Capítulo 4

Introdução aos microcontroladores STM32

4.1. Introdução aos microcontroladores

O desenvolvimento da tecnologia de circuitos integrados miniaturizados permitiu que pudéssemos instalar centenas de milhares de transistores em um único chip, que foi a precondição para a possibilidade de fabricação dos microprocessadores. Os primeiros computadores digitais foram desenvolvidos a partir do advento dos microprocessadores e da adição de circuitos periféricos externos, tais como memórias, linhas de entrada e saída, linhas de comunicação, temporizadores, conversores, entre outros. Com o aumento da densidade de elementos dentro dos chips, tornou-se possível o desenvolvimento de circuitos integrados que continham tanto o processador como também os periféricos, que foram chamados, mais tarde, de microcontroladores.

O microcontrolador é um circuito integrado programável que contém todos os componentes de um computador (*CPU*, memória, portas de entrada e saída, conversores A/D e D/A, etc). O microcontrolador foi projetado para ter todos os circuitos essenciais ao seu funcionamento em um único chip. Isso economiza tempo e espaço necessários para projetar um dispositivo microprocessado. Sua aplicação vai desde um simples controle remoto a máquinas mais complexas como, por exemplo, máquinas pneumáticas e hidráulicas, máquinas dispensadoras de produtos, controladores de velocidade de motores, temporizadores, sistemas automotivos, sistemas de controle, robôs industriais, sistemas de telefonia, equipamentos médico-hospitalares, entre outros.

Por outro lado, um microprocessador, mesmo sendo considerado uma poderosa máquina de computação, não é capaz, por si só, de se comunicar com o ambiente externo, sendo necessária a adição de circuitos especiais, chamados periféricos, tais como sistema de entrada e saída de dados, vias de comunicação e memória externa, apenas para citar alguns exemplos.

Existem diversas linhas de microcontroladores e entre os principais fabricantes podemos citar a Atmel, Microchip, Texas Instruments, ST Microelectronics, NXP, Renesas, Intel, Motorola, Hitachi, Zilog, entre outros. O Arduino Uno, por exemplo, mostrado na figura 1, na realidade consiste em uma placa de prototipagem eletrônica que utiliza um microcontrolador AVR da Atmel. O arduíno se popularizou por sua simplicidade e baixo custo, mas não deve ser confundido com um microcontrolador, pois essa palavra se refere a um circuito integrado.

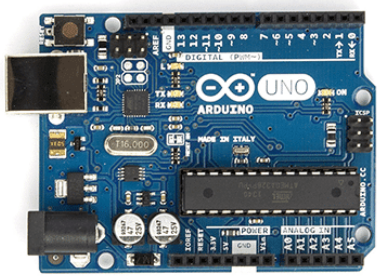

Figura 1 – Placa de prototipagem Arduino Uno.

Na figura 2 podem ser vistos alguns circuitos periféricos que são integrados no chip de um microcontrolador de uso geral. O microcontrolador é formado basicamente pelo microprocessador, responsável pelo controle de todo o funcionamento do chip e execução das instruções, e dos circuitos periféricos como memórias, conversores A/D, circuito oscilador, entre outros.

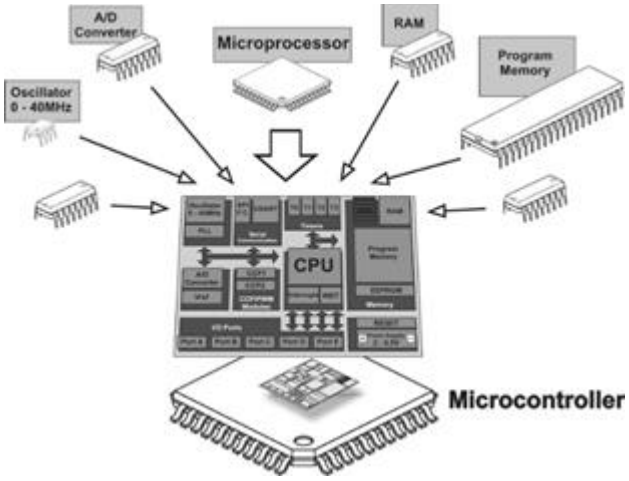

Figura 2 – Elementos construtivos de um microcontrolador de uso geral.

