Lab 1

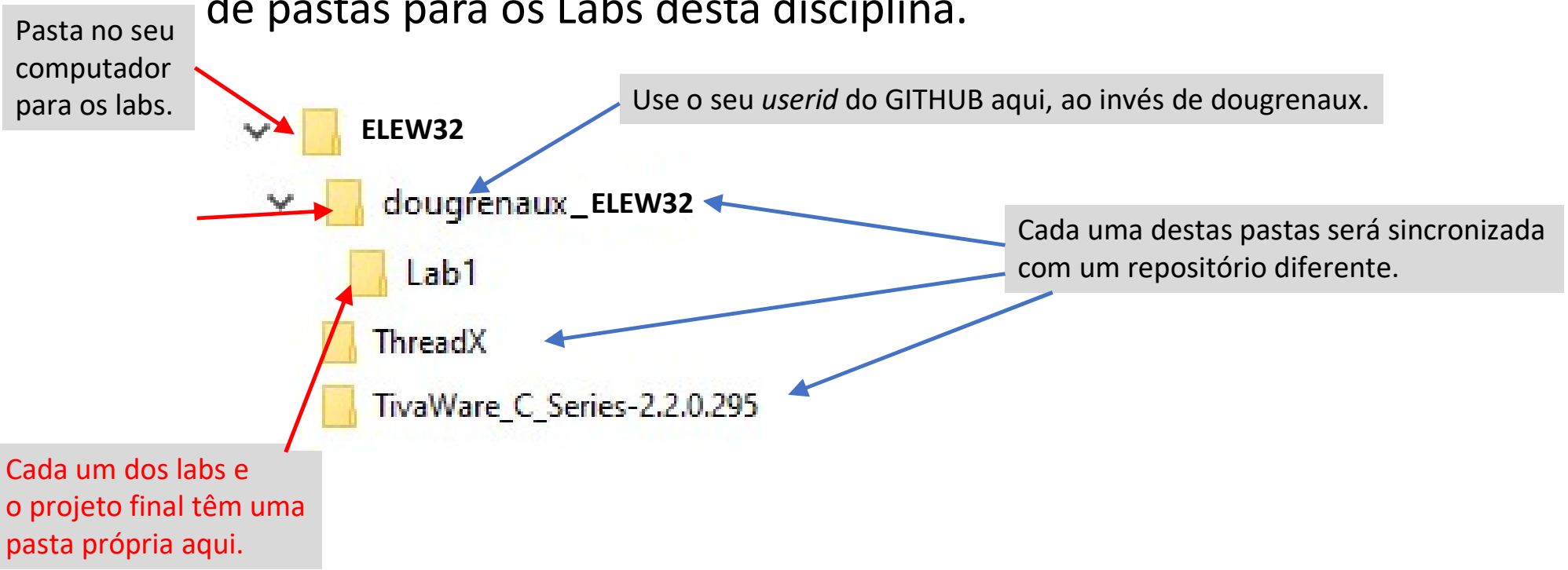
- Objetivos:
  1- validar a infraestrutura laboratorial necessária para os próximos experimentos: Keil uVision, Kit da placa Tiva e Github.
  Opcionalmente: Doxygen.
  2- entender as principais funcionalidades disponíveis no ambiente de desenvolvimento Keil uVision.

Lab 1 – passo a passo

1. Usem o Github para gerenciar versões do código e compartilhar com seu colega de equipe. Opcionalmente, usem o Doxygen de forma a documentar o código de cada lab no próprio código fonte. Ver dicas de uso nos slides a seguir.
   Não criem branches para cada um dos Labs.
2. Instalar o IDE da Keil versão **5.42a**.
   Há um passo-a-passo da instalação.
   Interessante ativar a licença “Community”.

Lab 1 – passo a passo

3. Em uma pasta vazia no seu computador, planeje a estrutura futura de pastas para os Labs desta disciplina.

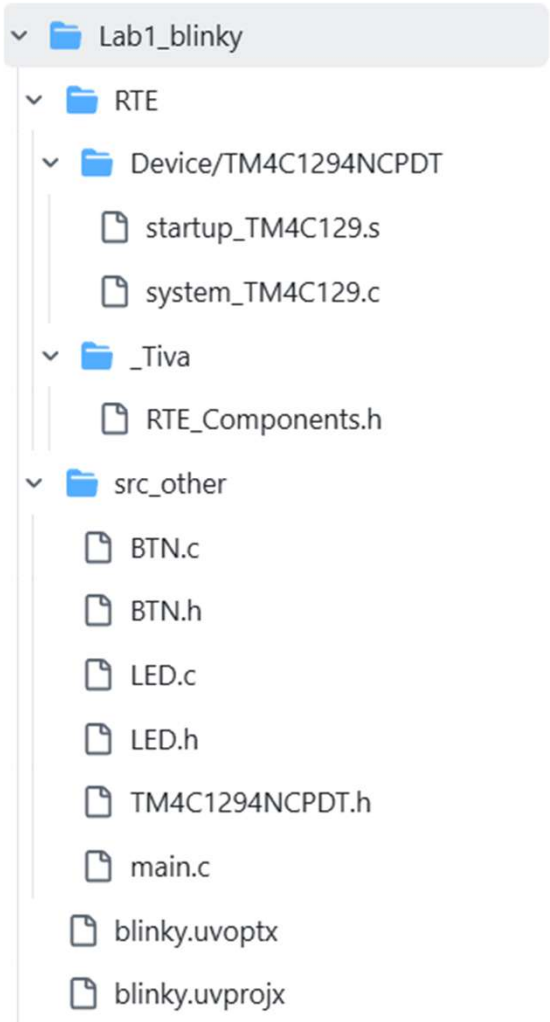


Lab 1 – passo a passo

- Os arquivos fonte na pasta *userid*_ELEW32 devem ser os arquivos de sua autoria, exceto por arquivos específicos de inicialização e configurações (p.ex. configuração do linker).

Lab 1 – passo a passo

4. Criar um projeto novo na pasta Lab1 a partir do meu github publico: https://github.com/dougrenaux/dougrenaux_ELEW32_public especificamente a pasta Lab1_blinky
   Verifique o funcionamento deste código.
   Entenda o que está sendo feito.



Lab 1 – passo a passo

5. Altere o projeto exemplo para incluir atribuir o valor destes símbolos pré-definidos (*predefined preprocessor symbols*):
   __cplusplus __DATE__ __TIME__ __FILE__ __LINE__ __STDC__ __ARMCC_VERSION __TARGET_ARCH_THUMB __STDC_VERSION__ à variáveis globais.
   Examine os valores atribuídos e interprete-os.
   (*dica: acesso o manual online do compilador ARM em uso para entender os valores de cada um destes símbolos)
   Que cuidados você precisará ter para que o compilador não descarte estes valores?

Lab 1 – passo a passo

6. Certifique-se que nas opções de configuração de projeto:
   - Texas TM4C1294NCPDT com coprocessador numérico
   - Depurador usado: Stelaris ICDI.
   O código deve ser executado na placa Tiva, não use o simulador.

7. Ao executar o programa,

- Como visualizar o valor dos símbolos pré-definidos?
- quais os valores dos símbolos pré-definidos que você apresentou ? o que significam ?

Lab 1 – passo a passo

8. Depuração:
   - quais os comandos referentes a controle da execução (passo-a-passo, step-over, step-out, breakpoints,...) ?
   - como visualizar registradores da CPU ?
   - como visualizar registradores de periféricos integrados ?
   - como visualizar/modificar endereços específicos de memória ?
   - como visualizar o valor de variáveis ou de expressões ?
   - porque nem todas as variáveis podem ter seu valor visualizado ?

Lab 1 – passo a passo

9. Entrega:
   - Conforme prazo estabelecido no Classroom.
   - Entregar via Classroom um link para seu repositório **privado** no github.
   - Repositórios privados devem autorizar o acesso pelo usuário *dougrenaux* (*douglasrenaux@utfpr.edu.br*) para que eu possa acessar seus resultados.
   - Utilizem um .gitignore para que arquivos objeto e binário não sejam salvos no seu repositório. Podem usar o .gitignore do meu repositório público como modelo.
   - Só encaminhe uma entrega por equipe.
   - Quando houver respostas textuais, a exemplo deste Lab, incluir a resposta no Readme.md, no github, na seção correspondente ao Lab1.

Obs: peço a gentileza de seguir cuidadosamente estes procedimentos ou terei que devolver as entregas aos autores para correção.

Lab 1 – passo a passo

9. Entrega (cont):
   O seu arquivo readme.md na pasta Lab1 deve incluir respostas para todas as perguntas realizadas nos slides anteriores.

   Dicas sobre sintaxe do arquivo readme.md:
   https://drive.google.com/open?id=1ox0My46xYgageoBRi0zmLy6ejxhywaLL&authuser=0

(*) – os arquivos que disponibilizo no Classroom só são acessíveis quando você está logado no navegador com usuário voce@alunos.utfpr.edu.br.