

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, como por exemplo da TivaWare, assim como do funcionamento do driverlib e do blinky. Também foi necessário entender o funcionamento correto do GitHub. A partir do material de aula e também de pesquisas na internet, foi possível entender qual a dinâmica da atividade.

2. Definição do problema a ser resolvido.

O principal problema dessa atividade, que trabalha com interrupções por meio do SysTick, foi entender a utilização dessa biblioteca para esse meio, além da integração com a Tiva e sua devida pinagem. Outra dificuldade também foi a modificação do arquivo blinky para ser utilizado no projeto do Lab 2.

3. Especificação da solução.

A solução para esse problema foi a criação de um código em C++, 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 a quantidade de clocks até esse momento.

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

Para um entendimento inicial sobre a placa Tiva, além de conhecimentos prévios de disciplinas, 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, com a qual não havia nenhuma familiaridade prévia, foi utilizado principalmente o manual desta disponibilizado pelo professor, juntamente com alguns exemplos pesquisados para entender seu funcionamento.

6. Projeto (design) da solução.

O LED SW1 é inicializado e a chave também, 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, o tempo de clock é mostrado na tela. Caso ela não ocorra, o clock máximo é impresso no terminal também juntamente com a mensagem de que a interrupção não foi feita.

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

Na etapa de identificação da funcionalidade do TivaWare e do Hardware, foram utilizados inicialmente os exemplos disponibilizados pelo professor, como por exemplo o blinky e o

driverlib. Depois desse passo, os manuais de ambos foram esmiuçados, assim como seus exemplos, para que o que foi solicitado pelo professor pudesse ser entregue.

8. Configuração do projeto na IDE (IAR).

A maior dificuldade da configuração da IDE foi a parte de deixar o Lab2 no mesmo formato que o blinky, pois haviam configurações não especificadas nos slides e no material que foram necessárias para que o programa funcionasse corretamente. Foi necessário também recriar o repositório no GitHub para que as pastas ficassem na forma solicitada pelo professor.

9. Edição do código da solução.

A parte mais complicada foi entender como o clock funcionava para poder trabalhar com o SysTick, assim como também entender a pinagem que seria utilizada para a solução e como configurar o jogo por meio dos comandos do SysTick e do GPIO. A placa possui uma frequência de 120MHz, então foi preciso setar o SYSTEM_CLOCK_FREQ como 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.

Após o código ser finalizado, os testes foram realizados diversas vezes até seu funcionamento correto. Abaixo podem ser mostrados os prints de funcionamento, em que o número de clocks é printado dependendo do tempo de reação, para ambos os casos.