Existem muitas coisas diferentes dentro de um microcontrolador. Em muitos microcontroladores, o processador ocupa menos de 10% da área do chip de silício e o restante é ocupado por outros componentes, como:

- Memória de programa (por exemplo, memória flash)
- SRAM
- Periféricos
- Infraestrutura de barramentos internos
- Gerador de clock, gerador de reset e rede de distribuição para esses sinais
- Regulador de tensão e circuitos de controle de energia
- Outros componentes analógicos (por exemplo, conversores AD, DA, circuitos de referência de tensão, comparadores,
- Portas de entrada e saída
- Circuitos de suporte para testes de fabricação.
- Circuitos de suporte à depuração.

Embora alguns desses componentes sejam diretamente visíveis para os programadores, alguns outros podem ser invisíveis para os desenvolvedores de software (por exemplo, circuito de suporte para testes de fabricação).

Como os periféricos de diferentes fornecedores de microcontroladores são diferentes, você precisa baixar e ler os manuais do usuário (ou documentos semelhantes) dos fornecedores de microcontroladores. Periféricos e registradores de controle para gerenciamento do sistema são acessíveis a partir do mapa de memória. Para tornar mais fácil para os desenvolvedores de software, a maioria dos fornecedores de microcontroladores fornece arquivos de cabeçalho C e bibliotecas de driver para seus microcontroladores. Na maioria dos casos, esses arquivos são desenvolvidos com o *Cortex Microcontroller Software Interface Standard* (CMSIS), o que significa que ele usou um conjunto de arquivos de cabeçalhos padronizados para acessar os recursos do processador.

Na maioria dos casos, o processador faz todo o trabalho de controlar os periféricos e lida com o gerenciamento do sistema. Em alguns microcontroladores, também existem alguns periféricos inteligentes que podem fazer pequenas quantidades de processamento sem a intervenção do processador. Isso depende dos periféricos específicos do fornecedor nos microcontroladores.

Os microcontroladores oferecem uma ampla gama de aplicações. É critério do projetista decidir o que ele quer que o microcontrolador faça na sua aplicação específica. O funcionamento da aplicação deve ser exaustivamente testado a fim de detectar problemas (ou *bugs*) de software (oriundos de instruções mal planejadas) e de hardware (oriundos de circuitos defeituosos). Se houver necessidade de alterações, melhorias ou atualizações no comportamento da aplicação, basta fazê-lo até que se sinta satisfeito.

A rigor, um projeto microcontrolado é executado obedecendo os seguintes passos:

1. Estudo do dispositivo ou aplicação, também chamada de "planta", a ser controlado pelo microcontrolador;
2. Identificação das necessidades de hardware, tais como o número de entradas/saídas digitais e analógicas, temporizadores, conversores A/D e D/A, etc., baseada nas características de funcionamento da aplicação a ser controlada.
3. Escolha de um microcontrolador que possua os requisitos identificados e que satisfaça às suas necessidades, levando-se em consideração o compromisso entre custo e benefício.

4. Desenvolvimento de um programa, escrito em uma linguagem de programação, geralmente Assembly ou C/C++, contendo as instruções que serão executadas durante o funcionamento do microcontrolador.
5. Realização de simulações, concomitantemente à etapa de programação, para testar o funcionamento da aplicação.

O programa escrito deve ser convertido em código de máquina e descarregado dentro da memória de programa do microcontrolador com o auxílio de um hardware programador ou gravador. Uma vez que o microcontrolador possua em sua memória as instruções a serem executadas, basta instalá-lo no dispositivo a ser controlado.

4.2. Microcontroladores STM32

STM32 é uma família de microcontroladores de 32 bits produzida pela *ST Microelectronics*, baseada no processador ARM Cortex-M. Oferece produtos que combinam alto desempenho, recursos em tempo real, processamento digital de sinais, operação de baixa potência/baixa tensão e conectividade, mantendo total integração e facilidade de desenvolvimento. Internamente, cada microcontrolador incorpora um núcleo de processamento ARM Cortex-M, RAM estática, memória flash, interface de depuração e vários outros periféricos que distinguem vários produtos.

A gama de microcontroladores STM32, tendo como base um núcleo padrão da indústria, vem com uma vasta escolha de ferramentas e softwares para dar suporte ao desenvolvimento de projetos de ponta a ponta, tornando essa família de produtos ideal para pequenos projetos, bem como grandes plataformas.

