Especificação de Firmware

# Objetivo

Este documento tem como objetivo descrever detalhadamente as funcionalidades que o firmware deverá exercer. Este documento representa a primeira proposta de arquitetura para implementação para o firmware dos produtos.

Sumário

[Objetivo 1](#_Toc481964295)

[Descrição geral 2](#_Toc481964296)

[Desacoplamento da plataforma física 2](#_Toc481964297)

[Desacoplamento entre módulos 2](#_Toc481964298)

[Portabilidade 2](#_Toc481964299)

[Atualização 2](#_Toc481964300)

[Camadas do Firmware 3](#_Toc481964301)

[Camada plataforma 4](#_Toc481964302)

[Camada adaptadora 4](#_Toc481964303)

[Camada funcional 5](#_Toc481964304)

[Camada aplicação 5](#_Toc481964305)

# Descrição geral

O firmware será todo o código fonte que um microcontrolador deverá executar para desempenhar as ações afim de providenciar as funcionalidades especificadas para cada produto. Será separado em conjuntos chamados de “camadas” com propósito para proporcionar ao projeto as características descritas a seguir.

## Desacoplamento da plataforma física

Tornar o quanto for possível do código desacoplado da plataforma física que irá executa-lo. Com isto o projeto conseguirá maior flexibilidade e consequentemente independência de flutuações no mercado de semicondutores, não dependendo da disponibilidade de um único microcontrolador e fornecedor.

## Desacoplamento entre módulos

Tornar trechos de código com funcionalidades distintas independentes entre si. Minimizar o referenciamento de código, e documentar detalhadamente a arvore de referências de um dado conjunto de código.

## Portabilidade

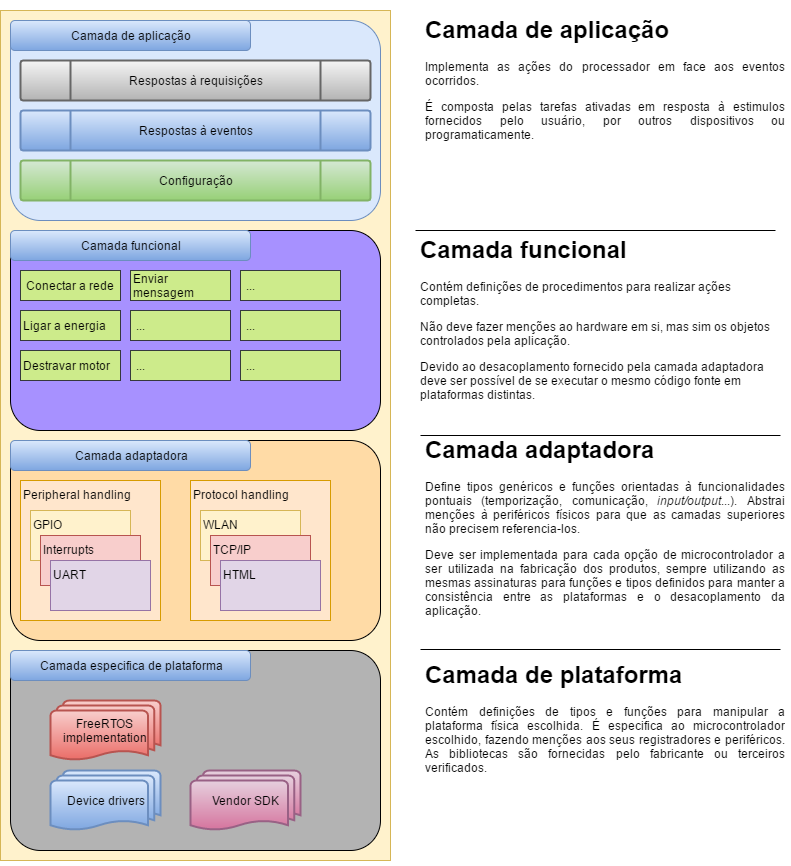
A partir do desacoplamento e da fragmentação do código minimiza-se a quantidade de esforço necessária para trazer o projeto para uma nova plataforma, seja qual for o motivo para tal migração – custo, *end of life* de dispositivo, entre outros.

## Atualização

Com a separação do código a atualização de trechos específicos há de ser transparente para o restante do código, facilitando a manutenção e aumentando a vida útil do firmware.

# Camadas do Firmware

O firmware será dividido em quatro camadas de projeto. São elas **aplicação**, **funcional**, **adaptadora** e **plataforma**. O infográfico abaixo ilustra e descreve brevemente as camadas. O nível de abstração decresce de cima para baixo, sendo a camada de plataforma a de mais baixo nível, lidando diretamente com os registradores do microcontrolador utilizado, enquanto a camada de aplicação implementa tarefas e ações utilizando-se das camadas abaixo afim de atingir os objetivos impostos.



## Camada plataforma

Esta camada consiste no agrupamento das bibliotecas e pacotes de código que manipulam diretamente os registradores do microcontrolador utilizado.

Contém *device drivers* do dispositivo, capazes de operar seus periféricos como por exemplo configurar as portas de *GPIO*, solicitar uma leitura do conversor analógico-digital ou ligar uma interrupção.

Contém também a implementação de protocolos já estabelecidos comercialmente que serão utilizados por camadas superiores, como por exemplo a implementação das rotinas necessárias para uma comunicação TCP/IP, rotinas para criptografia de dados ou para a manipulação de um sistema de arquivos.

Por fim, também deve conter a implementação de um RTOS (usualmente FreeRTOS) específica para a plataforma em questão. Deve implementar a designação de funções a serem tratadas como tarefa, um escalonador para chavear os contextos de cada tarefa e mecanismos que compartilhamento de recursos padrões (semáforos, *mutexes*, e *flags*).

O código que constitui esta camada é fornecido por terceiros ao projeto, seja o fabricante do dispositivo em questão ou pessoa física ou jurídica de credibilidade verificada. Deve estar acompanhado de documentação sucinta especificando as entradas, saídas e efeitos de cada trecho de código fornecido.

Os códigos contidos nesta camada são específicos ao microcontrolador ao qual fazem referência, não podendo ser reaproveitados em uma plataforma diferente.

## Camada adaptadora

É a camada padronizadora do projeto. Seu objetivo é tomar as definições contidas na camada inferior (de plataforma) e torna-las genéricas, abstraindo especificidades de fabricantes e de bibliotecas como por exemplo nomes de tipos e assinaturas de funções, afim de que as camadas superiores (funcional e aplicação) não precisem saber dos detalhes de implementação física da plataforma utilizada.

Conterá um mapeamento, onde os tipos e assinaturas de funções que as camadas superiores conhecerão serão mapeados para um ou para um conjunto de objetos da camada de plataforma afim de se atingir um objetivo especifico.

Por exemplo: a camada funcional necessita enviar dados pela porta serial do microcontrolador. Para isto, ela precisará configurar a porta serial e passa-la tais dados através de funções. A camada adaptadora deverá de fornecer tais funções com uma assinatura genérica, com implementação interior realizando as chamadas necessárias fornecidas pela plataforma para configurar o periférico especifico e manuseá-lo. Assim as camadas superiores precisarão conhecer somente as assinaturas genéricas da camada adaptadora, ignorando a implementação.

Como os códigos da camada adaptadora fazem referência aos contidos na camada de plataforma a camada adaptadora precisará ser implementada tantas vezes quanto o número de microcontroladores dos quais o produto final poderá fazer uso, podendo ser tanto parcial – como por exemplo a adequação da camada à um novo microcontrolador da mesma fabricante, de família similar porém com funcionalidades mais avançadas – quanto completa – como por exemplo a um novo microcontrolador de fabricante diferente, cuja arquitetura não possui nada em comum com a usual.

## Camada funcional

Esta camada deve de ser o conjunto de ferramentas com as quais a aplicação realizará suas ações. Deve fornecer definições de funções com efeitos completos.

Sua nomenclatura deve evitar mencionar componentes de hardware, favorecendo a evidenciação da funcionalidade implementada pela função.

*Continua...*

## Camada aplicação

...