

Lab 2 – interagindo com a placa Tiva

Objetivo:

a partir da biblioteca TivaWare, desenvolver uma aplicação embarcada que faz uso de interrupções e interage com dispositivos de I/O (botões e leds).

1. Obtenha o TivaWare versão 2.2.0.295.

Pode ser obtido do site da Texas, acessível a partir dos links que disponibilizei.

Observe que há versões específicas para cada placa/processador. Os próximos 2 slides apresentam alguns dos passos para tal.

- Baixando o TivaWare de <https://www.ti.com/tool/SW-TM4C>



Download options

SOFTWARE DEVELOPMENT KIT (SDK)

SW-TM4C

TivaWare for C Series Software (Complete)

Version: 2.2.0.295

Release date: 23 Apr 2020

FULL RELEASE


 [SW-TM4C-2.2.0.295.exe](#) (145879 K)

KIT SOFTWARE

 [SW-EK-TM4C123GXL-2.2.0.295.exe](#) (29855 K)

 [SW-EK-TM4C1294XL-2.2.0.295.exe](#) (76364 K)

 [SW-EK-TM4C129EXL-2.2.0.295.exe](#) (75111 K)

 [SW-DK-TM4C129X-2.2.0.295.exe](#) (85643 K)

LIBRARIES

 [SW-TM4C-DRL-2.2.0.295.exe](#) (9818 K)

 [SW-TM4C-GRL-2.2.0.295.exe](#) (7727 K)

 [SW-TM4C-SENSORLIB-2.2.0.295.exe](#) (1842 K)

 [SW-TM4C-USBL-2.2.0.295.exe](#) (3882 K)

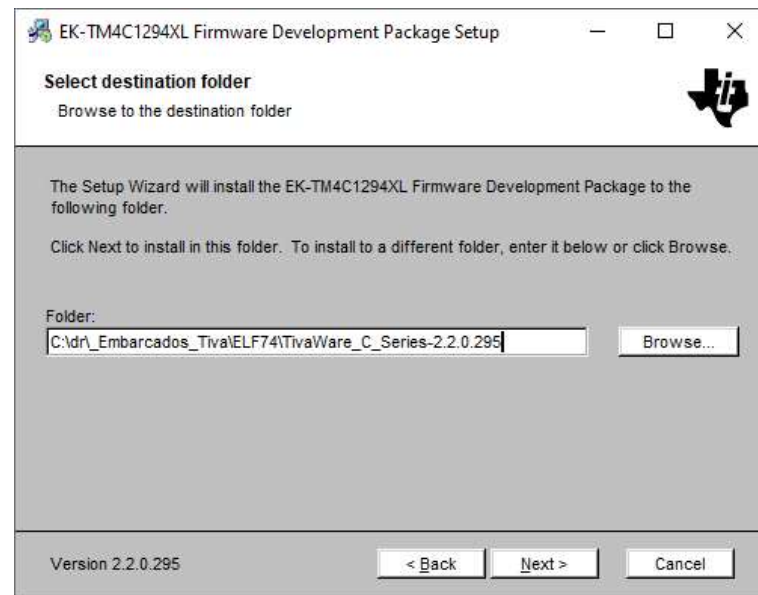
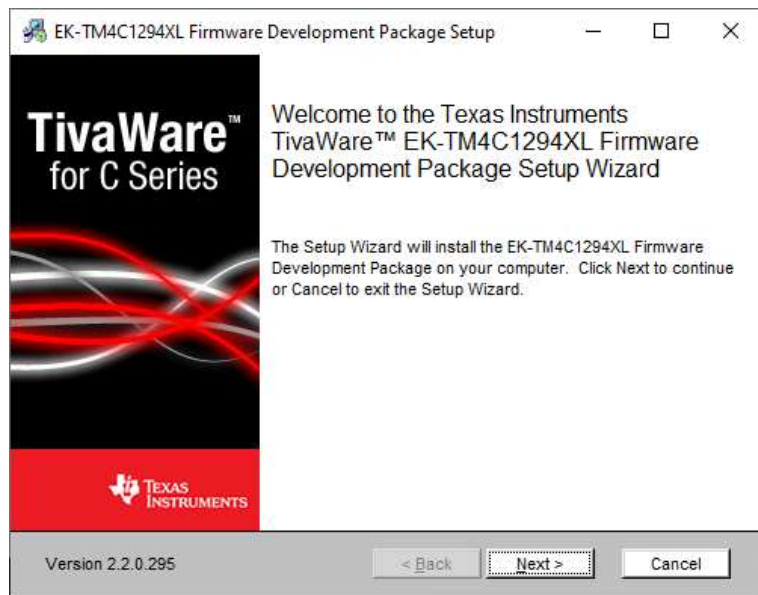
RELATED UTILITIES

