



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FICHA DE COMPONENTE CURRICULAR

CÓDIGO:	COMPONENTE CURRICULAR: SISTEMAS EMBARCADOS I	
UNIDADE ACADÊMICA OFERTANTE: FACULDADE DE ENGENHARIA ELÉTRICA	SIGLA: FEELT	
CH TOTAL TEÓRICA: 45 horas	CH TOTAL PRÁTICA: 30 horas	CH TOTAL: 75 horas

OBJETIVOS

Objetivos Gerais

Esta disciplina tem como objetivo integrar os conceitos apresentados em diversas disciplinas da Engenharia Elétrica através do estudo e implementação de hardware e firmware de dispositivos industriais.

- Visão sistêmica do processo de desenvolvimento de hardware e firmware.
- Técnicas de codificação eficiente.
- Estratégias de criação de firmware.
- Estudo da plataforma ARM Cortex M.
- Portabilidade.
- Conhecimento detalhado dos principais protocolos de comunicação digital (UART, I2C, SPI).
- Uso eficiente da linguagem C no desenvolvimento de firmware.
- Uso e desenvolvimento de hardware e firmware com o emprego de microcontroladores Cortex M.
- Apresentação de paradigmas diversos de construção de firmwares (voltado a interrupções, super loop, co-rotinas e RTOS).
- Estratégias para criação de código portável (encapsulamento, ponteiros opacos, abstração de hardware).
- Projeto de hardware microprocessados.

EMENTA

Estudo e desenvolvimento de sistemas embarcados microcontrolados. Visão sistêmica do processo de desenvolvimento de hardware e firmware; Técnicas de codificação eficiente; Estudo da arquitetura de um microcontrolador; Estratégias de criação de firmware; Portabilidade.

PROGRAMA

1. Introdução ao mercado de sistemas embarcados. Possibilidades de inovação e emprego da tecnologia embarcada.

- 2. Arquitetura de computadores**
 - 2.1. Paradigmas RISC e CISC
 - 2.2. Stack pointer, program counter e registros de status
 - 2.3. Modos de endereçamento
 - 2.4. Vetores e tratamento de interrupções
 - 2.5. Dispositivos de memória.
 - 2.5.1. Terminologia de memória;
 - 2.5.2. Princípios de operação da memória;
 - 2.5.3. Memória utilizadas em sistemas embarcados (DRAM, SRAM, ROM, Flash, EEPROM, FRAM)
- 3. Ferramentas de desenvolvimento (toolchains)**
 - 3.1. Ferramentas do toolchain GNU (ar, as, gcc, ld, strip, nm, strings)
 - 3.2. Arquivos de linker e de partida
 - 3.3. Uso de IDEs para desenvolvimento e uso de linha de comando
 - 3.4. Bibliotecas C e código de partida para sistemas embarcados
- 4. Ferramentas de desenvolvimento ST Microelectronics**
 - 4.1. Configuração de pinos/funções no CubeMX. Geração de código
 - 4.2. Compilação e depuração no System Workbench
- 5. Arquitetura ARM Cortex M**
 - 5.1. Família Cortex M. Conjunto de instruções Cortex M
 - 5.2. Arquitetura Cortex M. Registradores comuns do Cortex M. Estados e modos de operação do processador. Inicialização do processador
 - 5.3. Registros especiais (Control Register, Program Status Register, System Control Block)
 - 5.4. Exceções e interrupções no Cortex M. Controle de interrupções. Prioridades. Entrada e saída de modo interrompido. Contextos de interrupção. Contexto de interrupção com ponto flutuante. Latência de interrupções
 - 5.5. Mapa de memória
 - 5.6. Barramentos e DMA
 - 5.7. Acessos alinhados e não alinhados. Operações atômicas de acesso a bits. (bit-band)
 - 5.8. Modos de baixo consumo
 - 5.9. Proteção de memória
- 6. Microcontroladores ARM Cortex STM32**
 - 6.1. Visão geral dos controladores STM32
 - 6.2. GPIO no STM32, conceitos de portas e pinos.
 - 6.3.
 - 6.4. Interrupções no STM32.
 - 6.5. Teoria geral sobre UART. UART no STM32. Registros relacionados à UART no STM32. Uso com interrupções. UART com buffer circular.
 - 6.6. Conceitos gerais de timers e PWM. Timers e PWM no STM32.
 - 6.7. RS485 no STM32. Uso de transceiver RS485.
 - 6.8. , tipos de ADC e seus cionamento, circuitos sampling and holding e multiplexação)
 - 6.9. ADC no STM32 (pooling,).
 - 6.10. Conceitos gerais do protocolo SPI. Protocolo SPI no STM32.
 - 6.11. Conceitos gerais do protocolo I2C. Protocolo I2C no STM32.
 - 6.12. Conceito , tipos de DAC e seus
 - 6.13. , filtros de reconstrução).
 - 6.14. Conversor DAC no STM32.
- 7. Estratégias de construção de firmware**
 - 7.1. Super loop