A família de microcontroladores STM32 oferece uma estrutura para criação de uma vasta quantidade de sistemas embarcados, desde simples *dongles* até sistemas complexos de tempo real, como drones auto pilotados. Esta família de componentes inclui dezenas de configurações diferentes, fornecendo amplas opções de tamanho de memória, periféricos, performance e consumo de energia. Os componentes são razoavelmente baratos em pequenas quantidades e justificam seu uso em aplicações de baixo volume de produção. Na verdade, os componentes iniciais são comparáveis em valores aos componentes da família ATmega, usados nas placas de desenvolvimentos Arduíno, e ainda apresentam melhoras significativas de desempenho e periféricos mais sofisticados. Além disso, os periféricos utilizados são compartilhados por outros membros da família (por exemplo, os periféricos de comunicação USART são compatíveis com todos os componentes da família STM32) e são suportados por um único firmware. Portanto, aprender como programar um único dispositivo da família STM32 o capacita a programar praticamente todos eles.

Existem atualmente várias linhas de microcontroladores STM32: STM32F0, STM32F1, STM32F2, STM32F3, STM32F4, STM32F7, STM32L0, STM32L1, STM32L4, STM32L4+, STM32L5, STM32G0, STM32G4, STM32H7, STM32WB e STM32WL, suportadas por diferentes, mas estruturalmente semelhantes, bibliotecas de firmware.

Embora estas famílias compartilhem vários periféricos, algum cuidado deve ser tomado quando for migrar os projetos entre estas famílias. Cada uma dessas linhas tem aplicações específicas, devendo o projetista escolher a melhor opção para sua aplicação.

O STM32 é uma linha de produtos bastante complexa, abrangendo mais de dez subfamílias de produtos. Os produtos podem ser agregados em quatro grupos: *MCUs* de alto desempenho, convencionais *(mainstream)*, *wireless* e de ultra baixa potência.

Microcontroladores de alto desempenho são aqueles dedicados a aplicações de uso intensivo de CPU e multimídia. São *MCUs* baseados em Cortex-M3/M4F/M7, H7, com frequências máximas de clock que variam de 120MHz (F2) a 400MHz (H7). Todos os *MCUs* deste grupo fornecem o *ART Accelerator*, uma tecnologia de hardware da *ST Microelectronics* que permite a execução de código sem estados de espera pela memória flash.

Os *MCUs* convencionais são desenvolvidos para aplicações sensíveis ao custo, onde o custo do *MCU* deve ser o mais baixo possível e o espaço é uma forte restrição. Neste grupo, podemos encontrar *MCUs* baseados no Cortex-M0/M3, com frequências máximas de clock variando de 48MHz (F0) a 72MHz (F1/F3).

Os *MCUs wireless* formam uma nova linha de microcontroladores STM32 de núcleo duplo, com rádio de 2,4 GHz integrado, adequado para aplicações sem fio e Bluetooth. Esses *MCUs* apresentam um núcleo Cortex-M0+ (chamado processador de rede) dedicado ao gerenciamento de rádio e um núcleo Cortex-M4 programável pelo usuário (chamado processador de aplicativos).

O grupo de ultra baixa potência contém as famílias de *MCUs* para aplicações de baixo consumo de energia, usadas em dispositivos alimentados por bateria que precisam reduzir o consumo total de energia a níveis baixos, garantindo maior autonomia e vida útil da bateria. Neste grupo, podemos encontrar os *MCUs* baseados no Cortex-M0+, para aplicações sensíveis ao custo, e os microcontroladores baseados no Cortex-M4F com *Dynamic Voltage Scaling (DVS)*, uma tecnologia que permite otimizar a tensão interna da CPU de acordo com sua frequência.

Na Figura 2 é mostrado o portfólio de produtos da família STM32.