 [SW-USB-win-2.2.0.295.msi](#) (2243 K)

 = Requires export approval (1 minute)

[Release notes](#) | [Previous versions](#) | [Documentation](#) | [Supported products & hardware](#)

- Autorizações e declarações de exportação
 - Peço não colocarem este código num site público
- Instalação:



Lab 2 – interagindo com a placa Tiva

2. Testando o TivaWare e a conexão com a placa Tiva

Abra o workspace (arquivo com extensão .eww) na pasta
TivaWare_C_Series-2.2.0.295\examples\boards\ek-tm4c1294xl

Compile a biblioteca driverlib.

Compile o exemplo blinky.



Execute o exemplo blinky na placa Tiva. Um dos LEDs deve piscar.

Estude a configuração de projeto no IAR de uma biblioteca (examine o driverlib)

Estude a configuração de projeto no IAR de um executável (examine o blinky)

Estude os dois arquivos fonte (startup_ewarm.c e blinky.c) que compõem o projeto blinky.

A pasta no seu computador onde estão os projetos de lab para esta disciplina deve estar organizada assim:

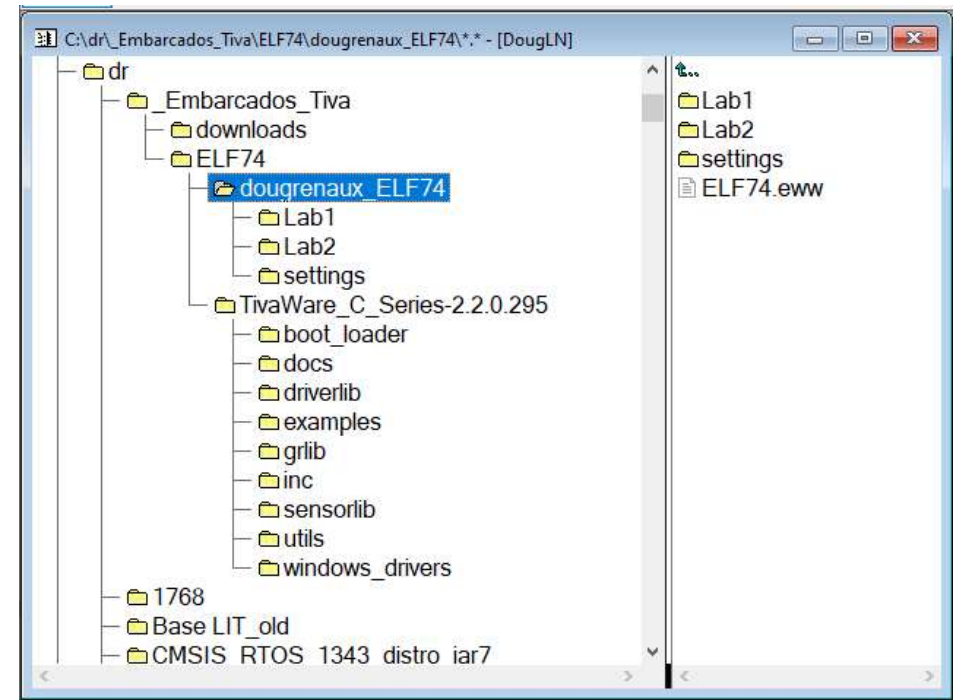
ELF74		
Name	Date modified	Type
 dougrenaux _CSW41	29-Sep-21 07:33	File folder
 TivaWare_C_Series-2.2.0.295	29-Sep-21 13:13	File folder

ao invés de dougrenaux_CSW41 você terá uma pasta seunome_CSW41 e esta pasta estará sincronizada com o seu repositório no GitHub. Não haverão contribuições suas na pasta TivaWare, portanto não há necessidade de sincronizá-la com o repositório.

