

IOT E COMPUTAÇÃO FÍSICA

Mestrado em Engenharia em Desenvolvimento de Jogos Digitais

Código: 27205

Área Científica Predominante: Sistemas e Controlo Inteligente

Docente: Tiago Rafael dos Santos Martins Pereira Rodrigues

Idioma de Instrução: Português

Regime: S2

Carga Letiva: 30h Carga Trabalho: 138h

ECTS: 5,0

Objetivos

Fornecer uma visão geral sobre a computação física e Internet of Things (IoT).

Promover o conhecimento de conceitos de eletrónica, componentes eletrónicos e de microcontroladores, com o foco no desenvolvimento de jogos.

Desenvolver conhecimentos para a programação de microcontroladores.

Promover o conhecimento para o desenvolvimento um jogo com recurso à computação física e IoT.

Resultados da Aprendizagem

Adquirir conhecimentos básicos de diferentes componentes eletrónicos, sensores e atuadores.

Adquirir conhecimentos básicos de construção de circuitos eletrónicos.

Adquirir conhecimentos para a programação de microcontroladores.

Adquirir competência para utilizar a IoT em projetos de computação física.

Adquirir capacidade de projetar e desenvolver um jogo com recurso à IoT e computação física.

Conteúdos Programáticos

1. Introdução à computação física e IoT

1.2. Noções básicas de eletrónica

1.2. Componentes eletrónicos

1.3. Sensores e atuadores

1.4. Microcontroladores e microprocessadores: comparação da arquitetura das plataformas de prototipagem: Arduino e RaspberryPi.

1.5. Internet of Things (IoT)

2. Programação de microcontroladores

2.1. Introdução ao Arduino IDE

2.2. Introdução à programação do Arduino (tipo de dados, ciclos, estruturas condicionais, funções e procedimentos, etc.)

2.2. Montagem de circuitos simples utilizando uma protoboard

3. I/O digital e analógico no Arduino

3.1. Configuração de entradas e saídas digitais no Arduino

3.2. Modulação de comprimento de pulso (PWM)

3.3. Princípio de funcionamento de um LED RGB e controlo de cor emitida a partir de um sinal de controlo PWM

3.4. Princípio de funcionamento de um motor DC e controlo da sua velocidade de rotação a partir de sinal de controlo PWM

3.5. Componentes de um servomotor, princípio de funcionamento e controlo de movimento a partir de um sinal de controlo PWM

3.6. Princípio de funcionamento de um piezo tweeter (Buzzer) e geração de sinais sonoros com diferentes frequências e duração

4. O processo genérico de conversão analógico-digital

4.1. Entradas analógicas no Arduino

4.2. Digitalização de um sinal analógico

4.3. Codificação binária (exemplo de 3 bits)

4.4. Características do módulo de conversão A/D

4.5. Exemplos de funcionamento de sensores analógicos

4.6. Conversão A/D no Arduino

4.7. Utilização do potenciómetro como sensor de posição

4.8. Exemplo de aplicação com elaboração de um device driver para medição de rotação angular usando um potenciómetro

4.9. Exemplo de aplicação com o sensor de luminosidade baseado num LDR

4.10 Exemplo de aplicação com o sensor de temperatura

5. Programação orientada a eventos no Arduino

5.1. Pooling vs. interrupções

5.2. Vantagens das interrupções e suas aplicações

5.3. Processo de atendimento a uma interrupção

5.4. Rotina de serviço à interrupção (ISR)

5.5. Interrupções no Arduino

5.6. Interrupções externas digitais (falling edge, rising edge, etc)

5.7. interrupções periódicas utilizando a biblioteca "Timer.h"

6. Programação orientada a objetos no Arduino

6.1. Pensar "Objetos"

6.1. Introdução à programação POO em Arduino – abstração, encapsulamento, herança e polimorfismo

7. Projeto prático 7.1. Acompanhamento ao projeto prático

Bibliografia Recomendada

McKerrow, P., Introduction to robotics, Addison-Wesley, 1991

Craig, J. J., Introduction to Robotics Mechanics and Control, Addison-Wesley Publishing Company, 1989

Jeremy, B., Exploring Arduino: Tools and Techniques for Engineering Wizardry, Wiley, 2013

Pfister, C., Getting started with the Internet of things, O'Reilly Media, Inc, 2011

Métodos de Ensino e de Aprendizagem

Fornecer uma visão geral sobre a computação física e Internet of Things (IoT).

Noções básicas de eletrônica

Componentes eletrônicos

Sensores e atuadores

Microcontroladores e microprocessadores: comparação da arquitetura das plataformas de prototipagem: Arduino e RaspberryPi.

Internet of Things

Promover o conhecimento de conceitos de eletrônica, componentes eletrônicos e de microcontroladores, com o foco no desenvolvimento de jogos.

Noções básicas de eletrônica

Componentes eletrônicos

Sensores e atuadores

Desenvolver conhecimentos para a programação de microcontroladores.

2. Programação de microcontroladores

2.1. Introdução ao Arduino IDE

2.2. Introdução à programação do Arduino (tipo de dados, ciclos, estruturas condicionais, funções e procedimentos, etc.)

2.2. Montagem de circuitos simples utilizando uma protoboard

3. I/O digital e analógico no Arduino

3.1. Configuração de entradas e saídas digitais no Arduino

3.2. Modulação de comprimento de pulso (PWM)

3.3. Princípio de funcionamento de um LED RGB e controlo de cor emitida a partir de um sinal de controlo PWM

3.4. Princípio de funcionamento de um motor DC e controlo da sua velocidade de rotação a partir de sinal de controlo PWM

3.5. Componentes de um servomotor, princípio de funcionamento e controlo de movimento a partir de um sinal de controlo PWM

3.6. Princípio de funcionamento de um piezo tweeter (Buzzer) e geração de sinais sonoros com diferentes frequências e duração

4. O processo genérico de conversão analógico-digital

4.1. Entradas analógicas no Arduino

4.2. Digitalização de um sinal analógico

4.3. Codificação binária (exemplo de 3 bits)

4.4. Características do módulo de conversão A/D

4.5. Exemplos de funcionamento de sensores analógicos

4.6. Conversão A/D no Arduino

4.7. Utilização do potenciómetro como sensor de posição

4.8. Exemplo de aplicação com elaboração de um device driver para medição de rotação angular usando um potenciómetro

4.9. Exemplo de aplicação com o sensor de luminosidade baseado num LDR

4.10 Exemplo de aplicação com o sensor de temperatura

5. Programação orientada a eventos no Arduino

5.1. Pooling vs. interrupções

5.2. Vantagens das interrupções e suas aplicações

5.3. Processo de atendimento a uma interrupção

5.4. Rotina de serviço à interrupção (ISR)

5.5. Interrupções no Arduino

5.6. Interrupções externas digitais (falling edge, rising edge, etc)

5.7. interrupções periódicas utilizando a biblioteca "Timer.h"

6. Programação orientada a objetos no Arduino

6.1. Pensar "Objetos"

6.1. Introdução à programação POO em Arduino – abstração, encapsulamento, herança e polimorfismo

Promover o conhecimento para o desenvolvimento um jogo com recurso à computação física e IoT.

7. Projeto prático

7.1. Acompanhamento ao projeto prático

Métodos de Avaliação

As classificações finais serão calculadas tendo em conta os seguintes critérios:

10% - Participação e Assiduidade

35% - Pré-Projeto

55% - Projeto Final

- A não entrega do pré-projeto ou a do projeto final serão classificadas com 0 (zero). - Os trabalhos entregues fora do prazo terão uma penalização de 20% com o limite de 24 horas de entrega. Para além desse tempo considera-se não entregue.