

## 1 Control - Componente de Software

A componente de *Control*, desenvolvida em Kotlin, implementa a lógica de alto nível do sistema de jogo da roleta, interagindo com os periféricos (teclado, LCD, display da roleta) através de módulos especializados. Esta arquitetura modular e bem separada permite manter o código coeso, testável e facilmente extensível. Para esta fase do projeto, este componente engloba os ficheiros da figura 1.

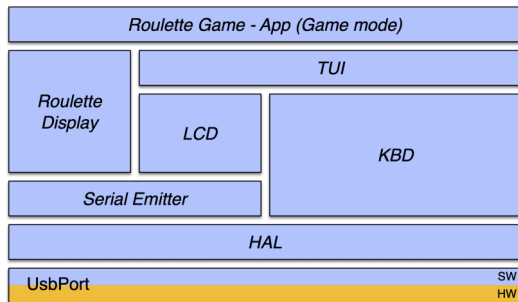


Figura 1: Control module

### 1.1 UsbPort

O módulo **UsbPort** é a base da comunicação entre o sistema de software e o hardware. Representa uma abstração das portas de entrada e saída da interface USB simulada, permitindo manipular diretamente os bits definidos pelo ficheiro `hardware.simul`.

Este módulo define as máscaras associadas aos sinais de controlo e dados utilizados pelos periféricos, como o teclado e o LCD.

### 1.2 HAL - Hardware Abstraction Layer

Sobre o **UsbPort** constrói-se o módulo **HAL**, que fornece um conjunto de funções de mais alto nível para interação com o hardware.

Entre as principais funções destacam-se:

- `init()` — Inicializa o HAL com os estados definidos;
- `readBits(mask)`,
- `writeBits(mask, value)`;

- `setBits(mask)`,
- `clrBits(mask)`,
- `isBit(mask)`.

Este módulo assegura a leitura e escrita nos bits corretos do sistema, escondendo os detalhes do acesso direto às portas USB simuladas.

### 1.3 KBD - Interface do Teclado

O módulo **KBD** representa a interface lógica do teclado 4x4, fazendo uso do HAL para identificar os sinais  $K_{val}$  e  $K_{3:0}$ . Através da correspondência dos códigos recebidos com caracteres, permite obter a tecla premida.

Funções principais:

- `getKey()` — Retorna o último valor lido (ou NONE);
- `waitKey(timeout)` — Espera por uma tecla com limite de tempo.

São suportadas as teclas de '0' a '9', 'A' a 'D', '\*' e '#'.

### 1.4 SerialEmitter

O **SerialEmitter** é o módulo responsável por enviar tramas codificadas através da interface série, usadas para comunicar com os módulos SLCDC (LCD) e SRC (roleta).

Cada trama contém:

- Bits de dados (4 ou 8 bits conforme destino);
- Bit RS ou comando (LCD ou Roleta);
- Bit de paridade ímpar.

As funções principais incluem:

- `send(destination, data, size)` — envia a trama codificada com 1 bit de paridade ímpar;
- `init()` — inicializa ambos os destinos e limpa a linha de clk;
- `pulseClock()` - impõe um ciclo de SCLK clock;

## 1.5 LCD

Este módulo fornece funções de alto nível para interação com o display LCD. Usa o `SerialEmitter` (modo série) ou interface paralela para enviar comandos ou dados ao LCD.

Funções principais:

- `init()` — Sequência de inicialização do LCD;
- `write(Char)`, `write(String)`
- `writeNibbleSerial`, `writeNibbleParallel`;
- `cursor(line, column)`, `clear()`.

As instruções são convertidas internamente para tramas apropriadas, com distinção entre comandos (RS=0) e dados (RS=1).

## 1.6 RouletteDisplay

O módulo `RouletteDisplay` é responsável por controlar o display da roleta, através de tramas enviadas ao componente SRC.

Comandos possíveis:

- `setValue(value: Int)` — apresenta um valor;
- `animation()` — inicia a animação do sorteio;
- `off(enabled: Boolean)` — ativa ou desativa o display.

As tramas seguem o formato de 8 bits (3 de comando, 5 de dados).

## 1.7 TUI - Text User Interface

O objetivo do TUI é permitir abstrair a interação com o utilizador final, através do LCD e teclado. Para isso ele deve fornecer utilitários para apresentar menus, pedir inputs, mostrar mensagens, limpar o ecrã, entre outros.

Este módulo serve de ponte entre a lógica do jogo e os periféricos visuais e de input.

No entanto, para esta fase do projeto, o TUI apenas oferece abstração a nível de escrita automática das mensagens no LCD.

## 1.8 APP - Game Mode (Lógica do Jogo)

O módulo principal do programa implementa dois modos distintos: **Modo de Jogo** e **Modo de Manutenção**.

### Modo Jogo

- Iniciado com a tecla \*, se houver créditos;
- Permite apostas com as teclas 0-9, A-D;
- Finaliza com a tecla #, executa a animação e sorteia;
- Compara apostas com resultado e atualiza créditos.

### Modo Manutenção

Através de sequências específicas:

- Tecla A — ver contadores;
- A + \* — reset aos contadores;
- Tecla C — ver estatísticas;
- C + \* — limpar estatísticas;
- Tecla D — saída do modo manutenção e persistência dos dados.

## 1.9 Conclusão

A componente de software está organizada de forma hierárquica e modular, com clara separação entre acesso ao hardware, controlo de periféricos e lógica de aplicação. Esta estrutura facilita o desenvolvimento, manutenção e testes, garantindo a robustez e extensibilidade do sistema da roleta.