A posição relativa destas pastas é muito importante. Não altere.

Lab 2

3. Na pasta *seunome_CS41*, crie uma pasta Lab2 caso esta não exista.



Lab 2

4. Inclua dois projetos neste workspace:

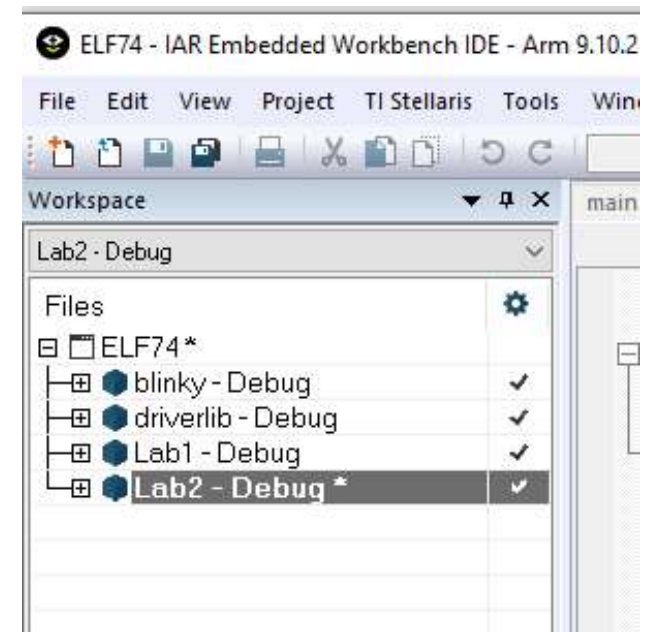
- driverlib (que está na pasta TivaWare)
- blinky (que está na pasta TivaWare)

e crie, caso não exista, um projeto novo: Lab2.

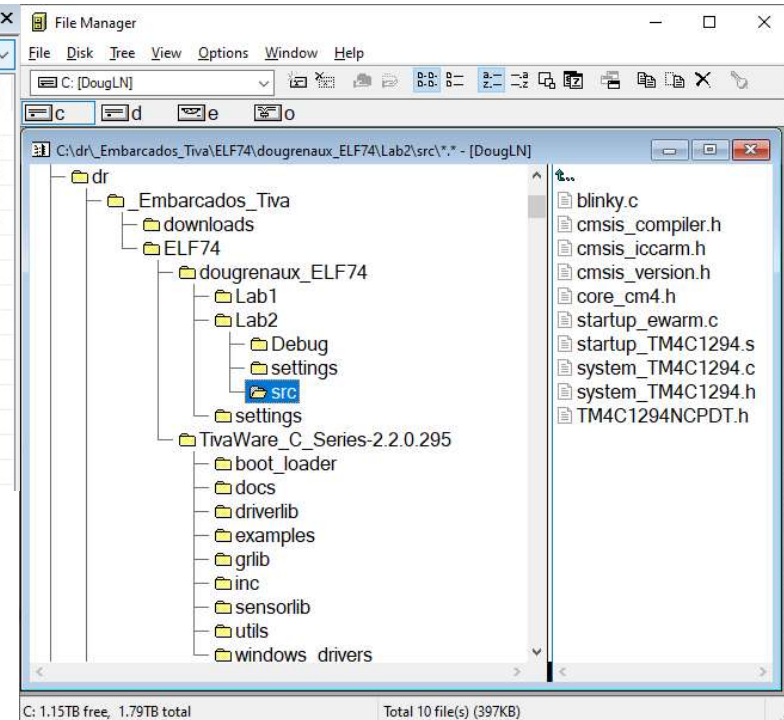
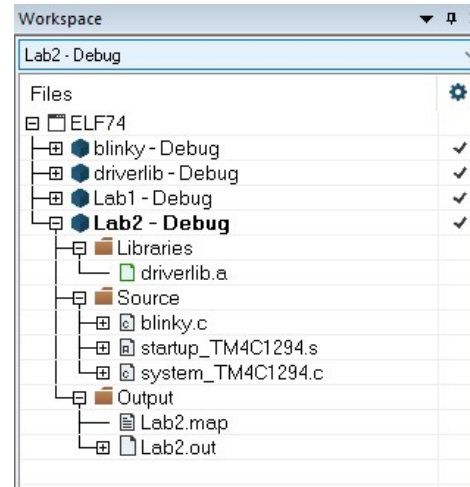
Certifique-se que os projetos Lab1, driverlib e blinky estão funcionando.