- 7.2. Voltado a interrupções
- 7.3. Máquinas de estados
- 7.4. Co-rotinas
- 7.5. Visão geral de um RTOS (Sistema operacional de tempo real).
- 7.6. MISRA C

8. Projeto com microcontroladores

- 8.1.
- 8.1.1.
- 8.1.2. Fan-in Fan-out
- 8.1.3. Atrasos de propagação e frequência máxima de operação
- 8.1.4. (coletor/dreno aberto, tri-state e totem-pole)
- 8.1.5.
- 8.1.6. Cuidados adicionais no uso de pinos CMOS não utilizadas
- 8.2. Reguladores de tensão
- 8.3. Desacoplamento
- 8.4. CAD para projeto eletrônico
 - 8.4.1. Visão geral do uso de ferramenta CAD
 - 8.4.2. Esquemáticos
 - 8.4.3. Biblioteca de componentes
 - 8.4.4. Netlist e lista de materiais
 - 8.4.5. Projeto da placa de circuito impresso
 - 8.4.6. Criação de footprints
 - 8.4.7. Geração de arquivos de fabricação
- 8.5. Projeto de hardware com controladores STM32
 - 8.5.1. Escolha do controlador e mapeamento dos pinos
 - 8.5.2. Alimentação (digital e analógica)
 - 8.5.3. Desacoplamento
 - 8.5.4. Relógios (clocks)
 - 8.5.5. Roteamento de linhas especiais
 - 8.5.6. Interligação com periféricos diversos
- 8.6. Montagem e depuração de placas eletrônicas

BIBLIOGRAFIA BÁSICA

ALMEIDA, R.; MORAES, C.; SERAPHIM, T. **Programação de sistemas embarcados:** desenvolvendo software para microcontroladores em linguagem C. 1. ed. Rio de Janeiro: Elsevier, 2016.

PRESSMAN, Roger S. **Engenharia de software:** uma abordagem profissional. 7. ed. Porto Alegre: McGraw-Hill, 2011.

TANENBAUM, Andrew S. **Organização estruturada de computadores.** 6. ed. São Paulo: Pearson, c2013.

TOCCI, Ronald J. **Sistemas digitais:** princípios e aplicações. 11. ed. São Paulo: Pearson Education do Brasil, c2011.

YIU, J. **The definitive guide to ARM Cortex-M3 and Cortex-M4 processors.** 3. ed. Newnes/Elsevier, 2014.

BIBLIOGRAFIA COMPLEMENTAR

BACKES, André. **Linguagem C**: completa e descomplicada. Rio de Janeiro: Elsevier, 2013.

BARR, Michael; MASSA, Anthony. **Programming embedded systems**: with C and GNU development tools. O'Reilly Media, 2006.

HOOK, Brian. **Write portable code**: an introduction to developing software for multiple platforms. No Starch Press, 2005.

HYDE, Randall. **Write great code**: understanding the machine. No Starch Press, 2012. v.1.

KLEMENS, Ben **21st Century C**: C tips from the new school. O'Reilly Media, 2015.

GRENNING, James W. **Test driven development for embedded C**. Pragmatic Bookshelf, 2011.

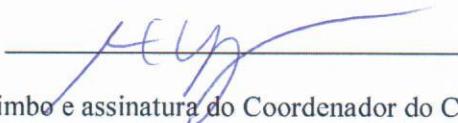
MONTGOMERY, Stephen L. **MISRA C**: Guidelines for the Use of the C Language in Critical Systems 2012. Misra, 2013.

SINK, E. **Version control by example**. Pyrenean Gold Press, 2011.

WHITE, E. **Making embedded systems**: design patterns for great software. O'Reilly Media, 2014.

APROVAÇÃO

06 / 09 / 18


Carimbo e assinatura do Coordenador do Curso

Universidade Federal de Uberlândia
Prof. Dr. Marcelo Rodrigues de Sousa
Coordenador do Curso de Engenharia da Computação
Portaria R. N° 1234/2017

10 / 09 / 18


Carimbo e assinatura do Diretor da
Unidade Acadêmica
Universidade Federal de Uberlândia
Prof. Dr. Sérgio Ferreira da Paula Silva
Diretor da Faculdade de Engenharia Elétrica
Portaria R N°. 708/17