTEnable();
- UARTCharPut();

7 Projeto (design) da solução

Conforme as Figuras 1 e 2 ilustram, todo o processamento de software é realizado na TM4C129XL, ou seja todo o software projetado será implementado no microprocessador. O diagrama da Figura 2 expõe que as entradas que alimentarão o software partem da BoosterPack, sendo assim processadas e com isso serão calculadas as saídas que alimentarão a boosterpack (Cor do LED) e o PC (valores do Joystick via UART). A seção atual tem por objetivo expor o processamento dos dados na TM4C1294XL. O diagrama de estados da Figura 5 ilustra de forma superficial o comportamento da solução.

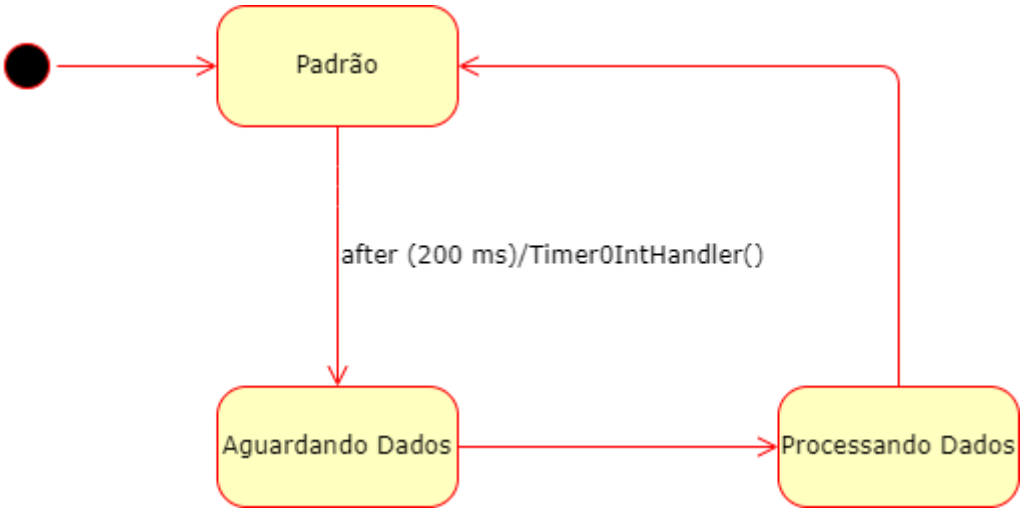

Figura 5: Diagrama de estados da solução.

7.1 Estado Padrão

O estado padrão corresponde ao período de tempo em que todos os dados de saída já foram processados, esperando assim a próxima interrupção para a leitura de novos dados. Pela transição do diagrama, é possível perceber que as interrupções ocorrem após 200 ms no estado Padrão (a ideia é ter interrupções periódicas a cada 200 ms, cumprindo assim o requisito do laboratório).

7.2 Estado “Aguardando Dados”

O estado corresponde ao período de "coleta" de sinais dos periféricos (Valores do Joystick - valores horizontal, vertical e botão). Sendo assim, o estado contém uma atividade relacionada a leitura de dados dos periféricos:

- Adc0SS2IntHandler (void): Rotina de tratamento de interrupção relacionada ao conversor AD ligado ao Joystick. Nela, os valores relacionados à posição horizontal e vertical, além do botão do Joystick, são coletados.

Conforme o diagrama, nenhuma transição está descrita, portanto ao final da função Adc0SS2IntHandler (void) a transição para o estado "Processar Dados" ocorre.

7.3 Estado "Processando Dados"

O estado de processamento de dados é composto por funções de conversão e escrita de dados nos periféricos. Sendo assim, o estado de processamento contém as seguintes funções (atividades):

- convertADOutputToPwmPulseWidth(): Converte o valor do conversor para um valor válido de PWM.
- PWMPulseWidthSet(): Relacionada a atualização do valor de PWM do LED.
- GPIOPinWrite(): Relacionada com a atualização do valor relativo a cor azul do LED.
- informDataUART(): Transmite ao PC os valores lidos do Joystick.
-

Assim, ao término das atividades descritas pelas funções acima, ocorre a transição para o estado padrão, retornando o ciclo de execução.

8 Configuração do projeto na IDE (IAR).

As configurações consistem em:

- Apontar para o arquivo de configuração do linker Tiva.icf
- Apontar que a pasta List/Obj/Exe para pasta ewarm
- Apontar para que o programa de inicialização seja o startewarm_up.c
- Configurar para que o projeto seja compilado em C
- Configurar o microcontrolador utilizado pela IDE
- Demais configurações utilizadas de labs anteriores

9 Teste e depuração.

Os testes foram particionados de acordo com as etapas elaboradas no projeto, em que a cada etapa completada, era feito um teste isolado. Sendo assim, os testes consistiram em:

- Ao ter implementado a conversão A/D, primeiramente foi configurado apenas para converter o pino relacionado ao joystick na vertical e confirmado o funcionamento (modificação do valor digital de acordo com a movimentação do joystick), posteriormente fez o mesmo teste, configurando o pino apenas para horizontal. Por fim, foi configurado para receber as duas amostas e verificado a modificação dos valores digitais (horizontal e vertical) enquanto era movimentado o joystick.

- Ao ter implementado o PWM e relacionado com os pinos do LED RGB da placa BoosterPack, foi configurado diferentes *pulse width* e alternando os pinos do LED ligados, verificando a alternância de cores.
- Ao ter implementado o PWM para se relacionar com o LED RGB, foi implementado para, ao receber o valor digital da conversão A/D, atualizar o *pulse width* do PWM. Assim, o teste consistiu em verificar a modificação da cor do LED ao movimentar o joystick.
- Configurado a UART, o teste inicial consistiu em enviar apenas um único char e verificar o recebimento através do PuTTY. Posteriormente, foi testado enviando um texto e novamente verificado o recebimento no PuTTY. Por fim, o teste foi o envio das informações dos valores dos conversores A/D por UART e verificar o que foi recebido do UART com o valor verificado via print no terminal I/O da IDE.