Certifique-se da correta configuração do projeto Lab2, para uso da placa Tiva e do TivaWare.

Certifique-se que não há caminhos absolutos nas configurações dos seus projetos. Todos os caminhos devem ser relativos, possivelmente fazendo uso das variáveis de ambiente do IAR: \$WORK_DIR\$ e \$PROJ_DIR\$.



Lab 2



5. Entenda como é o processo de inicialização que ocorre antes da função main ser chamada. Dica: na configuração de depuração, deselectione o “Run to main”, assim você poderá executar passo-a-passo, até mesmo em assembly, o processo de inicialização.
6. O exemplo original do blinky usa um loop para gerar a temporização da piscada do LED. Porque a variável *ui32Loop* é declarada como **volatile** dentro de *main()* ?
7. No próximo passo você fará uma alteração no projeto blinky, então é aconselhável copiar os arquivos fonte e os arquivos de configuração deste projeto para uma pasta blinky dentro da sua pasta *nome_ELF74*.
8. Altere este programa para que a temporização seja efetuada consultando o SysTick. Há funções prontas de acesso ao SysTick no TivaWare. Estão descritas no manual do TivaWare.

Lab 2

- O objetivo da aplicação a ser desenvolvida no Lab2 é medir o tempo de reação do usuário, isto será feito acendendo um LED e medindo o tempo até o usuário pressionar um botão. Pode até ser entendido como um jogo onde o objetivo é responder no menor tempo possível.

Lab 2

Um projeto bem elaborado deve incluir as seguintes atividades:

1. Planejamento das fases do processo de desenvolvimento.
2. Definição do problema a ser resolvido.
3. Especificação da solução.
4. Estudo da plataforma de HW (placa Tiva e seu processador).
5. Estudo da plataforma de SW (TivaWare).
6. Projeto (design) da solução.
7. Identificação (e entendimento) da funcionalidade do TivaWare e do HW que serão utilizadas na solução.
8. Configuração do projeto na IDE (IAR).
9. Edição do código da solução.
10. Teste e depuração.
11. Entrega dos resultados (sincronizar seu código com o seu GitHub).
Acrescente na pasta Lab2, um documento em pdf (ou no read.me do Github referente ao Lab2) relatando sobre o estudo que sugeri nos slides anteriores e suas atividades nos passos 1 a 10 deste slide, bem como o atendimento à Especificação (descrita no slide a seguir). **O formato deste relato está no arquivo word em anexo.** Submeter apenas uma entrega por equipe. Estimo cerca de 8 a 10 páginas em cada relatório.

Lab 2 - Especificação

Requisitos funcionais:

RF1 - O jogo deve ligar o LED D1 para informar o jogador do início da contagem de tempo.

RF1.1 - o LED deve ser aceso até 1 segundo após o início da operação da placa.

RF2 - O jogo usa o botão SW1 para entrada de dados pelo usuário.

RF3 - O jogo deve apresentar a contagem de tempo no Terminal do IAR indicando o número de clocks entre o LED acender e o botão SW1 ser pressionado e o valor de tempo correspondente em ms.

Requisitos e Restrições não funcionais:

RNF 1 - o limite superior de contagem de tempo é o equivalente a 3 segundos.

RNF 2 - usar funções da TivaWare para acesso a I/O, SysTick e temporização.

RNF 3 - a solução deve fazer uso de interrupções, obrigatoriamente de GPIO e opcionalmente do SysTick.

RNF 4 – o vetor de exceções deve estar em memória Flash e não na RAM.