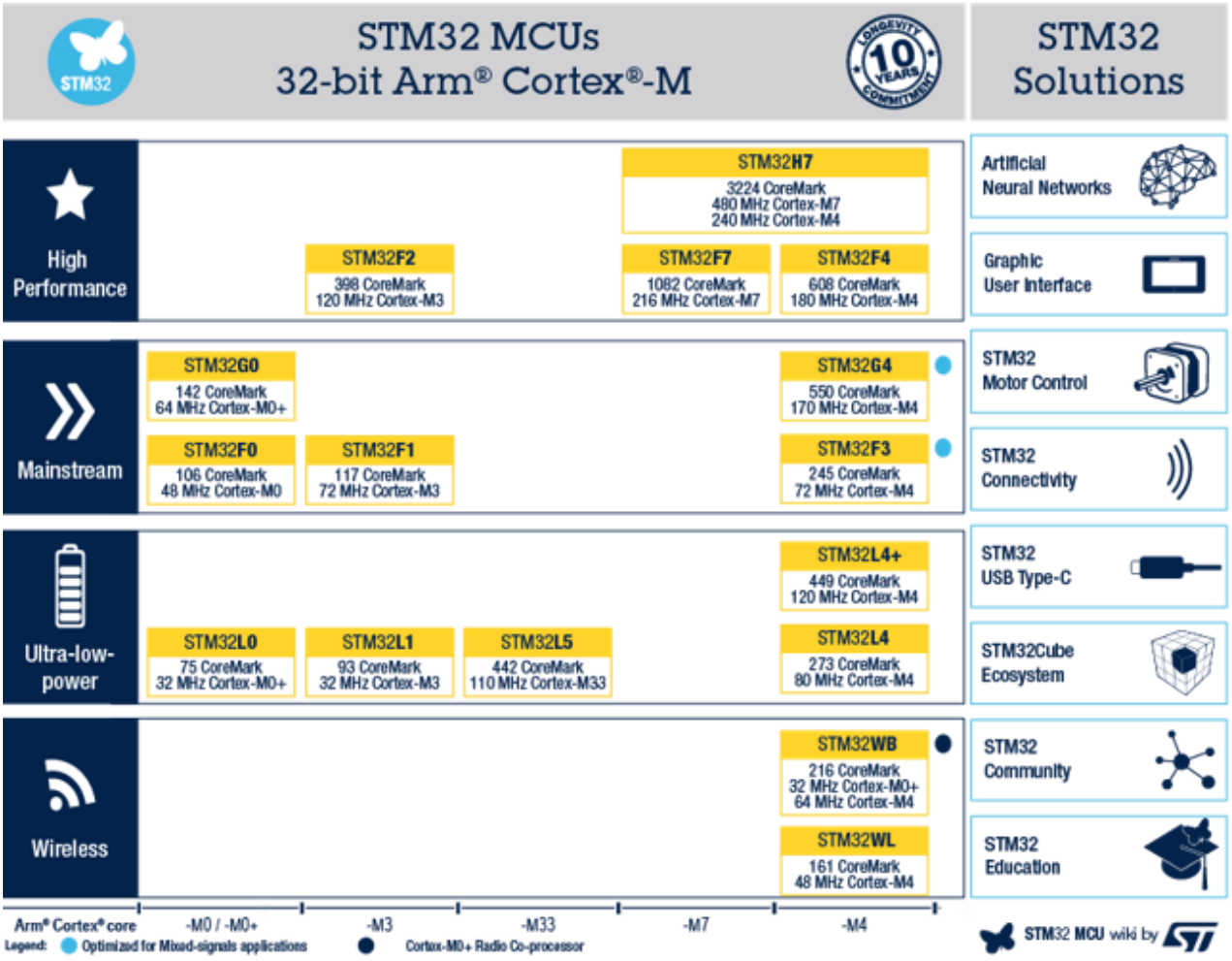

Figura 2 – Portfólio de produtos da família STM32 e suas aplicações.

Infelizmente, poder de processamento e flexibilidade são obtidos com um custo: o desenvolvimento de softwares para a família STM32 pode ser extremamente desafiador para um iniciante, com uma vasta documentação e bibliotecas para vasculhar. Por exemplo, o *Reference Manual RM0041*, manual de referência para uma grande quantidade de dispositivos STM32, possui 675 páginas e sequer cobre o núcleo de processamento ARM Cortex-M3. Por outro lado, não é necessário ler todo este manual para começar a desenvolver aplicações para o STM32, embora seja uma referência importante. Além disso, um iniciante se depara com um vasto conjunto de opções de ferramentas de desenvolvimento (*tool-chain*).

Em contraste, a plataforma Arduino oferece bibliotecas para aplicações simples e ambiente de programação acessível a programadores inexperientes. Para muitos sistemas simples, oferece um rápido atalho para construir protótipos. No entanto, simplicidade traz custos: As plataformas de software do Arduino não se adaptam bem ao gerenciamento de atividades concorrentes em sistemas complexos de tempo real e, para interação com softwares externos, depende de bibliotecas projetadas fora do ambiente de programação Arduino, utilizando técnicas e ferramentas semelhantes às utilizadas para o STM32. Além do mais, a plataforma Arduíno não provê capacidade de depuração, o que limita severamente o desenvolvimento de sistemas mais complexos. Novamente, a depuração requer a saída do ambiente confinado da plataforma Arduino. Finalmente, o ambiente Arduino não suporta sistemas operacionais de tempo real (*RTOS – Real Time Operating System*), essenciais para a construção de sistemas embarcados mais complexos.

