

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)

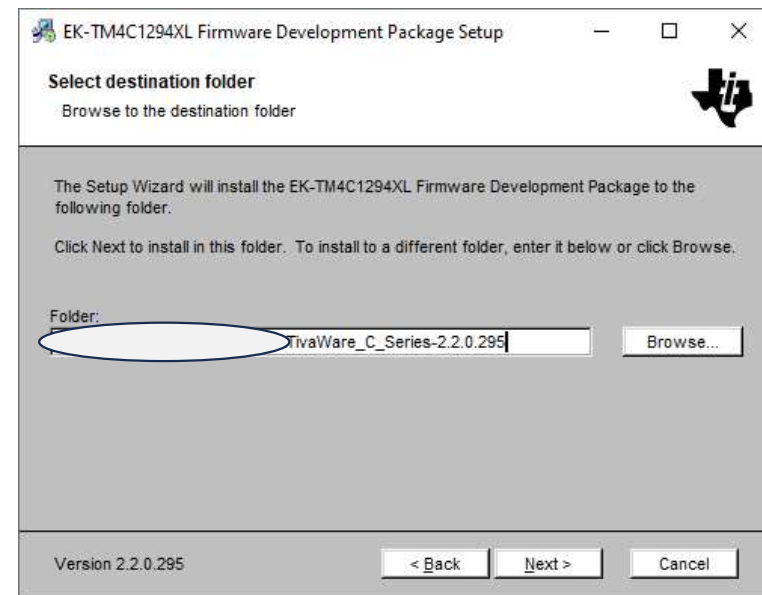
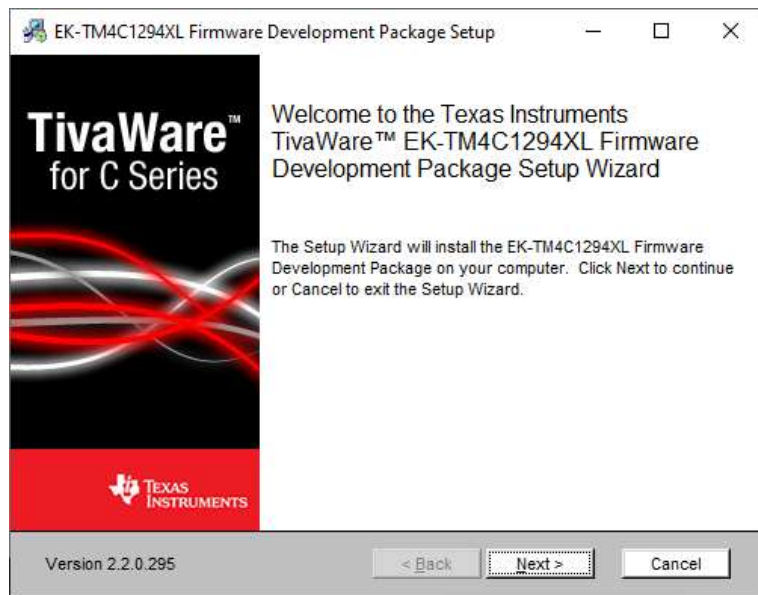
RELATED UTILITIES

 [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:



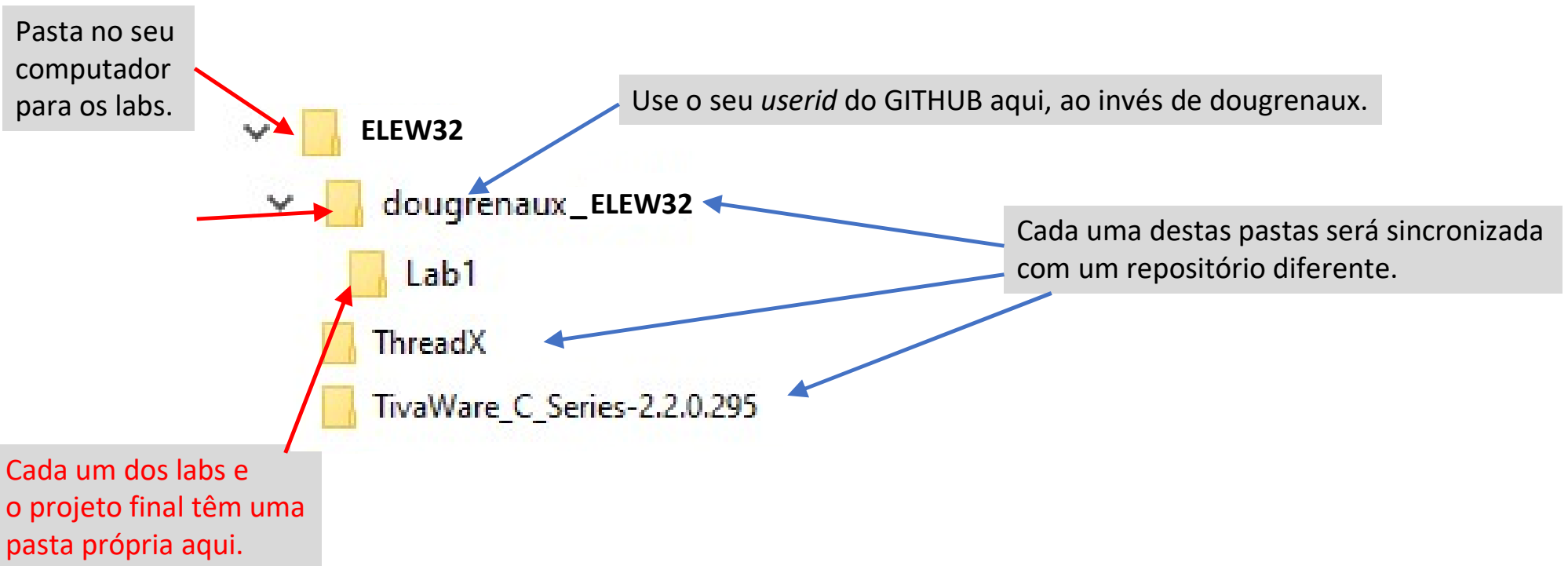
A pasta no seu computador onde estão os projetos de lab para esta disciplina deve estar organizada assim:

ELF74		
Name	Date modified	Type
 dougrenaux _ELEW32	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_ELEW32 você terá uma pasta seuid_ELEW32 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.

Slide do Lab 1 para reforçar a posição da pasta do TivaWare:



Lab 2 – interagindo com a placa Tiva

2. Estudando o TivaWare

2a) na pasta examples/boards/ek-tm4c1294xl, abram o projeto project0, revisem as configurações, executem. Estudem este projeto principalmente do ponto de vista de inicialização do clock e inicialização / uso do GPIO.

2b) na pasta examples/boards/ek-tm4c1294xl, abram o projeto watchdog, revisem as configurações, executem. Estudem este projeto principalmente do ponto de vista de interrupções com ISRs em Flash.

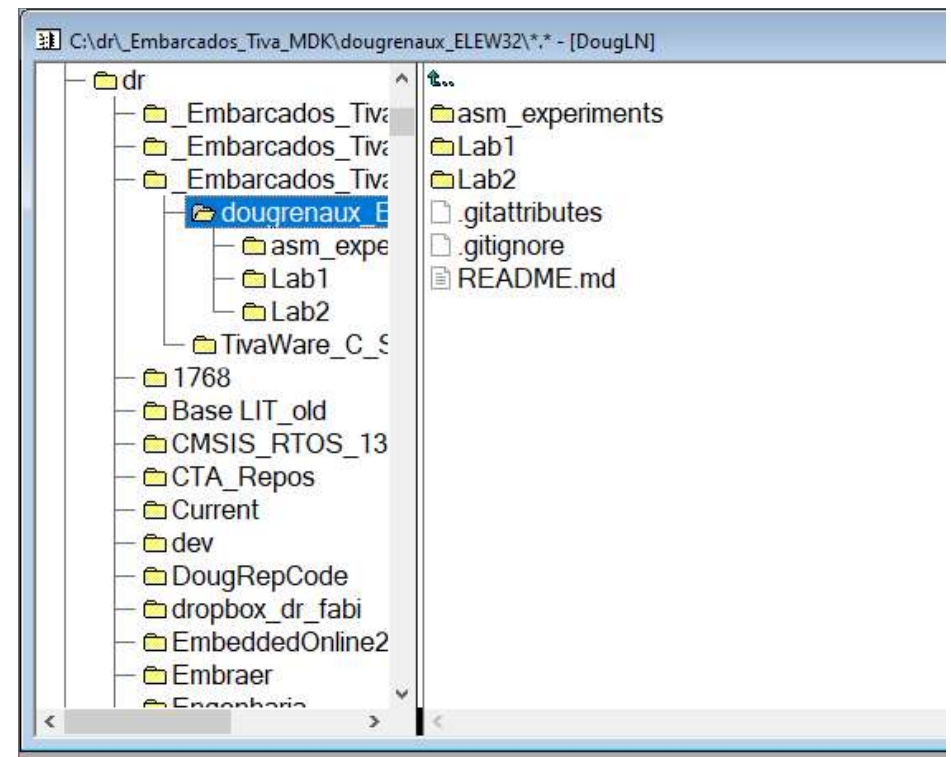
ATENÇÃO: as diretivas IF :DEF: ELSE e ENDIF são do compilador/assembler v5 e não são compatíveis com a versão 6 usada neste semestre. Se houver erros de compilação causados nestas linhas, substituam pelas diretivas atuais de pré-processamento:
#ifdef #else #endif

