

# Sistemas Embarcados CSW41

Matheus Mattos

## LAB 2

### 1. Planejamento das fases do processo de desenvolvimento.

O planejamento se deu por meio dos estudos iniciais da atividade a ser desenvolvida, das plataformas, a instalação do pacote da TivaWare, o entendimento da IDE para a criação e configuração correta do projeto e o entendimento do funcionamento do driverlib e do blinky.

### 2. Definição do problema a ser resolvido.

Deve ser desenvolvido um jogo de tempo de reação do usuário utilizando a placa Tiva C Series (EK-TM4C1294XL). Nesta atividade iremos trabalhar com interrupções por meio do SysTick, fazendo as modificações necessárias dentro do arquivo blinky.

### 3. Especificação da solução.

A solução para esse problema foi feita em cima dos requisitos funcionais e não funcionais definida na especificação do Lab 2:

#### 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.

Portanto foi alterado o código blinky e compilado no IAR, para resolver o jogo da reação. Para isso foi preciso fazer uso das configurações da GPIO e do SysTick, além do botão SW1 para a interrupção. Foi necessário também calcular o tempo de clock, utilizando essas ferramentas, para após a interrupção ser realizada pelo usuário, ser mostrada na tela.

Caso em até 3 segundos ela não seja efetuada, irá mostrar uma mensagem informando que não foi pressionado o botão dentro deste prazo.

### 4. Estudo da plataforma de HW (placa Tiva e seu processador).

Para um entendimento inicial sobre a plataforma de HW foi feito um estudo sobre a placa Tiva-C que é uma plataforma de prototipagem eletrônica de microcontroladores criada pelas Texas Instruments, onde possui um microcontrolador ARM Cortex-M4F CPU de 32 bits que opera de 80 a 120 MHz.

A placa utiliza simuladores da Stellaris LaunchPad para a comunicação com a CPU.

Foram utilizados os vídeos disponibilizados no ambiente virtual Classroom e também o material complementar, onde constavam seu datasheet e manuais que foram primordiais para esse laboratório.

## 5. Estudo da plataforma de SW (TivaWare).

Para o estudo da plataforma de Software, a TivaWare, possui a base para o desenvolvimento utilizando a placa Tiva, onde pode ser consultado pelo manual disponibilizado pelo professor. Possui bibliotecas para diversos drivers como periféricos, USB, Gráficos e Sensor. Também possui exemplos de códigos para o kit de desenvolvimento.

## 6. Projeto (design) da solução.

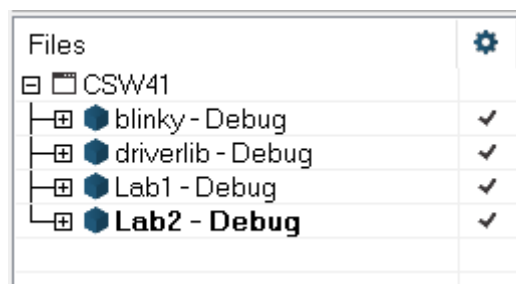
O LED SW1 e a chave são inicializados e após 1 segundo o LED D1 é acendido e depois disso se inicia o loop. Caso em menos de 3 segundos ocorra uma interrupção no botão SW1, e mostrado na tela a quantidade de clock até pressionar o botão e o tempo de clock que levou para ocorrer essa ação. Caso isso não ocorra dentro de 3 segundos, é impresso no terminal uma mensagem avisando que o botão não foi pressionado e portanto a interrupção não foi feita.

## 7. Identificação (e entendimento) da funcionalidade do TivaWare e do HW que serão utilizadas na solução.

Inicialmente foi feito uma análise das funcionalidades do TivaWare e do Hardware, foi utilizado os exemplos disponibilizados sendo eles o blinky e o driverlib. Após isso foi estudado os manuais de ambos.

