EA871 – LAB. DE PROGRAMAÇÃO BÁSICA DE SISTEMAS DIGITAIS

EXPERIMENTO 2 – Ferramentas de Desenvolvimento de *Software*

Profa. Wu Shin-Ting

OBJETIVO: Apresentação do ambiente de desenvolvimento de programas para MKL25Z128

ASSUNTOS: Construção de um projeto no ambiente *IDE CodeWarrior 10*: linguagem de programação C, documentação e depuração de programas

O que você deve ser capaz ao final deste experimento?

Ter uma visão da organização das ferramentas disponíveis no IDE CodeWarrior.

Saber criar um projeto no IDE CodeWarrior.

Ter uma noção de como um *firmware* (um código executável) é construído

Ter uma visão da relocação dos endereços de um *firmware*.

Depurar um código executado no nosso microcontrolador.

INTRODUÇÃO

Neste experimento vamos aprofundar o nosso entendimento do ambiente de desenvolvimento dos programas para controlar os periféricos integrados à placa FRDM-KL25 [1] que introduzimos no primeiro experimento [2]. Antes de escrever um programa, é recomendável organizar, no formato de um pseudocódigo, as ideias num fluxo de controle processável por uma máquina digital. Para torná-lo executável, é necessário utilizar um **editor** para escrever a sequência de ações definidas no pseudocódigo em instruções de uma linguagem de programação, usar um **compilador** para gerar a partir dos códigos-fonte códigos-objeto (linguagem de máquina), e um **ligador** para construir a partir dos códigos-objeto um código executável. E para executá-lo no microcontrolador, é necessário transferir via OpenSDA o código executável no formato elf para a memória FLASH do microcontrolador. Finalmente, para depurar e testar o programa no microcontrolador, um **depurador** serviria para controlar o fluxo de execução do programa, com paradas e passos de execução configuráveis, e monitorar/modificar as variáveis envolvidas.

Todas estas funções podem estar integradas num único ambiente de desenvolvimento, denominado IDE (*Integrated Development Environment*). Usaremos nesta disciplina o *CodeWarrior* 10, um *IDE* baseado na plataforma *Eclipse*. No *IDE CodeWarrior* são disponíveis compiladores para duas linguagens de alto nível, C e C++. Optamos por C para desenvolvimento dos programas nesta disciplina [3].

O IDE CodeWarrior dispõe de uma série de componentes visuais (ferramentas) na sua bancada de trabalho, passíveis a serem organizadas em diferentes perspectivas funcionais, para nos assistir o desenvolvimento de um *firmware* [4,5]. Por exemplo, adotando o procedimento padrão de criação de um novo projeto, é gerada automaticamente uma série de arquivos nas pastas `Project_Headers` e `Project_Settings/Startup_Code` que são ligados ao arquivo `main.c` e aos outros que você desenvolveu para gerar um executável de extensão `.elf`. É ainda gerado um *script*, na pasta `Project_Settings/LinkerFiles`, contendo informações necessárias para relocação dos endereços quando se carrega um *firmware* no espaço de endereços do microcontrolador.

Uma vez transferido o *firmware* para o microcontrolador, o IDE permite ainda que um desenvolvedor sincronize a execução de uma sequência de instruções com o seu fluxo de monitoramento através da interface SWD [6]. Para facilitar o diagnóstico dos erros, é possível visualizar um mesmo dado sob diferentes formatos de representação, criar pontos de parada condicionais e alterar o conteúdo da memória.

EXPERIMENTO

1. Crie um projeto no ambiente CodeWarrior com o seguinte trecho de instruções:

```
int main(void) {
    int counter, fatorial;

for(;;) {
        fatorial = counter;
            while (counter > 1) {
                fatorial *= counter-1;
                counter--;
            }
        }
        return 0;
}
```

Não use a entrada e saída serial nem as funções do *Processor Expert*.

2. Observe que o valor da variável **counter** não é definido e não há nenhuma instrução para mostrar a variável **fatorial**. Como você conseguiria obter os resultados dos seguintes fatoriais: 5!, 15! e 25! na perspectiva *Debug* do CodeWarrior? Descreva sucinta e claramente um procedimento reprodutível por seus colegas.

3. Utilize um recurso do ambiente IDE CodeWarrior para obter os resultados dos três fatoriais em quatro formatos: decimal, hexadecimal, octal e binário. Anote os valores em quatro formatos.

4. Quais foram os endereços relocados para as variáveis **counter** e **fatorial**? Capturar um *screenshot* da aba de Variáveis como justificativa da sua resposta.

5. Na aba "Memory" é possível configurar o número de *bytes* por coluna. Por *default*, são 4 *bytes* por coluna. Mude para 1 *byte* por coluna e responda: como são armazenados os *bytes* de uma varíavel? O *byte* mais sgnificativo no endereço de menor valor dentre o espaço de memória ocupado pela variável, ou vice-versa?

6. Qual é o endereço inicial carregado no PC quando se reseta o *firmware*? Capturar um *screenshot* da aba de Disassembly mostrando a primeira instrução e o respectivo endereço na memória.

7. Em qual espaço de memória do microcontrolador estão armazenadas todas as instruções do projeto? Qual é o endereço inicial e qual é o endereço final?

8. Altere os endereços iniciais dos segmentos de instrução (text), do segmento de dados (data) e do topo da pilha para 0x0000_1000, 0x1fff_f500 e0x2000_2f60, respectivamente. Como vocẽ altera os enderços desses segmentos? Como os endereços das instruções e dos dados seriam relocados? Registre os novos endereços para onde foram relocadas as variáveis e o segmento de instruções. Comparando com os endereços do item (7), o espaço de memória ocupado pelos códigos relocados é maior, igual ou menor? Justifique.

RELATÓRIO

O relatório deve ser devidamente identificado, contendo a identificação do instituto e da disciplina, o experimento realizado, o nome e RA do aluno. Para este experimento, responda os itens 1-8 do roteiro e suba-o no sistema *Moodle*.

REFERÊNCIAS

Todas as referências podem ser encontradas nos *links* abaixo ou na página do curso.

[1] *FRDM-KL25Z User's Manual – Freescale Semiconductors (doc. Number KL25P80M48SF0RM,* Setembro 2012*)*
ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/FRDMKL25Z.pdf
[2] Wu, Shin-Ting. EA871 - Roteiro 1 – 2s2020.
http://www.dca.fee.unicamp.br/cursos/EA871/2s2020/ST/roteiros/roteiro1.pdf
[3] Freescale CodeWarriorU. *Learn Programming with C*.
http://cache.freescale.com/files/training_presentation/TP_C_PROGRAMMING.pdf?lang_cd=en
[4] Freescale. *CodeWarrior Development Suite: Eclipse Quick Reference Windows*.
ftp://ftp.dca.fee.unicamp.br/pub/docs/ea076/complementos/CWMCUQRCARD.pdf
[5] Wu Shin-Ting e A.A.F. Quevedo. Ambiente de Desenvolvimento – *Software*
ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/apostila_C/AmbienteDesenvolvimentoSoftware.pdf
[6] 2-Pin Debug Port
https://developer.arm.com/architectures/cpu-architecture/debug-visibility-and-trace/coresight-architecture/serial-wire-debug

Agosto de 2016.
Revisado em Fevereiro de 2017.
Revisado em Julho de 2017.
Revisado em Fevereiro de 2018.
Revisado em Setembro de 2020.