- As funções do TivaWare “sabem” se adaptar aos diferentes modelos de processadores da família TM4C. De que maneira informamos ao ambiente de compilação qual o processador que estamos usando?
- O que significa o prefix MAP_ antes das chamadas de funções do TivaWare, a exemplo de:

MAP_SysCtlPeripheralEnable

Lab 2

3. Na pasta *seuid_ELEW32*, crie uma pasta Lab2.



Lab 2

4. 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.

Inclua no seu relatório a sua explicação deste processo de inicialização.

Lab 2

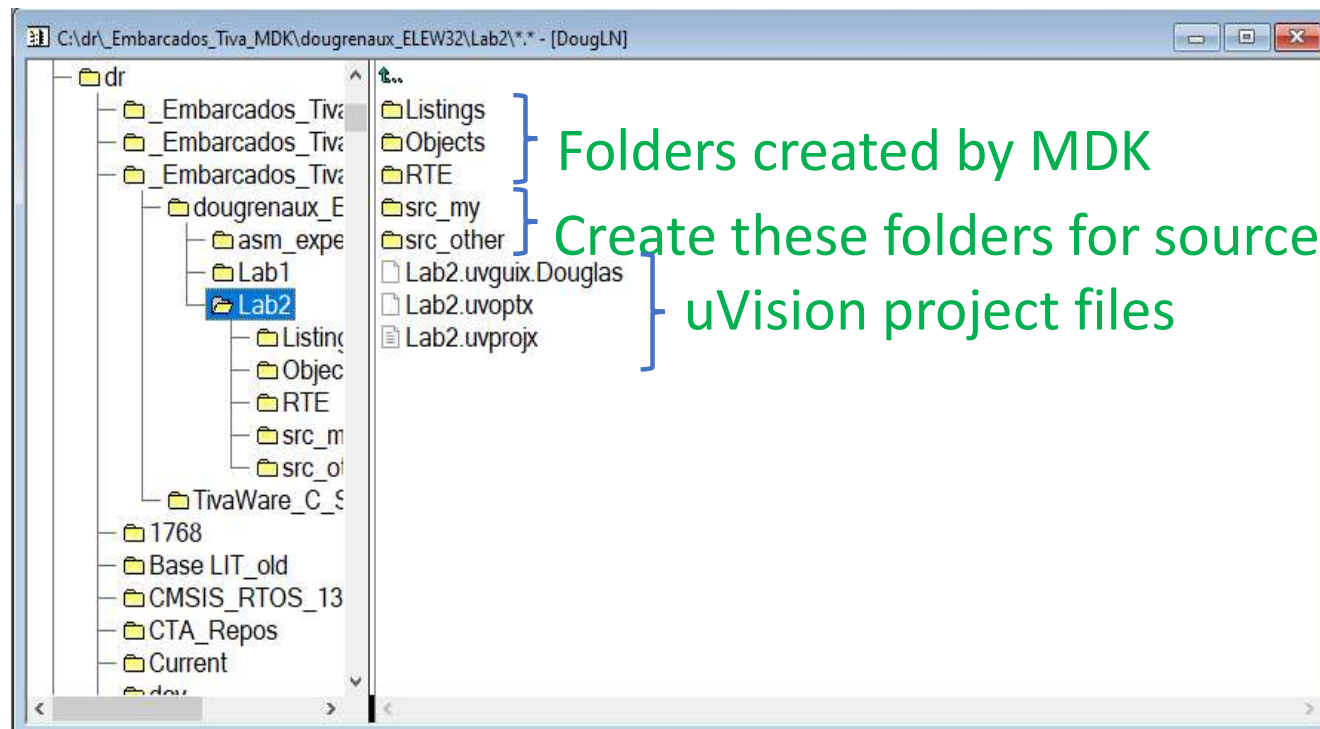
5. Crie um projeto novo na pasta Lab2. Observe cuidadosamente as opções de projeto dos projetos exemplo estudados anteriormente.

Observe o pdf que descreve a criação de um projeto no Keil-MDK.

Lab 2

In this project, the folders **Listings** and **Objects** do not contain input files, hence, these folders should **NOT** be uploaded to github.

