

Laboratórios de Telecomunicações e Informática I

Mestrado Integrado em Engenharia de Telecomunicações e Informática

Projeto da UC

Ano Letivo 2020/2021

1. Objetivos

Com este projeto pretende-se que sejam atingidos os seguintes resultados de aprendizagem:

- Compreender a utilidade do modelo OSI, as funcionalidades associadas a cada camada, e a sua adaptação a diferentes contextos de comunicação;
- Conhecer e utilizar diferentes interfaces de comunicação série para comunicação com periféricos, nomeadamente as interfaces SPI e UART/RS-232;
- Desenvolver um sistema de comunicação de dados sem fios com base em componentes disponíveis comercialmente;
- Aprender a conceber e executar testes de avaliação de desempenho de sistemas de comunicação, a documentar os testes e analisar resultados;
- Entender a necessidade da camada 2 do modelo OSI e aplicar as suas principais funcionalidades;
- Perceber e implementar as funcionalidades da camada 7 do modelo OSI e compreender a diferença entre as camadas de aplicação e ligação de dados;
- Aprofundar as noções de programação estruturada e modular através do uso de APIs;
- Perceber a importância de usar uma metodologia adequada na abordagem ao projeto;
- Aprofundar as competências do trabalho em equipa e autónomo;

- Aplicar os conhecimentos adquiridos noutras UCs do 1º semestre do 3º ano e dos anos anteriores.

2. Organização e Funcionamento

O projeto deverá ser desenvolvido em grupo dentro das aulas da unidade curricular (2 sessões semanais de 3 horas cada) e fora do horário das aulas, com uma **dedicação mínima de 12 horas efetivas de trabalho para cada aluno por semana**.

Os alunos deverão organizar-se em grupos de 4 alunos (máximo de 7 grupos) o mais rápido possível, para não haver atrasos no desenvolvimento do projeto. O delegado da turma deverá compilar a lista de grupos formados e enviar aos docentes num ficheiro Excel com 3 colunas: nº do grupo (1, 2, 3 ...), nº mecanográfico e nome do aluno, após a primeira aula da UC.

3. Enquadramento do Projeto

Em termos de aplicação, este projeto consiste no desenvolvimento de uma **aplicação de conversação (chat)** que permita a **conversação em modo texto, em tempo real, entre dois computadores pessoais** (PC – *Personal Computer*) do sistema de comunicação desenvolvido no âmbito do projeto. Neste sentido, ao pressionar-se a tecla Enter, o texto escrito num PC deverá ser imediatamente enviado e apresentado noutro PC do sistema, e vice-versa. A aplicação deve também permitir **funcionalidades adicionais, como a transferência de ficheiros (de qualquer formato, incluindo texto e imagem)** entre os computadores.

Em termos de **arquitetura de rede**, o sistema de comunicação a desenvolver consiste numa **rede local sem fios por radiofrequência (RF)** que possibilita a **comunicação bidirecional, fiável e simultânea entre duas ou mais estações**.

Este projeto requer a aprendizagem e utilização de funcionalidades de diferentes camadas do modelo OSI (*Open Systems Interconnection*), conforme descrito nas seções seguintes.

3.1. Camada física do sistema de comunicação (Primeira camada)

A camada física é responsável pela transmissão do sinal (elétrico, radiofrequência, ótico, etc.) entre dois dispositivos num meio físico adequado, sendo sua competência os aspetos associados à codificação e transmissão do sinal.

Neste projeto, serão utilizados módulos *transceivers* de RF comerciais para a transmissão de dados sem fios, não sendo assim necessário desenvolver o hardware de raiz. Não obstante, é necessário compreender as funcionalidades oferecidas pelo módulo (com recurso à consulta do seu *datasheet* e demais documentos relevantes), de forma a possibilitar a programação do seu funcionamento de forma adequada. Para este projeto recomenda-se a aquisição e utilização de módulos nRF24L01+.

3.2. Camada de ligação de dados (Segunda camada)

Num sistema de comunicação, a camada de ligação de dados é responsável pela implementação destas duas importantes funcionalidades:

- **Controlo da Ligação Lógica (LLC – *Logical Link Control*).** Esta funcionalidade permite controlar a troca de informação na ligação física ao nível local, entre um emissor e um recetor, sendo por sua vez decomposta em outras subfuncionalidades, nomeadamente:
 - Delimitação das tramas, que consiste na identificação do início e fim de uma trama, no emissor e no recetor.
 - Controlo de erros e controlo de fluxo, de forma a evitar perdas de informação, através da implementação de mecanismos de deteção e correção de erros.
 - Endereçamento das estações envolvidas.
- **Controlo de Acesso ao Meio (MAC - *Medium Access Control*).** Num sistema de comunicações com múltiplas estações a transmitir dados no mesmo meio de comunicação (i.e., o meio é partilhado) também é necessário haver um conjunto de regras que determine o momento em que cada estação pode transmitir de modo a evitar colisões na transmissão das tramas.

Neste projeto deseja-se implementar uma rede local sem fios (WLAN – *Wireless Local Area Network*) que possibilite comunicação bidirecional e fiável. Para este efeito, **será necessário implementar as funcionalidades da camada 2** identificadas acima, nomeadamente o controlo da ligação lógica, sendo o controlo de acesso ao meio de implementação opcional. Os serviços implementados pela camada 2 serão disponibilizados às camadas superiores através de APIs (*Application Programming Interfaces*).

Recomenda-se a utilização de módulos Arduino (ou compatíveis) para desenvolvimento do código da camada 2 e ligação com os módulos *transceivers* de RF.

3.3. Camada de aplicação (Sétima camada) e respetiva aplicação

O modelo OSI propõe um conjunto de 7 camadas para um sistema de comunicação. Neste contexto, a camada 7 (camada de aplicação) implementa um protocolo de aplicação específico para um determinado domínio de aplicação (ex. HTTP, SMTP, FTP, etc.). Este protocolo define os serviços que estarão disponíveis para serem usados pelas aplicações.

As aplicações (programas), por sua vez, são desenvolvidas para utilizar os serviços da camada de aplicação. Neste sentido, quando é necessário garantir a interoperabilidade entre diferentes programas, estes devem fazer uso do mesmo protocolo de aplicação. Conforme referido anteriormente, neste projeto pretende-se desenvolver uma aplicação de *chat* que também tenha capacidade de transferência de ficheiros entre estações.

Cada estação da rede deverá ser composta por três componentes essenciais: um módulo *transceiver* de RF, um módulo Arduino e um PC. Neste contexto, o código da aplicação, incluindo o protocolo da camada 7, deverá ser desenvolvido no PC.

Em cada estação, a ligação transceiver-Arduino e a ligação Arduino-PC deverão utilizar interfaces digitais adequadas, como UART (*Universal Asynchronous Receiver/Transmitter*) ou SPI (*Serial Peripheral Interface*).

4. Fases do Projeto

O trabalho a desenvolver no âmbito do projeto está dividido em fases e tarefas, conforme detalhado a seguir. Cada fase abrange uma demonstração prática e a entrega de um relatório. A última fase abrange ainda uma apresentação oral.

O relatório de cada fase deverá apresentar uma estrutura coerente no que concerne à organização do conteúdo nas suas diferentes secções e subsecções. Neste sentido, convém ter atenção à informação sobre as tarefas apresentadas a seguir e às recomendações que serão fornecidas pelos docentes durante as aulas.

Fase 1

Tarefa 1.1 – Especificação do projeto

Objetivos:

- Definição da arquitetura e funcionalidades a serem oferecidas pelo sistema.
- Identificação das tecnologias a serem utilizadas e fazer uma estimativa dos recursos humanos que irão necessários (para o projeto inteiro).
- Discussão sobre as competências necessárias, indicando claramente o conjunto das que ainda não são dominadas na altura da escrita do documento.
- Planeamento temporal geral de todas as fases, tarefas e subtarefas através dum diagrama de Gantt.

Notas:

- A parte do relatório correspondente à tarefa 1.1 deverá abordar todos os tópicos referidos acima, em subsecções separadas.

Tarefa 1.2 – Comunicação via porta série PC-Arduino-Arduino-PC

Objetivos:

- Conhecer as características das interfaces de comunicação série entre os módulos Arduino e entre cada modulo Arduino e o respetivo PC.

- Conhecer e configurar os parâmetros de configuração da interface de comunicação série UART/RS-232 e os seus modos de funcionamento: normal/transparente (*raw*).
- Efetuar as ligações físicas necessárias entre os componentes utilizados.
- Implementar um programa no PC para transferência de um ficheiro de texto e de um ficheiro de imagem entre dois computadores ligados por cabo via porta série.
- Implementar um programa no módulo Arduino para encaminhamento dos dados recebidos do PC para o outro módulo Arduino e vice-versa, utilizando uma ligação por cabo (interface UART ou SPI).

Notas:

- O tamanho dos ficheiros a transferir é conhecido à partida, sendo *hard-coded* no programa (esta simplificação deverá ser eliminada no programa final).
- O programa desenvolvido não precisa detetar ou corrigir erros na transmissão (esta funcionalidade será implementada mais adiante, quando forem utilizados os *transceivers* de RF). No entanto, dado o uso de cabos e a curta distância entre os computadores, a comunicação deverá ser fiável (sem erros na comunicação de dados).
- O programa deverá ser testado com diferentes valores de *baud rate* da porta série, com o propósito de minimizar o tempo de transferência do ficheiro de imagem.

Fase 2

Tarefa 2.1 – Comunicação por radiofrequência Arduino-TRF-TRF-Arduino

Objetivos:

- Conhecer as características da interface de comunicação série entre cada *transceiver* de RF (TRF) e o respetivo módulo Arduino.
- Efetuar as ligações físicas necessárias entre os componentes utilizados.
- Conhecer e compreender o impacto no desempenho do sistema dos diferentes parâmetros configuráveis dos *transceivers* de RF.

- Desenvolvimento de código, nos módulos Arduino, para configuração e transmissão/receção de dados dos respetivos *transceivers* de RF.
- Conceção e execução de testes experimentais para avaliação de desempenho do sistema, abrangendo parâmetros como:
 - Atraso (*delay*) e/ou *round-trip time* (RTT);
 - Débito (*throughput*) máximo;
 - Alcance de transmissão (sem erros/perda de informação).

Notas:

- O *transceiver* de RF deve ser configurado para operar apenas com as funcionalidades de transmissão de dados a nível da camada física, ou seja, **os mecanismos disponibilizados pelo *transceiver* para deteção e correção de erros (através da retransmissão de pacotes) devem ser desabilitados**. Estes mecanismos deverão ser implementados na fase seguinte, no âmbito do desenvolvimento do protocolo da camada 2.
- Para evitar interferências nas comunicações, os *transceivers* dos diferentes grupos devem ser configurados para utilizar canais diferentes.
- Os resultados experimentais obtidos em cada fase devem ser comparados com os valores esperados calculados com base em análise teórica, tendo em consideração os parâmetros de configuração dos módulos e interfaces. **Qualquer discrepância observada entre valores teóricos e experimentais deverá ser rigorosamente analisada e discutida pelo grupo.**

Tarefa 2.2 – Controlo de acesso ao meio (opcional)

Objetivos:

- Perceber em que situações se torna necessário utilizar um protocolo de controlo de acesso ao meio (MAC).
- Analisar as vantagens e desvantagens dos diferentes tipos de protocolos de controlo de acesso ao meio para redes sem fios.
- Implementar sobre o código Arduino desenvolvido na fase anterior o(s) protocolo(s) de controlo de acesso ao meio escolhido(s) pelo grupo.

- Avaliar o desempenho do sistema com a transmissão simultânea de dados de dois dispositivos na mesma rede, comparando os resultados obtidos nos testes sem e com o protocolo MAC.

Notas:

- Para o teste do sistema no âmbito desta tarefa deverão ser utilizados dois emissores e um recetor. **Os emissores deverão gerar dados para transmissão em simultâneo e continuamente sem intervalos/atrasos.** O recetor deverá receber as tramas enviadas pelos dois emissores intercaladas, apresentando, para cada trama, o endereço do dispositivo e o número de sequência, que devem ser incluídos em cada trama enviada.

Fase 3

Tarefa 3.1 – Controlo da ligação lógica

Objetivos:

- Perceber a noção de trama/pacote, cabeçalho, *payload* e cauda;
- Perceber como representar corretamente a estrutura de um pacote (nome, tamanho e conteúdo de cada campo).
- Definir o protocolo de comunicação, especificando e concebendo as regras de comunicação, tramas e primitivas de serviço a oferecer à camada superior.
- Definir os tipos de trama e os campos que estas necessitam de conter para proporcionar as funcionalidades desejadas, no âmbito da camada 2 do projeto, para transmissão fiável de dados ponto-a-ponto, com ênfase na deteção e correção de erros.
- Disponibilizar as primitivas às camadas superiores através de APIs.
- Efetuar a transferência fiável de um fluxo de dados entre dois módulos Arduino sobre a ligação RF sem fios.

Notas:

- A camada 2 implementada deve garantir a transmissão fiável dos dados, e em caso de erros de transmissão, quer sejam erros devido à corrupção de bits na trama ou perda de bytes da trama ou perda total de tramas.