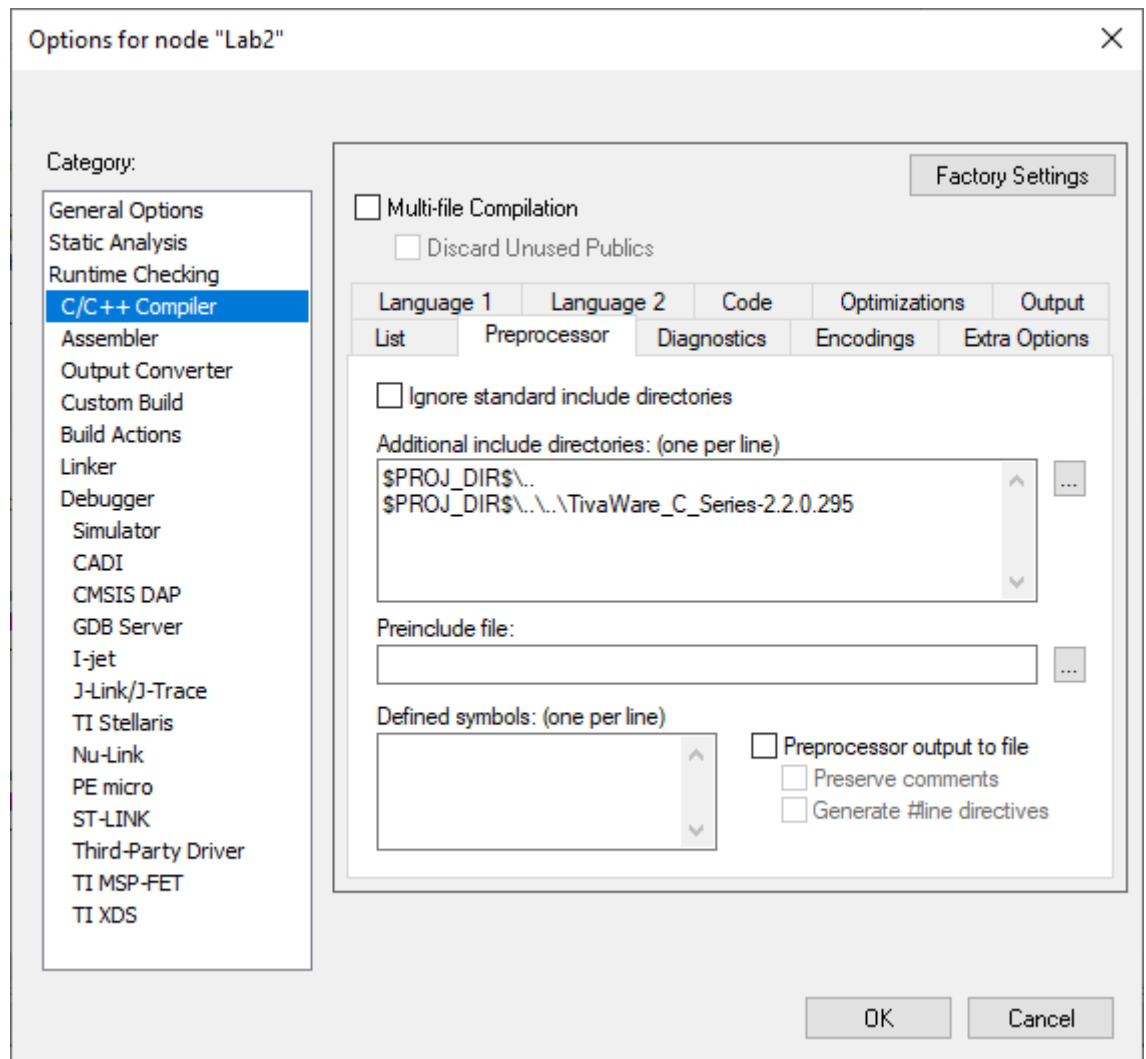
## 8. Configuração do projeto na IDE (IAR).

Primeiramente foi necessário importar os arquivos blinky e driverlib para o nosso workspace.