The folders **RTE**, **src_my**, **src_other**, and the **uVision project files** should be uploaded to github.



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 (use a especificação no próximo slide como referência e faça os refinamentos que julgar necessário).
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 (Keil uVision).
9. Edição do código da solução.
10. Teste e depuração.
11. Entrega dos resultados (sincronizar seu código com o seu GitHub e relatório no formato .md).

Lab 2 - Especificação

Requisitos funcionais:

RF1 - O jogo deve ligar o LED D1 para informar ao jogador o 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 Serial 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 o SysTick como temporizador.

RNF 3 - usar funções da TivaWare para acesso a I/O, SysTick e temporização.

RNF 4 - a solução deve fazer uso de interrupções, obrigatoriamente de GPIO e opcionalmente do SysTick.

RNF 5 – o vetor de exceções deve estar em memória Flash e não na RAM.

Questões:

- qual o papel do registrador VTOR de um Cortex-M4 ? em que periférico integrado está localizado ?
- consultando o valor corrente do VTOR é possível saber se o vetor de exceções está na Flash ou na RAM ?
- que funções do TivaWare alteram a posição do vetor de exceções ? porque ?

Entrega

- Sincronizar o código fonte do Lab2 e arquivos de configuração com o seu github. Os projetos exemplo do TivaWare **não** devem ser colocados no github.
- Incluir um readme.md na pasta do Lab2 (no github) com o relatório deste experimento.
- O relatório deve tratar dos passos 1 a 10. **O formato deste relato está no arquivo em anexo.** Use o arquivo anexo apenas como referência pois a submissão é no readme.md. Submeter apenas uma entrega por equipe. O relatório também deve responder a todas as questões colocadas no enunciado deste experimento. Acrescente mais seções se necessário. Estimo cerca de 8 a 10 páginas em cada relatório.

Entrega

Questões adicionais para incluir no relatório:

Na situação em que a execução está parada na primeira linha da função main, logo após a carga do programa.

- qual o valor do registrador VTOR e que este valor significa?
- qual o valor corrente do PC e o respectivo significado?
- qual o valor corrente do registrador CONTROL e o que isto significa (examine cada campo deste registrador)?
- qual o valor corrente do registrador xPSR e o que isto significa (examine cada campo deste registrador)?
- quem está em uso, o MSP ou o PSP? qual seu valor? para onde ele aponta (RAM, Flash)?

Dicas - Manuais

Prof. Douglas

Material Suplementar

Bug debugger Keil - entra e já sai

Draft

Manuais da ARM

Edited 7:36 PM

ARMv7-M Architecture Reference Manual -
descrição da Arquitetura ARM versão 7 para Cortex-M. Modelo de memória, detalhamento das instruções.
<https://drive.google.com/open?id=1hnRNtwhQRjLobij9sg-WdMxOFMbWOOSq&authuser=0>
<https://developer.arm.com/documentation/ddi0403/latest/>

Cortex-M4 Devices Generic User Guide (ARM) -
visão do programador, conjunto de instruções, periféricos: NVIC, SCB, SysTick, MPU, FPU
<https://drive.google.com/open?id=16dNLG6QZ-pqn1ieeZJS9YRkRYe1mWPYo&authuser=0>
<https://developer.arm.com/documentation/dui0553/a/>

ARM Cortex-M4 Technical Reference Manual -
visão do programador, tempos de execução de instruções (tabela 3-1), Modelo de Memória, Write Buffer, Bit-Banding, periféricos: NVIC, MPU, System Control Regs, Debug/Trace Regs.
<https://drive.google.com/open?id=1D4DDwaQkTYk9s5-vdnyGN8P-DGTDHpzt&authuser=0>
<https://developer.arm.com/documentation/100166/0001/>

UAL - Unified Assembly Quick Reference Card
Tabela resumida do conjunto de instruções de todos os processadores ARM. Cuidado para identificar apenas as instruções Thumb2 da arquitetura ARMv7-M.
<https://drive.google.com/open?id=1vD746PNki6A5qEcc6ZGRvp8zfEDx14pU&authuser=0>
<https://developer.arm.com/documentation/qrc0001/m>

Quick Reference Card
Outra tabela das instruções do Cortex-M
<https://drive.google.com/open?id=1KgFadp8mUGuzavo8gM-9uBh-6P6Pltfd&authuser=0>

Manual Keil uVision:
<https://developer.arm.com/documentation/101407/0542?authuser=0>

Manual do Compilador v6:
<https://developer.arm.com/documentation/100748/0624/?lang=en>