- A camada 2 implementada deve ser capaz de efetuar *timeout* caso haja perda de tramas ou um atraso superior a determinado *threshold*, e recuperar a transmissão da trama.
- Deverão ser feitos novos testes para avaliação do desempenho em termos de RTT, atraso, débito máximo e PER (*Packet Error Rate*) utilizando o protocolo implementado nesta fase. Os resultados obtidos devem ser comparados com os obtidos na tarefa 2.1.

Fase 4

Tarefa 4.1 – Camada de aplicação e respetiva aplicação

Objetivos:

- Implementar o protocolo de nível 7 (camada de aplicação).
- Criar um programa de conversação e transferência de ficheiros que utilize o protocolo desenvolvido.
- Criar uma interface gráfica para a aplicação (opcional).

Notas:

- A comunicação entre a camada 2 e o restante código deverá ser feito através de funções bem conhecidas, que formam a sua API.
- Durante o desenvolvimento do projeto deverá ser seguido o princípio da independência entre camadas. Por exemplo, a camada 2 não deve ter conhecimento de que a informação que está a transportar pertence (ou não) a um ficheiro. Por outro lado, a camada 7 não deve ter conhecimento sobre as tramas de controlo de erros (e.g., ACK - *acknowledgment*) trocadas no interior da camada 2.
- O grupo deve ter em atenção e procurar aprimorar o desempenho do sistema ao longo de todas as fases, tendo em consideração os parâmetros referidos anteriormente.

Tarefa 4.2 – Integração e testes finais

Objetivos:

- Integrar as funcionalidades desenvolvidas em todas as camadas ao longo do projeto.
- Executar os testes experimentais finais, analisando e discutindo os resultados obtidos (tal como também deve ser feito nas tarefas anteriores).
- Otimizar o desempenho do sistema, reduzindo o tempo de transferência dos ficheiros ao mínimo possível.
- Apresentar as conclusões finais referentes ao desenvolvimento do projeto como um todo.

5. Avaliação

A avaliação da UC será individual e terá em conta a contribuição de cada aluno para o seu grupo de trabalho.

Os elementos de avaliação de cada uma das quatro fases incluem uma demonstração prática e um relatório. Todos os relatórios deverão incluir no final uma secção obrigatória de auto-avaliação/reflexão. Esta secção deverá descrever (num parágrafo para cada aluno) as contribuições efetivas de cada elemento do grupo para o desenvolvimento da respetiva fase do projeto. Não deve ser indicada a nota esperada pelo aluno ou grupo.

Na última fase há lugar ainda para uma apresentação oral (cerca de 15 minutos) acompanhada dum ficheiro Powerpoint, que deverá fornecer uma visão global e objetiva do sistema desenvolvido, incluindo a descrição de todas as partes que compõem o sistema final. Neste sentido, a apresentação não deve incluir conteúdo sobre o planeamento do projeto, mas sim sobre a implementação e testes. Também não deve organizada de forma cronológica nem deve focar em implementações falhadas anteriores, mas sim na solução otimizada final, embora possam ser referidos brevemente os problemas encontrados e a sua resolução. A apresentação deverá incluir os principais resultados da análise de desempenho, bem como um *slide* final com as principais conclusões técnicas sobre o sistema implementado no projeto.

A UC inclui ainda um elemento de avaliação contínua, que terá em consideração parâmetros como a assiduidade (faltas), pontualidade (atrasos), participação, conhecimentos demonstrados, empenho e comportamento nas aulas.

Os pesos dos elementos de avaliação são indicados abaixo:

Fase 1 – Demonstração prática e relatório (20%)

Tarefa 1.1 – Especificação do projeto

Tarefa 1.2 – Comunicação via porta série PC-Arduino-Arduino-PC

Fase 2 – Demonstração prática e relatório (20%)

Tarefa 2.1 – Comunicação por radiofrequência Arduino-TRF-TRF-Arduino

Tarefa 2.2 – Controlo de acesso ao meio (opcional)

Fase 3 – Demonstração prática e relatório (20%)

Tarefa 3.1 – Controlo da ligação lógica

Fase 4 – Apresentação oral, demonstração prática e relatório (30%)

Tarefa 4.1 – Camada de aplicação e respetiva aplicação

Tarefa 4.2 – Integração e testes finais

Avaliação contínua (10%)

O calendário com as datas das avaliações é disponibilizado num ficheiro à parte, na plataforma Blackboard.

6. Bibliografia

- Andrew S. Tanenbaum, “Computer Networks”, Prentice Hall, 2010.
- Jochen Schiller, “Mobile Communications”, Addison-Wesley, 2004.
- William Stallings, “Wireless Communications and Networks”, 2005.
- Outros documentos indicados pelos docentes e pesquisados pelos alunos no decorrer das aulas.