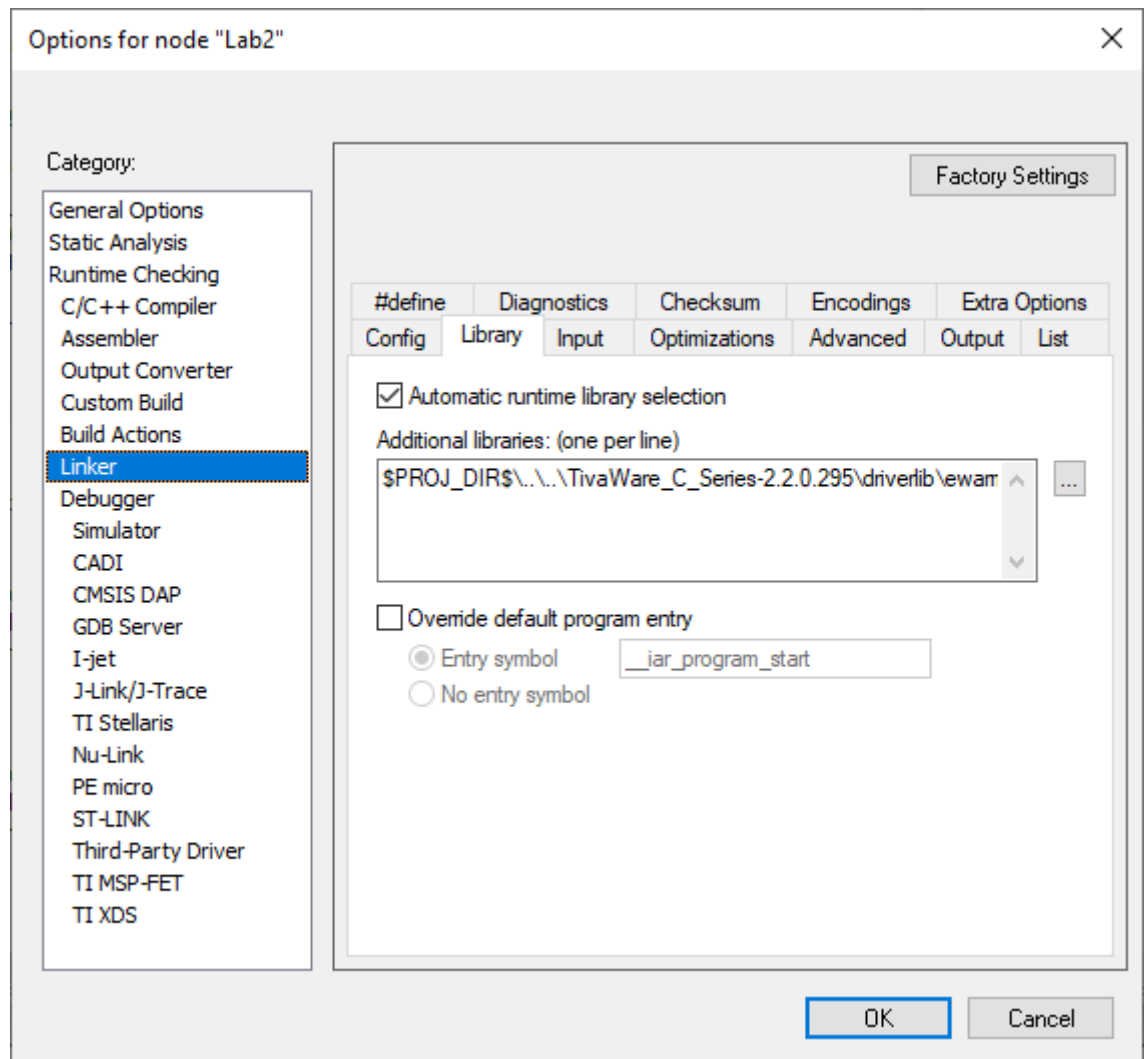


Após importado foi necessário efetuar as configurações de mapeamento de diretivas, pré-processador e de bibliotecas.