Para experientes em programação com linguagem C, a família STM32 se mostra como uma plataforma muito superior à Arduino para o desenvolvimento de sistemas microcontrolados se essas dificuldades iniciais puderem ser superadas.

4.2. Vantagens dos microcontroladores STM32

A plataforma STM32 oferece várias vantagens para desenvolvedores de sistemas embarcados:

■ *MCUs* baseados no processador ARM Cortex-M, o que garante várias ferramentas de desenvolvimento disponíveis no mercado. ARM tornou-se um padrão no mundo dos sistemas embarcados.
■ Ferramentas baseada em ARM, possibilitando trabalhar com cadeias de ferramentas totalmente gratuitas, o que é extremamente importante para os hobbystas e estudantes;
■ Reutilização de *know-how*: STM32 é um portfólio bastante extenso, baseado em um denominador comum: Sua *CPU*. Isso garante que a experiência adquirida em uma determinada *CPU* STM32 possa ser aplicada facilmente a outros dispositivos da mesma família. Além disso, trabalhar com processadores Cortex-M permite reutilizar grande parte das habilidades adquiridas ao se decidir mudar para os *MCUs* Cortex-M de outros fabricantes.
■ Compatibilidade pino a pino com a maioria dos *MCUs* STM32.
■ A maioria dos pinos do STM32 é tolerante a 5V. Isso significa que você pode fazer interface com outros dispositivos que não fornecem entrada/saída de 3,3V, sem precisar usar deslocadores de nível;
■ STM32 é a escolha certa se você deseja migrar de *MCUs* de 8/16 bits para uma plataforma com maior poder computacional, com possibilidade de utilização de RTOS para aprimorar aplicações;
■ *MCUs* STM32 possuem um *bootloader* integrado que permite reprogramar a memória flash interna usando alguns periféricos de comunicação (USART, I^2C, etc.).

4.3. Introdução ao STM32F407

O microcontrolador STM32F407 é um microcontrolador baseado no núcleo RISC de 32 bits ARM Cortex-M4 que opera a uma frequência de até 168 MHz (para comparação, o Arduino Uno opera a 16 MHz, o Arduino Mega opera a 20 MHz e o Arduino Due opera a 84 MHz). O núcleo embarcado do Cortex-M4 possui uma unidade aritmética de ponto flutuante (*FPU*) com precisão simples. Também implementa um conjunto de instruções de processamento digital de sinais (*Digital Signal Processing-DSP)* e uma unidade de proteção de memória (*Memory Protect Unit - MPU*) que pode aprimorar a segurança das aplicações.

O STM32F407 incorpora memórias de alta velocidade (memória Flash de até 1 MB e até 192 KB de SRAM) e uma ampla variedade de entradas, saídas e periféricos aprimorados, interconectados por vários barramentos de 32 bits de alta velocidade que garantem a operação em paralelo de vários periféricos.

O microcontrolador oferece ainda três conversores Analógico/Digital (*Analog to Digital Converter – ADC*) de 12 bits, dois conversores Digital/Analógico (*Digital to Analog Converter – DAC*), também de 12 bits, um relógio de tempo real (*Real Time Clock – RTC*) de baixa potência, doze temporizadores de uso geral de 16 bits, incluindo dois temporizadores PWM para controle de velocidade de motores, dois temporizadores de 32 bits de uso geral, um gerador de números aleatórios (*Random Number Generator - RNG*), além de várias interfaces de comunicação padrão e avançadas, como UART, USART, SPI, I2C, I2S, CAN, SDIO/MMC, USB e Ethernet. O STM32F407 opera na faixa de temperatura de –40 a +105 °C, requer uma fonte de alimentação de 1,8 a 3,6 V (3,3V é a tensão de alimentação típica). Um conjunto abrangente de modos de economia de energia permite o desenvolvimento de aplicações de baixo consumo. Todos esses recursos tornam o microcontrolador STM32F407 adequado para uma ampla gama de aplicações como:

• Acionamento de motores e aplicações de controle;
• Equipamentos médicos/hospitalares;
• Aplicações industriais: CLP, inversores, disjuntores;
• Impressoras e scanners;
• Sistemas de alarme, vídeo porteiro e *HVAC - Heating, Ventilating and Air Conditioning*
• Aparelhos de áudio domésticos, entre outros.

Um diagrama de blocos da estrutura interna do STM32F407 é mostrado na Figura 3, na página seguinte. A *CPU* Cortex-M4F é mostrada no canto superior esquerdo, com uma frequência máxima de 168 MHz. A maior parte da figura mostra os periféricos e suas interligações. Há três barramentos de alta velocidade (AHB1/2/3) e dois barramentos de alta velocidade para comunicação com periféricos (APB1/2).

ART Accelerator

O *ART Accelerator* é um acelerador de memória otimizado. Ele equilibra a vantagem de desempenho do núcleo ARM Cortex-M4 com as tecnologias de memória Flash, que normalmente exigem que o processador aguarde pela resposta da memória quando está operando em frequências mais altas. Para aproveitar o desempenho completo do processador nessas frequências, o acelerador implementa uma fila de pré-busca de instruções e cache de desvios, o que aumenta a velocidade de execução do programa a partir do barramento de memória Flash de 128 bits. O desempenho alcançado, graças ao acelerador, é equivalente a não ter estados de espera na execução da memória Flash a uma frequência de *CPU* de até 168 MHz.

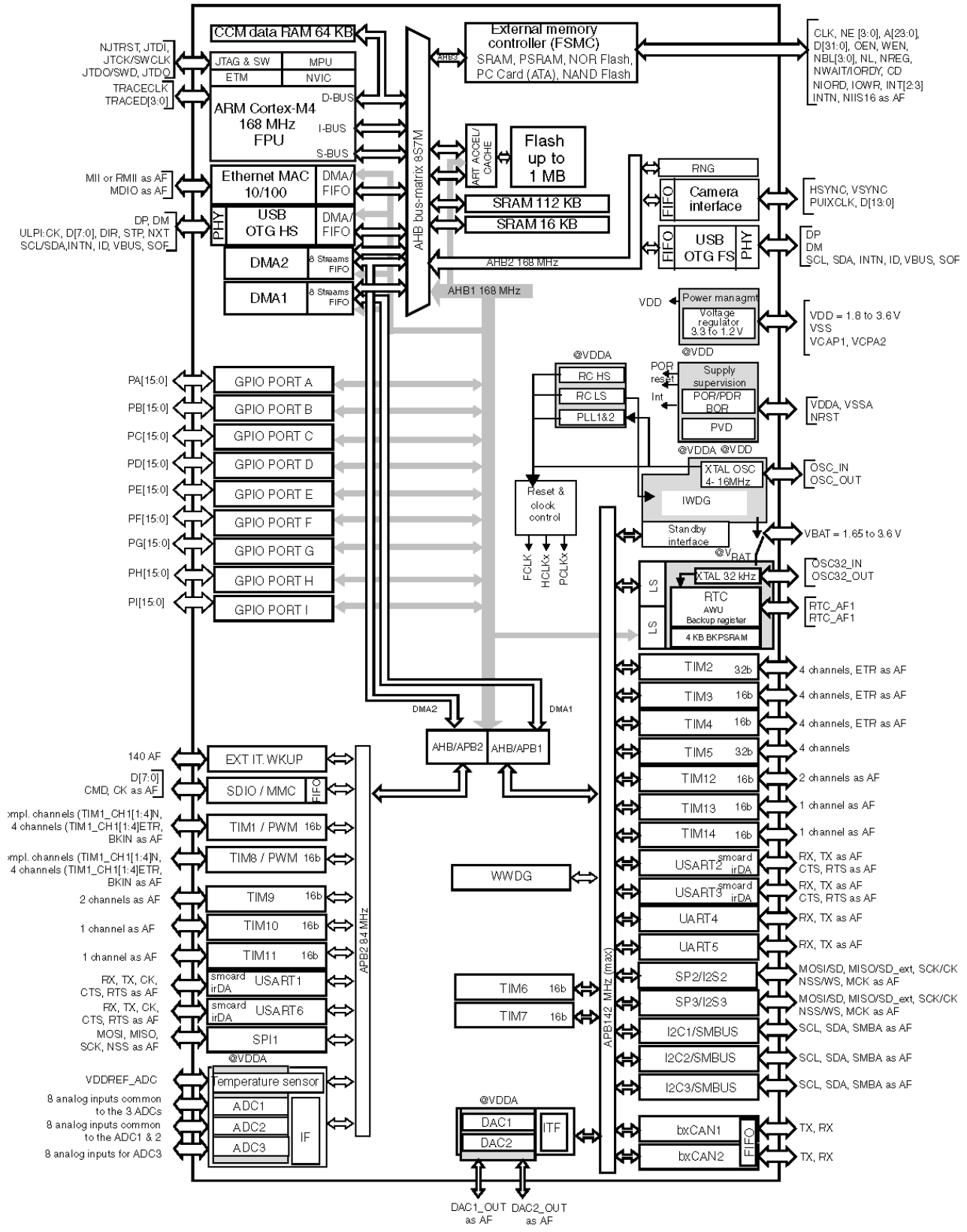

Figura 3 – Diagrama de blocos da estrutura interna do STM32F407.

Memory Protection Unit

A unidade de proteção de memória (*MPU*) é usada para gerenciar os acessos da *CPU* à memória para impedir que uma tarefa corrompa acidentalmente a memória ou os recursos usados por qualquer outra tarefa ativa. A *MPU,* geralmente gerenciada por um *RTOS,* é útil para aplicações em que algum código crítico deve ser protegido contra o mau comportamento de outras tarefas. Se um programa acessar um local de memória proibido pela *MPU*, o *RTOS* poderá detectá-lo e executar uma ação de correção.

Controlador de interrupção vetorada e aninhada (NVIC)

O STM32F407 incorpora um controlador de interrupção vetorada e aninhada (*Nested Vectored Interupt Controller - NVIC*) capaz de gerenciar 16 níveis de prioridade e manipular até 82 canais de interrupção mascaráveis, além das 16 linhas de interrupção do Cortex-M4. Ele fornece o processamento de interrupções com baixa latência. O estado do processador é salvo automaticamente.

Clock e startup

O STM32F407 possui várias fontes de clock. Na inicialização, um oscilador RC interno de 16 MHz é selecionado como a fonte de clock padrão da *CPU*. O oscilador RC interno de 16 MHz é ajustado de fábrica para oferecer 1% de precisão em toda a faixa de temperatura de operação. A aplicação do usuário pode então selecionar como fonte de clock do sistema o oscilador RC ou uma fonte externa de 4-26 MHz. Este clock pode ser monitorado quanto a falhas. Se uma falha for detectada, o sistema retornará automaticamente ao oscilador RC interno e uma interrupção será gerada (se tiver sido ativada). O sinal dessa fonte de clock pode passar por um circuito multiplicador de clock (PLL), permitindo aumentar a frequência até 168 MHz. Vários divisores (*prescalers*) permitem a escolha da velocidade do clock dos barramentos do sistema. A frequência máxima dos três barramentos AHB (*Advanced High speed Bus*) é de 168 MHz, enquanto a frequência máxima do barramento APB (*Advanced Peripheral Bus*) de alta velocidade é de 84 MHz e a frequência máxima do barramento APB de baixa velocidade é 42 MHz.

Real Time Clock

O relógio em tempo real (RTC) é um contador independente. Registradores dedicados armazenam a contagem de segundos, minutos, horas (nos formatos 12/24 horas), dia da semana, data, mês, ano, no formato BCD (decimal codificado em binário). A correção para 28, 29 dias (ano bissexto), 30 e 31 dias no mês é realizada automaticamente. O RTC fornece um alarme programável e interrupções periódicas programáveis. O valor de sub-segundos também está disponível em formato binário.

Ele tem clock selecionável proveniente de um oscilador RC interno de baixa velocidade (32 kHz), de um cristal/ressonador externo de 32,768 kHz, ou o clock externo de alta velocidade dividido por 128.

Entradas e saídas de uso geral (General Purpose Input/Output - GPIO)

Cada um dos pinos de *GPIO* pode ser configurado pelo software como saída digital (*push-pull* ou *open-drain*, com ou sem resistores de *pull-up* ou *pull-down*), como entrada digital (flutuante, com/sem resistor de *pull-up/down*), como entrada analógica ou, alternativamente, como entrada/saída (digital ou analógica) de um determinado periférico. Todos os *GPIOs* possuem seleção de velocidade para melhor gerenciamento do ruído interno, consumo de energia e emissões eletromagnéticas. ■