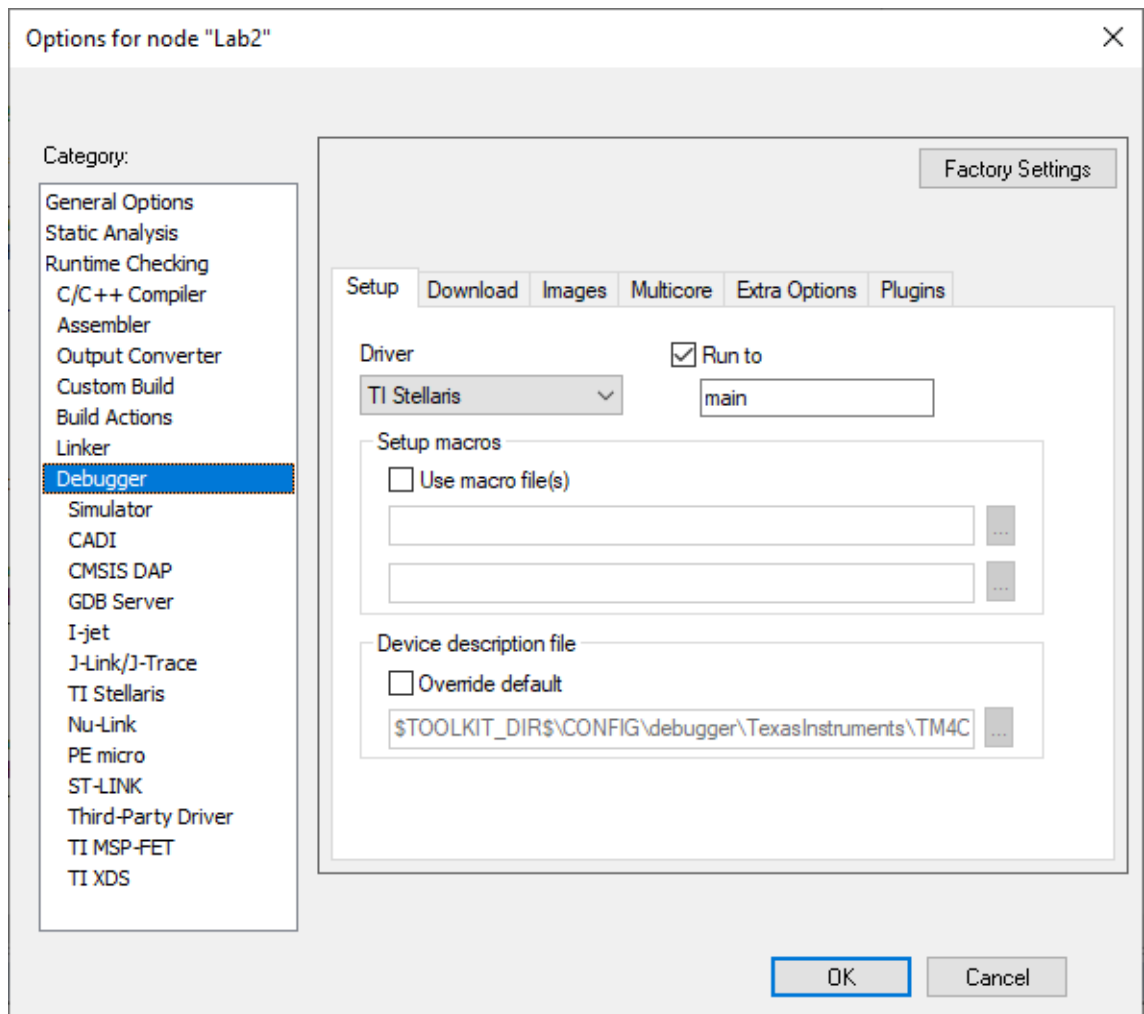
Pré-Processador:



Biblioteca - Driverlib:



Debugger – TI Stellaris:

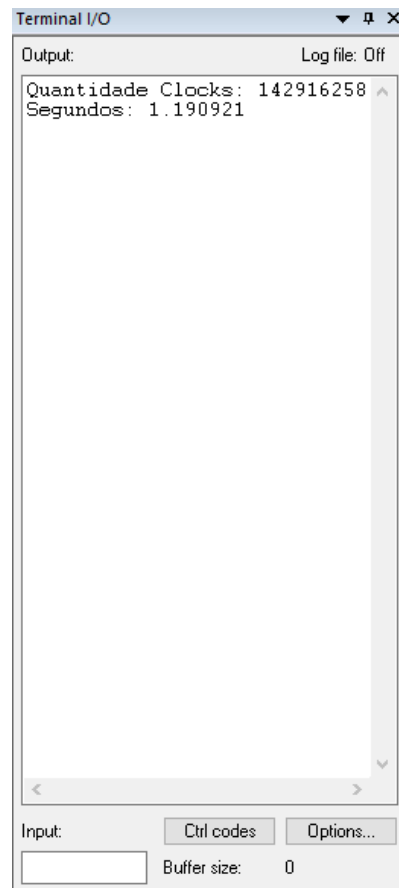


## 9. Edição do código da solução.

Foi necessário adicionar a interrupção do SysTick e um contador para calcular a quantidade de clock, além de setar as configurações da GPIO. Como a placa possui uma frequência de 120 MHz, então foi preciso setar o SysTickPeriodSet com 10000000 para que o tempo de 1 segundo solicitado, fossem o equivalente a 12 interrupções. Como o tempo máximo era de 3 segundos, foi setado o equivalente a 36 interrupções.

## 10. Teste e depuração.

Pressionando o botão na placa obtém-se o tempo de resposta e caso seja menor que 3 segundos é apresentando a seguinte mensagem:



Caso não seja pressionado nenhum botão em até 3 segundos é apresentada a seguinte mensagem:

