

Unidade 2

Capítulo 2 - Arquitetura de
Sistemas Embarcados

 Capacitação Inicial em Sistemas embarcados

Prof. Pedro Klecius

Sumário

1. Introdução	2
2. Arquitetura do SE	2
3. Arquitetura do hardware	3
4. Arquitetura de camadas	5
5. Componentes de um SE	8
• Microcontrolador	9
• Field-Programable Gate Array–FPGA	10
• Memória Volátil	10
• Memória Não-Volátil	12
6. Periféricos/Interfaces	13
• Timer	13
• Contadores	14
• Teclado	15
• Interfaces Seriais	16
7. Aplicação dos Sistemas Embarcados	25
• Indústria	25
• Segurança	25
• Agriculturas	25
• Medicina	26
• Logística	26
8. Conclusão	27
9. Referências	28

1. Introdução

Olá, aluno, do EmbarcaTech. Bem-vindo à nossa aula sobre sistemas embarcados. Vimos na aula anterior que podemos diferenciar os embarcados com relação ao modo de funcionamento, definir e classificar os vários tipos de aplicação e a reconhecer as características de cada um. Por fim, compreendemos que existe uma série de restrições que devem ser levadas em consideração durante o desenvolvimento de um sistema.

Nesta aula você aprenderá sobre a arquitetura e os componentes de cada uma de suas camadas. Conhecerá, através de exemplos, as funcionalidades nas diversas áreas de aplicação.

Espero que você goste da aula. Segundo Einstein "As pessoas podem tirar tudo de você, menos o seu conhecimento". Então vá em frente, aprenda, conheça cada vez mais. Estamos aqui para lhe auxiliar em tudo que pudermos.

2. Arquitetura do SE

São duas as principais partes funcionais de um sistema embarcado. A primeira consiste no **hardware embarcado**: Componentes físicos que formam o hardware interligados dentro de um mesmo chip e de interfaces montados e interligado por uma PCI (placa de circuito impresso) [1]. A segunda parte são os **programas de software** [1], que farão funcionar o hardware, gravados nas memórias do sistema e que consistem em sistema operacional, firmware, rotinas de debug e drivers.

A Figura 1 - Arquitetura do Sistema Embarcado, ilustra de forma visual a estrutura dessa arquitetura.

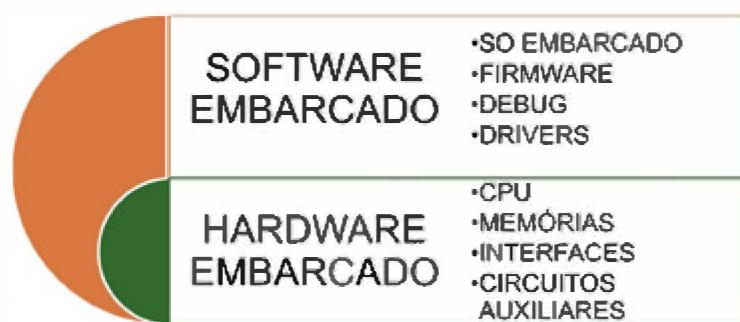


Figura 1 - Arquitetura do Sistema Embarcado (Fonte: Autor)

A imagem apresenta uma divisão entre "Software Embarcado" e "Hardware Embarcado", ilustrada em um formato visual com duas formas semicirculares sobrepostas nas cores laranja e verde à esquerda. À direita, há uma caixa retangular subdividida em duas partes: a parte superior, rotulada como "Software Embarcado", lista os componentes "SO Embarcado", "Firmware", "Debug" e "Drivers"; já a parte inferior, rotulada como "Hardware Embarcado", lista "CPU", "Memórias", "Interfaces" e "Circuitos Auxiliares". A imagem está cercada por uma linha pontilhada azul que delimita o contorno da figura.

3. Arquitetura do Hardware

Os componentes do hardware de um sistema embarcado são o **microcontrolador (MCU)**, as **memórias externas** e, caso seja necessário, as **interfaces, sensores e atuadores** [2].

Estes componentes de hardware são interligados por **ligações físicas** que são agrupados pelo tipo de informação a serem transportados e são denominados **barramento**.

São três tipos de barramento, mostrados na Figura 2 - Arquitetura do Hardware, a saber:

- **Dados**: usado pela MPU para leitura/escrita de dados com a memória/interfaces;
- **Endereços**: usado pela MPU para endereçar memória e as interfaces;
- **CTRL**: barramento usado pela MPU para controlar memória e interfaces.



Figura 2 - Arquitetura do Hardware (Fonte: Autor)

A imagem apresenta a "Arquitetura do Hardware", ilustrada com blocos interconectados. À esquerda, há um bloco identificado como "MPU", que se conecta através de barramentos com os rótulos "Dados", "Endereços" e "CTRL". Esses barramentos conectam o MPU a outros dois blocos: um identificado como "Memórias" e outro como "Interfaces". À direita, há uma seção separada, intitulada "Periféricos", que contém dois grupos: "Sensores", representados por ícones de componentes de hardware, e "Atuadores", também representados por ícones de dispositivos. A imagem está circundada por linhas pontilhadas vermelhas, e acima do bloco principal está o título "Microcontrolador". Ao lado direito, o título "Periféricos" aparece próximo aos ícones.

Essa arquitetura pode funcionar de duas maneiras. A primeira é chamada de **Harvard**. Cada memória possui seu próprio barramento de endereços e dados, significando que a MPU acessa as memórias simultaneamente. A vantagem dessa forma é sua grande rapidez pois a MPU lê uma instrução ao mesmo tempo que acessa os dados, permitindo que as instruções sejam executadas em 1(um) ciclo de Clock.

Os microcontroladores que funcionam desta forma são chamados de "Microcontroladores RISC" (Reduced Instruction Set Computer). A Figura 3 – Arquitetura Harvard mostra a interligação dos componentes pelos seus respectivos barramentos.

MICROCONTROLADOR

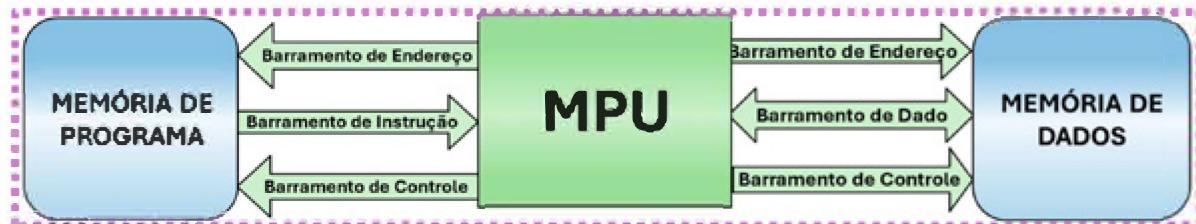


Figura 3 - Arquitetura Harvard (Fonte: Autor)

A imagem ilustra a arquitetura de um microcontrolador, com três blocos principais. No centro, encontra-se o bloco "MPU", representado em verde. À esquerda, está o bloco "Memória de Programa", e à direita, o bloco "Memória de Dados", ambos em azul. Esses blocos estão conectados ao MPU por diferentes barramentos: "Barramento de Endereço", "Barramento de Instrução" e "Barramento de Controle" conectam a Memória de Programa ao MPU; enquanto "Barramento de Endereço", "Barramento de Dado" e "Barramento de Controle" conectam a Memória de Dados ao MPU. Toda a estrutura está contornada por uma linha pontilhada roxa, e a parte superior da imagem traz o título "Microcontrolador".

A segunda forma é a Von NEUMANN, onde **as memória e interfaces possuem o barramento de endereços e dados em comum**. Isto significa que **as instruções e dados estão armazenados no mesmo espaço de memória**, ou seja, que **a MPU acessa uma informação, instrução ou dados, de cada vez**. A desvantagem dessa forma é sua **lentidão**, dado que, para executar uma instrução, a MPU lê a instrução, depois lê o dado, somente então a instrução estará finalizada.

Os microcontroladores que funcionam desta maneira são chamados "Microcontrolador CISC" (Complex Instruction Set Computer), ou microcontrolador com um Conjunto Complexo de Instruções.

MICROCONTROLADOR

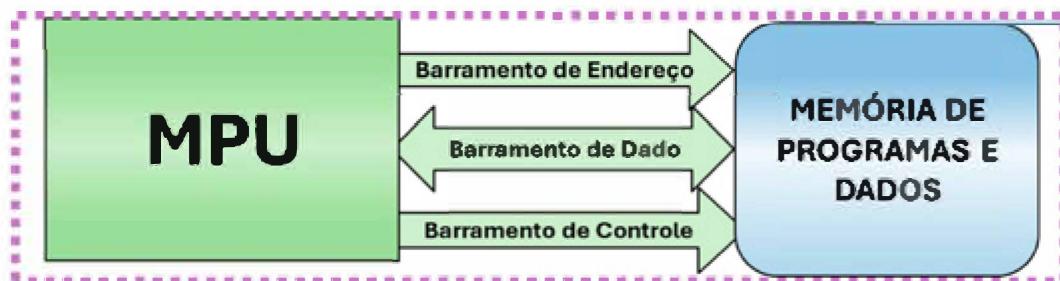


Figura 4 - Arquitetura Von NEUMANN (Fonte: Autor)

A imagem apresenta uma arquitetura de microcontrolador com dois blocos principais. À esquerda, está o bloco "MPU", representado em verde, e à direita, o bloco "Memória de Programas e Dados", representado em azul. Esses dois blocos estão conectados por três barramentos: "Barramento de Endereço", "Barramento de Dado" e "Barramento de Controle". A imagem é circundada por uma linha pontilhada roxa, e o título "Microcontrolador" está posicionado acima dos blocos principais.

4. Arquitetura de Camadas

Os componentes de hardware e software estão agrupados, conforme mostra a Figura 5 – Arquitetura de camadas, em um modelo chamada de “três camadas”. As camadas são a camada Hardware, a camada de software básico e a camada de software de aplicação.

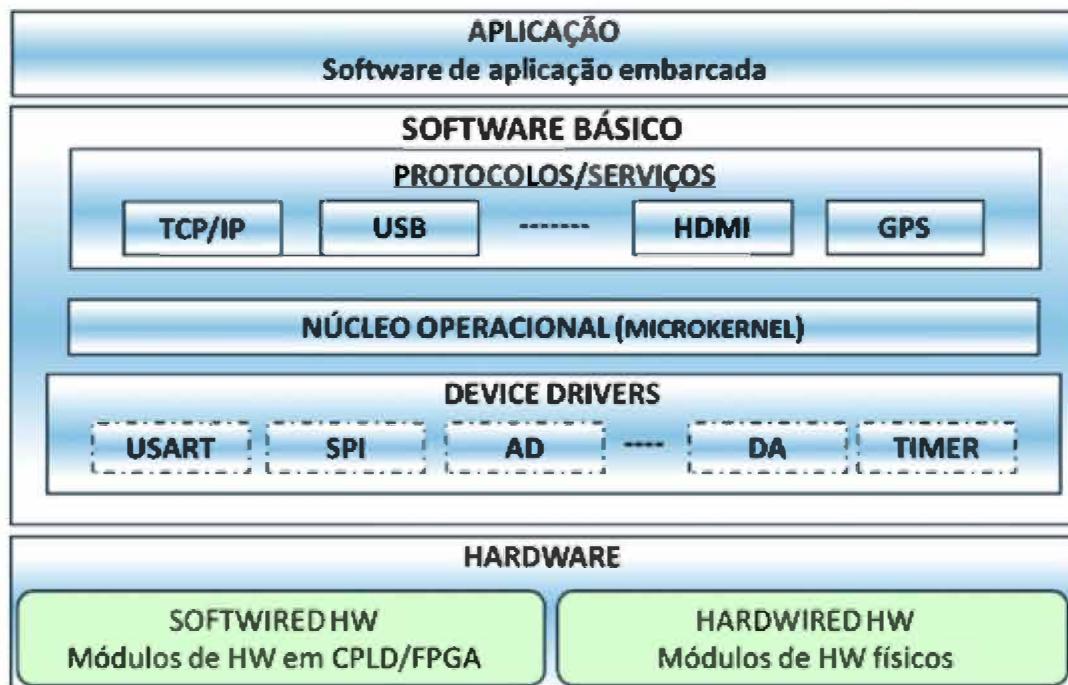


Figura 5 – Arquitetura de Camadas (Fonte: Autor)

A imagem ilustra a estrutura de um sistema embarcado, dividida em várias camadas. No topo, está a camada de “Aplicação”, representando o “Software de aplicação embarcada”. Abaixo dela, há a camada “Software Básico”, que contém duas seções: a primeira, chamada “Protocolos/Serviços”, inclui protocolos como “TCP/IP”, “USB”, “HDMI” e “GPS”. A segunda seção, “Núcleo Operacional (Microkernel)”, é seguida pela seção “Device Drivers”, que lista drivers como “USART”, “SPI”, “AD”, “DA” e “TIMER”. Na base, está a camada de “Hardware”, dividida em “Softwired HW” (Módulos de HW em CPLD/FPGA) e “Hardwired HW” (Módulos de HW físicos). Toda a imagem é organizada em blocos horizontais com tons de azul e verde.

A camada hardware abriga os componentes de hardware do sistema podendo esse hardware ser do tipo **hardwired HW** e/ou **softwired HW**.



Figura 6 – Camada de Hardware (Fonte: Autor)

A imagem apresenta a divisão da camada “Hardware” em dois blocos principais. O bloco à esquerda é rotulado como “Softwired HW” e descreve “Módulos de HW em CPLD/FPGA”, enquanto o bloco à direita é rotulado como “Hardwired HW” e descreve “Módulos de HW físicos”. Ambos os blocos estão dispostos lado a lado em verde claro, com o título “Hardware” no topo da imagem em um fundo azul claro.

Quando o hardware é composto de **componentes físicos de hardware** dizemos que é do tipo **Hardwired-HW** e os componentes (MPU, timers, Uart,etc.) interligados fixamente dentro de um mesmo chips e/ou realizadas através de trilhas em placas de circuito impresso.

Os sinais elétricos comandados por dispositivos semicondutores são usados para a realização das atividades e os transistores são interligados formando portas lógicas, blocos funcionais, processadores, memórias, e dispositivos de entrada e saída.

Chamamos de Softwired Hw quando o hardware é programado em software armazenado em dispositivos lógicos programáveis ou matriz de portas lógicas cujos pontos de interligação podem ser ativados ou não por software. A programação destes dispositivos é tipicamente realizada em linguagem VHDL, SystemC, VERILOG, Handel-C, SDL entre outras.

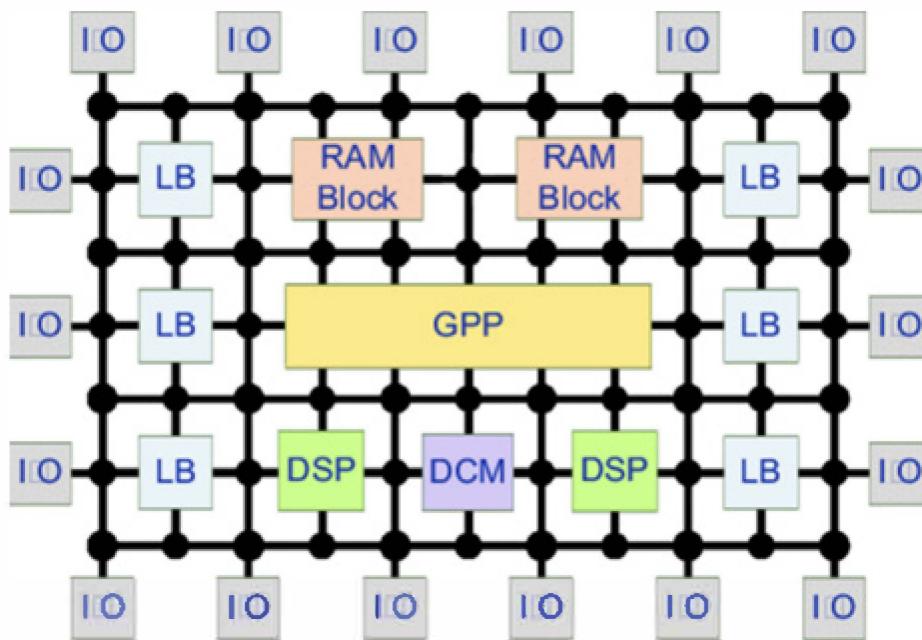


Figura 7 - Representação arquitetura de uma FPGA

(Fonte: https://www.researchgate.net/figure/FPGA-internal-structure-example_fig6_273055941)

A imagem apresenta uma representação gráfica de uma arquitetura de hardware em formato de grade, com vários blocos conectados por linhas pretas que formam uma malha interligada. Nos cantos e nas bordas da grade, há blocos rotulados como "I/O" (Entrada/Saída). Dentro da grade, há diversos blocos rotulados: "LB" (em azul claro), "RAM Block" (em laranja), "GPP" (em amarelo), "DSP" (em verde) e "DCM" (em roxo). Os blocos "LB" estão dispostos em várias partes da grade, enquanto os blocos "RAM Block", "GPP", "DSP" e "DCM" ocupam posições centrais. A imagem parece representar uma arquitetura modular, conectando componentes de processamento e memória por meio de uma malha de interconexão.

A camada de software básico está composta da subcamada device drivers, núcleo operacional e da subcamada de protocolos e serviços.

Os devices drivers interagem diretamente com os dispositivos de hardware da camada inferior. São módulos de software que encapsulam cada um dos dispositivos de hardware (temporizadores, portas seriais, controladores de interrupção, cartões de memória etc.).

As camadas superiores acessam os dispositivos de hardware através dos device drivers.



Figura 8 - Camada de Device Drivers (Fonte: Autor)

A imagem apresenta uma seção intitulada "Device Drivers", com uma lista de componentes organizados em uma linha. Esses componentes estão representados em retângulos pontilhados. A lista inclui, da esquerda para a direita, os seguintes drivers: "USART", "SPI", "AD", um espaço com uma linha pontilhada, "DA" e "TIMER". O fundo dos retângulos tem um efeito gradiente em azul claro, e os componentes estão dispostos de maneira uniforme dentro da caixa.

O núcleo operacional, o kernel ou RTOS (Real-Time Operating System), quando implementado, tem como principal objetivo permitir a execução concorrente de várias tarefas. O **Microkernel**, módulo central do núcleo operacional, faz o gerenciamento e escalonamento de tarefas, de memória, de comunicação entre tarefas e de serviços de temporização.

Sobre os canais de comunicação trafegam informações (Ethernet, USB, RS-232, RS-485, GPRS) que são estruturadas e definidas por um protocolo padronizado de comunicação, formando a subcamada de protocolo e serviços.

O protocolo define o formato dos pacotes de dados a serem transmitidos e a forma de interação entre os equipamentos em comunicação. Os protocolos são implementados em módulos de software que oferecem serviços de comunicação às camadas superiores. A finalidade do protocolo é permitir a interoperabilidade entre os sistemas de comunicação.

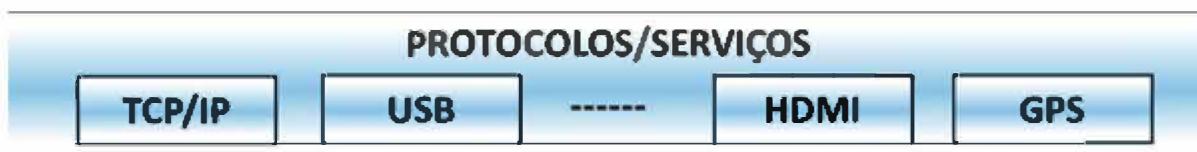


Figura 9 - Camada de Protocolos/Serviços (Fonte: Autor)

A imagem exibe uma seção intitulada "Protocolos/Serviços", com uma série de protocolos organizados em uma linha horizontal. Os protocolos estão representados em retângulos com bordas e fundo gradiente em azul claro. Da esquerda para a direita, são exibidos: "TCP/IP", "USB", um espaço com uma linha pontilhada, "HDMI" e "GPS". Cada protocolo está destacado em seu próprio retângulo, dispostos de maneira uniforme.

Os módulos de serviços implementam módulos de software que são reutilizados em diversos sistemas embarcados. Ex. Módulos de georreferenciamento e de criptografia. A camada de software de aplicação, chamada firmware, implementa a funcionalidade específica de um determinado sistema embarcado.

5. Componentes de um SE

Como vimos, um sistema embarcado está composto de inúmeros componentes necessários a realização de sua função específica. Vela a relação abaixo exemplos de diversos desses componentes:

Sensores: Fotoelétrico, Detectores de sons, Velocidade, Temperatura, Magnéticos, Pressão, Acelerômetros, Giroscópios, Gás e fumaça.



Figura 10 - Sensores

(Fonte: www.eletropecas.com/produtos/sensores)

A imagem apresenta quatro tipos de sensores dispostos em linha. Da esquerda para a direita, estão: um sensor Capacitivo, representado por um bloco preto retangular; um sensor de Efeito Hall, com formato cilíndrico metálico e um cabo conectado; um sensor Magnético, cilíndrico e fino, com conectores coloridos; e um sensor de Temperatura, também cilíndrico, metálico com uma ponta vermelha. Todos os sensores estão identificados com suas respectivas legendas abaixo das imagens.

Atuadores: Relés, Drivers de Motores, Drivers de LEDs, Emissores de IR, SCR, TRIAC.

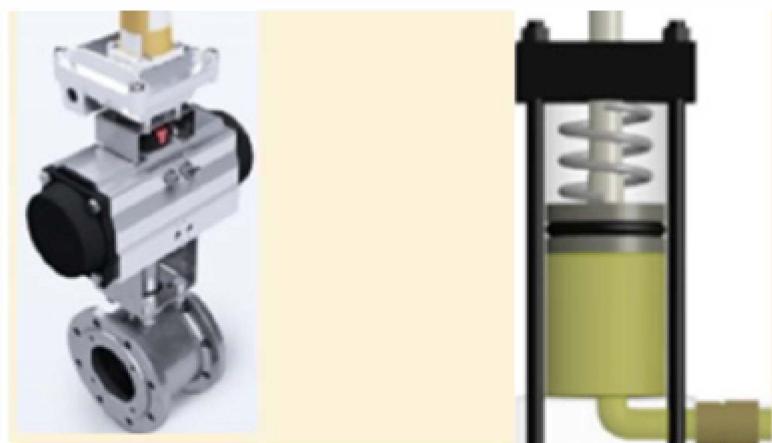


Figura 11 - Atuadores

(Fonte: [https://www.youtube.com/watch?v=svKi0LRfDGA](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSyiT00Uzp8SFKU0ppn2toiH0bNohIm68-myd-MA54pwB1lo4Bdghttps://www.youtube.com/watch?v=svKi0LRfDGA))

A imagem apresenta dois tipos de atuadores industriais. À esquerda, há um atuador rotativo metálico com uma base de flange e um corpo prateado, utilizado em válvulas de controle de fluxo. À direita, está um atuador linear, representado por um cilindro de cor amarela dentro de uma estrutura vertical preta, com uma mola visível na parte superior, usado para aplicações de movimento linear. Ambos os atuadores têm estrutura robusta e são comumente utilizados em processos industriais.

Componentes Digitais: Microprocessadores, coprocessadores, memórias, barramentos, Microcontroladores, Application Specific Hardwares.

Conversores: Conversor Analógico-Digital (ADC), Digital-Analógico.

Software: Sistema Operacional, Device Driver, Aplicações (MPEG-x, GSM-kernel).

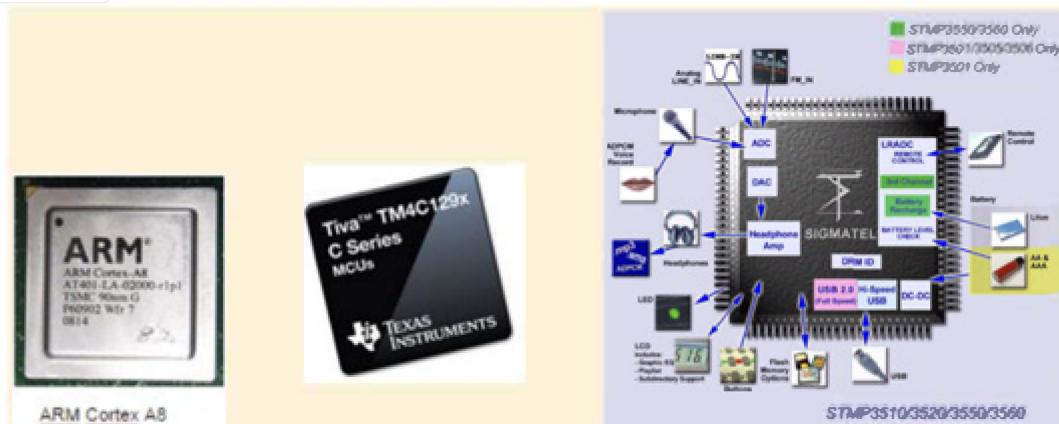


Figura 12 - Exemplos de microcontroladores e processadores embarcados: à esquerda, o chip ARM Cortex A8; no centro, o microcontrolador Tiva™ TM4C129x da Texas Instruments; e à direita, o diagrama do processador SIGMATEL STMP3510/3520/3550/3560, com suas principais conexões e funcionalidades.

(Fonte: <https://www.hardware.com.br/static/00000000/img-bec95177.png>)

A imagem mostra três diferentes componentes eletrônicos. À esquerda, há uma foto de um chip ARM Cortex A8, com o logotipo "ARM" visível na superfície do chip. No centro, há uma imagem de outro chip identificado como Tiva™ TM4C129x C Series MCUs, fabricado pela Texas Instruments, com o logotipo da empresa. À direita, há um diagrama de um chip SIGMATEL, onde são ilustrados vários componentes e conexões ao redor do chip, incluindo entradas e saídas para microfone, alto-falante, USB, bateria, entre outros. O chip é identificado pelos números de modelo "STMP3510/3520/3550/3560". O diagrama contém caixas coloridas em verde, azul e roxo, representando a funcionalidade dos diferentes modelos da série STMP.

• Microcontrolador

Microcontroladores são processadores que **incorporam muitas funções em um único chip**. Um microcontrolador pode incorporar conversores analógico-digitais (ADC) e digital-analógicos (DAC), temporizadores, contadores, interfaces seriais, memória de instruções/dados, controladores de interrupção, geradores de clock, controladores de DMA, etc.

É comum vários **microcontroladores pertencentes a mesma família com diferentes dispositivos, frequência de clock, potências consumidas, faixa de temperatura suportada, encapsulamento e preços**.

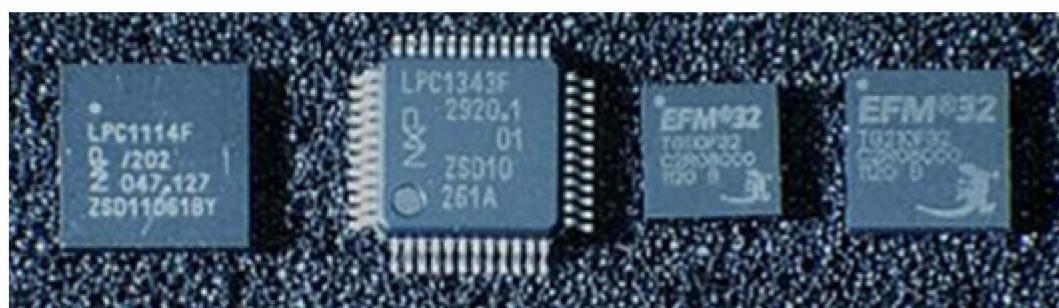


Figura 13 - Microcontroladores de diferentes modelos: LPC1114F e LPC1343F da família LPC, e dois microcontroladores EFM32, com diferentes variações, utilizados em sistemas embarcados.

(Fonte: https://upload.wikimedia.org/wikipedia/commons/thumb/3/3d/ARM_Cortex-M0_and_M3_ICs_in_SMD_Packages.jpg/540px-ARM_Cortex-M0_and_M3_ICs_in_SMD_Packages.jpg)

A imagem mostra quatro microcontroladores dispostos em uma linha, todos em um fundo texturizado escuro. Da esquerda para a direita, o primeiro chip é identificado como LPC1114F com inscrições menores detalhando seu número de série. O segundo é um chip maior com pinos ao redor, identificado como LPC1343F, com mais inscrições na superfície. O terceiro e o quarto chips são identificados como EFM32, ambos com tamanho similar e com diferentes números de série.

São famílias de microcontroladores: ARM Cortex-M core (aplicações de MCU), Microchip Technology Atmel AVR, Cypress Semiconductor M8C core PSoC, Freescale Cold Fire(32-bit), Intel 8051, Infineon 8-bit XC800, 16-bit XE166, 32-bit XMC4000, Microchip Technology PIC, NXP Semiconductors LPC1000, LPC2000, Renesas Electronics, RL78, RX 32-bit MCU; SuperH, Silicon Laboratories Pipelined8-bit 8051, STMicroelectronics STM8 (8-bit), ST10 (16-bit), STM32 (32-bit), Texas Instruments TI MSP430, MSP432, Toshiba TLCS-870.

• Field-Progamable Gate Array – FPGA

São dispositivos de hardware **programáveis e reconfiguráveis** on-line (tipo RAM). Possuem tipos diferentes de programação: **tipo PROM** (programáveis uma única vez), **tipo EPROM** (reprogramáveis em laboratório), tipo **RAM** (reprogramáveis durante a execução).

SOC

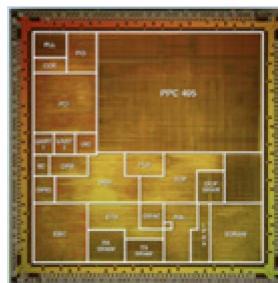


Figura 14 - Visão detalhada do layout interno de um chip com o núcleo PPC 405 e diversos módulos periféricos, como PCI, UART, GPIO, e controladores de memória SRAM e Ethernet.

(Fonte: www.inf.ufrgs.br/sim-emicro/papers/sim_palestras_moraes.pdf)

A imagem mostra o layout interno de um chip, com diversos blocos rotulados, cada um representando diferentes módulos e funcionalidades dentro do circuito. No centro, há um grande bloco identificado como PPC 405. Ao redor, há vários outros blocos menores rotulados como PLL, PCI, UART 0, UART 1, GPIO, ETH, SRAM, entre outros. O layout é colorido em tons de dourado e marrom, representando a estrutura interna do chip.

• Memória Volátil

As memórias chamadas voláteis são aquelas nas quais o microcontrolador pode ler, armazenar, ler-modificar-reescrever os dados no momento que ele der um comando para isso.

Entretanto, o dado só permanece gravado na memória enquanto ela estiver ligada a uma fonte de energia (por exemplo alimentada com 3,3V ou 5V).

Quando o sistema é desligado da energia, os dados são apagados da memória.

São chamadas de **memória de acesso randômico** (RAM- Random access memory) porque a qualquer instante um dado pode ser lido ou gravado na memória em um endereço qualquer dela. Isso significa dizer que os endereços que estão sendo acessados pelo microcontrolador **não precisam estar em ordem sequencial**, mas em qualquer ordem dependendo de sua localização na memória.

Memórias de acesso aleatório podem ser implementadas usando diversas tecnologias. As **memórias SRAM**, por exemplo, são memórias estáticas de acesso aleatório, permitindo acesso aos endereços em qualquer ordem com capacidades de leitura e escrita.

Uma vez armazenado um dado em um determinado endereço, ele permanece até que seja sobrescrito ou até que a alimentação de energia da memória seja interrompida; por isso, são chamadas de **RAM estáticas**. Essas memórias são usadas comumente para **armazenar dados quase permanentes do sistema** e são construídas por células de **armazenamento do tipo latches estáticos**. O tempo de acesso dos dados é pequeno, mas o custo por bit da memória é elevado dada a quantidade de componentes lógicos de cada latche.

Outro tipo de memória volátil é a **DRAM** (Dynamic Random Access Memory). Elas são construídas usando uma tecnologia na qual **as células de armazenamento são constituídas de capacitores** onde os dados são armazenados. Por esse motivo a memória possui **grande capacidade de armazenamento** em um pequeno chip, o que faz baixar o custo por bit da memória.

Entretanto, como o armazenamento dos dados são realizados por capacitores, **esses perdem seu conteúdo em breve período de tempo**, fazendo-se necessário um sistema de rearmazenamento periódico dos dados já gravados. Esse sistema é chamado de **refreshing**.



Figura 15 - Módulos de memória RAM com diferentes especificações, usados em sistemas computacionais para armazenamento temporário de dados.

(Fonte: www.inf.ufrgs.br/sim-emicro/papers/sim_palestras_moraes.pdf)

A imagem mostra três módulos de memória RAM dispostos em um formato de leque, sobrepostos parcialmente. Os módulos têm placas de circuito verde com chips pretos soldados e adesivos de identificação. O módulo no topo possui uma cobertura metálica com etiquetas e informações técnicas visíveis.

• Memória Não-Volátil

A **ROM** (read-only memory) é um tipo de memória na qual os dados gravados permanecem, mesmo que a alimentação da memória seja desligada, ou seja, os dados são gravados de forma permanente ou semipermanente.

Isso significa dizer que uma vez gravado um dado na memória ele não pode ser apagado, exceto se a memória for submetida a uma operação de apagamento total dos dados, para então serem gravados novos dados.

Dados podem ser lidos da ROM, porém não existe operação de escrita. Essa memória é normalmente usada para armazenar programas e dados permanentes (por exemplo, configuração).

Dentre as tecnologias de fabricação desta memória destacamos a PROM (Programmable ROM), a EPROM (Erasable PROM), EEPROM (Electrically Erasable PROM) e a Flash EPROM (Fast erasable EPROM).

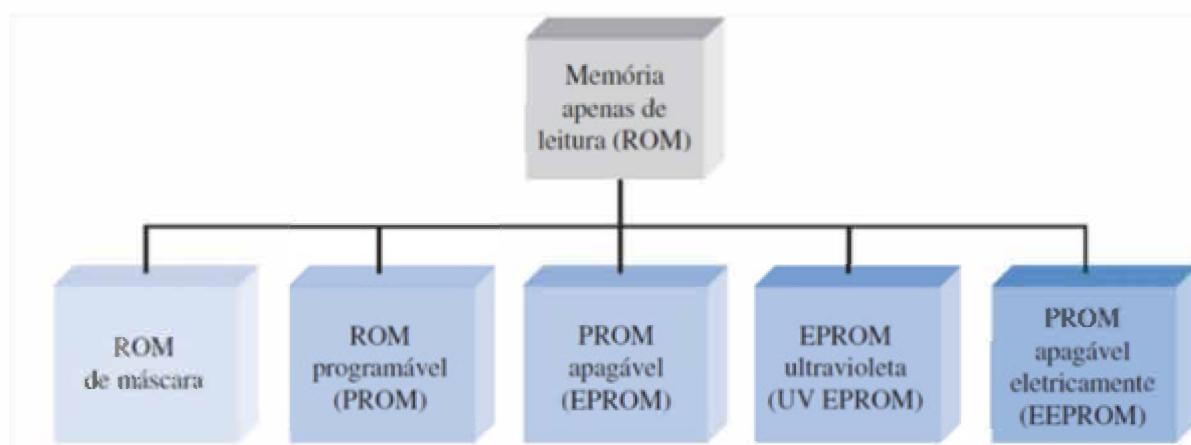


Figura 16 - Diagrama mostrando os tipos de memórias ROM: de máscara, programável (PROM), apagável (EPROM), EPROM ultravioleta (UV EPROM) e EEPROM, ilustrando as diferentes categorias de memórias não voláteis.

(Fonte: Sistemas Digitais - Fundamentos e aplicações - Floyd, Thomas)

A imagem exibe um diagrama hierárquico que detalha os diferentes tipos de Memória Apenas de Leitura (ROM). No topo, há um bloco cinza rotulado como "Memória apenas de leitura (ROM)". Abaixo, cinco blocos azuis estão conectados a esse bloco principal, representando os diferentes tipos de ROM:ROM de máscara,ROM programável (PROM),ROM apagável (EPROM),EPROM ultravioleta (UV EPROM),ROM apagável eletricamente (EEPROM).

6. Periféricos/Interfaces

Como você já sabe, **os periféricos são interligados a MCU através das interfaces**. São muitas as interfaces disponíveis. A lista abaixo contém alguns exemplos:

- Interface para teclado, interface para LCD, Temporizadores, Contadores
- Interfaces Seriais, Transdutores, Conversores Digital-Analógicos
- Analógico-Digitais, Modulação de largura de pulso -PWM
- Interface para Motores de passo, Interface para Sensores e Atuadores

• Timer

Os Timers são circuitos internos do microcontrolador cuja função é contar os pulsos de Clock gerados pelo próprio microcontrolador.

Por essa razão, esses são circuitos temporizadores ou contadores de tempo, onde o tempo contado é determinado pelo valor da contagem multiplicado pelo período do Clock fornecido pelo microcontrolador. Os Timers são construídos com **flip/flops** e funcionam de forma independente do microcontrolador, ou seja, se ele está programado e recebe os pulsos de Clock ele executa a contagem, independente do que o microcontrolador esteja fazendo. Logo, o Timer pode ser usado para medir o tempo de execução de um evento, o tempo transcorrido entre um evento e outro, para temporizações em geral, PWM etc.

O Timer possui registradores internos que são incrementados juntamente com o Clock até que alcancem o valor de tempo pré-configurado, ou até que o evento acabe, gerando então um alarme para o microcontrolador, permitindo que este meça o tempo dado pelos registros internos.

Como o Clock é geralmente alto, um circuito “prescaler” divide a frequência do Clock por um valor que pode ser programado por software, conforme a figura abaixo.

Como o Clock é geralmente alto, um circuito “prescaler” divide a frequência do Clock por um valor que pode ser programado por software, conforme a figura abaixo.

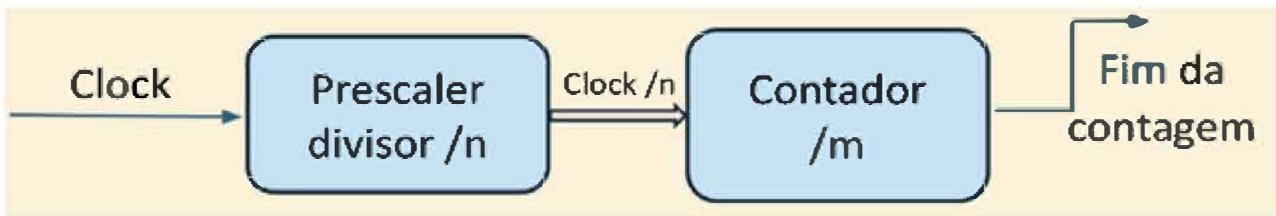


Figura 17 - Diagrama de blocos representando o funcionamento de um prescaler e contador, utilizados para dividir a frequência de um sinal de clock e contar pulsos até o fim da contagem.

(Fonte: Sistemas Digitais - Fundamentos e aplicações - Floyd, Thomas)

A imagem mostra um diagrama de blocos representando um processo de contagem utilizando um Prescaler e um Contador. À esquerda, há uma entrada de Clock, que passa primeiro pelo bloco do Prescaler, identificado como “divisor /n”, que divide a frequência do clock original. O sinal resultante, chamado Clock /n, é então enviado ao próximo bloco, o Contador, identificado como “Contador /m”. Após o contador atingir o valor desejado, o diagrama exibe uma seta indicando o Fim da contagem à direita.

• Contadores

Os contadores são circuitos Timers na função de contar eventos originados externamente ao microcontrolador através dos pinos dele (Tn). A figura 18 ilustra o circuito do Timer/Counter de 16 bits ($n=1, 3, 4$ ou 5) do microcontrolador Atmel. O contador TCNTn, composto por dois contadores de 8 bits, conta os sinais recebidos do bloco Control Logic, que recebe um pulso do bloco Clock Select. Este bloco escolhe, conforme programado, entre sinal de "eventos externos" (contador) ou "eventos internos" (temporizador).

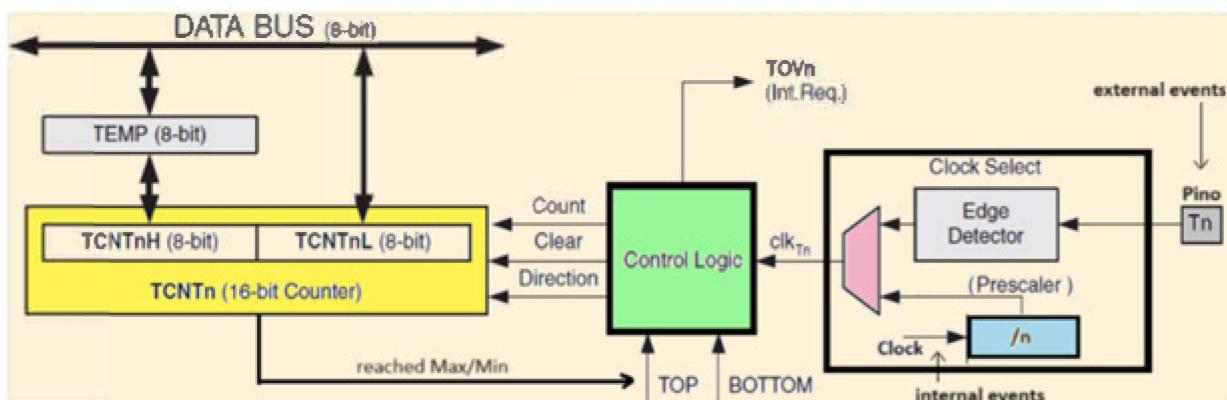


Figura 18 - Diagrama de um sistema de contagem baseado em um contador de 16 bits, com barramento de dados de 8 bits, lógica de controle, prescaler e detecção de borda para processamento de sinais de clock.

(Fonte: Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V)

A imagem apresenta um diagrama de blocos de um sistema de contagem baseado em um contador de 16 bits(TCNTn). À esquerda, o diagrama mostra a interface com o DATA BUS (barramento de dados de 8 bits), que se conecta a um bloco identificado como TEMP (8-bit). O contador principal (TCNTn) é destacado em amarelo, com entradas e saídas para TGNnH (8-bit) e TGNnL (8-bit). O diagrama segue para o bloco de Control Logic (em verde), que recebe sinais de Clear, Count, e Direction, e fornece uma saída para o contador. À direita do bloco de controle, há um bloco de Edge Detector(detector de borda) e um Prescaler, que ajusta a frequência de clock. O diagrama finaliza com uma saída chamada Tn. O sistema inclui também uma entrada para IOV in (MAtched) e outras funções adicionais.

• Teclado

Um teclado é uma matriz de linhas e colunas, ligadas aos pinos do microcontrolador, que se cruzam e em cada cruzamento possui uma chave que liga uma coluna com uma linha quando pressionada.

Para detectar uma tecla, o microcontrolador coloca nível 0 na H0 e lê as verticais V0 a V3. Na vertical na qual o microcontrolador lê um nível 0, a tecla foi pressionada e ele calcula do código da tecla, usando uma tabela por exemplo.

Depois repete o processo colocando um nível 0 em para H1, depois H2 e depois H3, lendo as verticais para cada detectar uma chave pressionada. Após repete o procedimento.

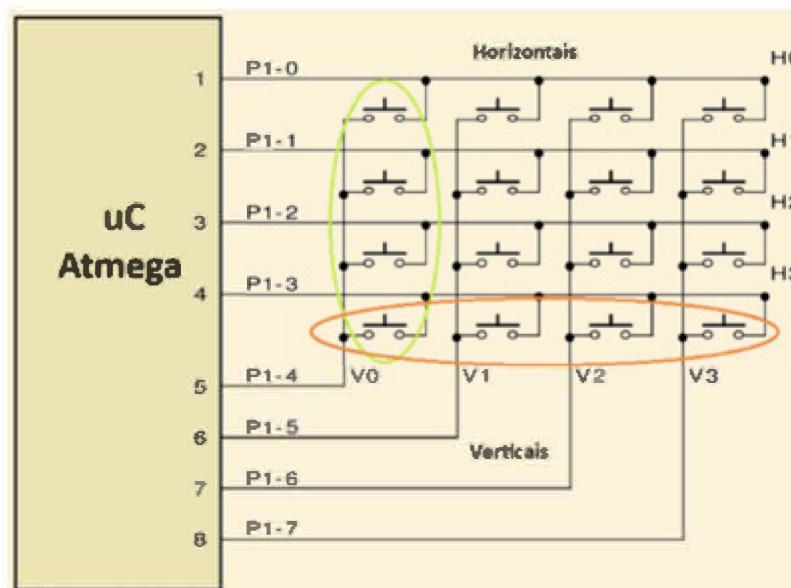


Figura 19 - Matriz de teclado (Fonte: Autor)

A imagem mostra um diagrama esquemático de conexões envolvendo um microcontrolador Atmega (à esquerda) e um conjunto de interruptores (à direita), organizados em uma matriz. O microcontrolador possui várias portas, rotuladas de P1 a P7, conectadas a diferentes filas de interruptores na matriz. A matriz é dividida em dois grupos: na parte superior, há interruptores conectados a linhas rotuladas como HG, e na parte inferior, conectados a linhas rotuladas como HB. As colunas intermediárias são rotuladas como W0, W1, W2, e W3. Duas elipses, uma verde e outra laranja, destacam grupos específicos de interruptores dentro do diagrama.

• Interfaces Seriais

As interfaces seriais têm por função transportar dados entre microcontroladores e periféricos de forma serial, ou seja, um único bit é enviado por vez no canal de comunicação entre eles.

Para isso a interligação entre a interface e o periférico pode ser realizada por um único fio ou por dois fios TX e RX. Também permite que o microprocessador utilize o mesmo canal para se interligar com mais de um periférico.

Esse tipo de comunicação facilita muita pois diminui a quantidade de conexões físicas entre o microcontrolador e os periféricos simplificando o projeto, montagem, manutenção e custo do sistema.

Existem duas formas de comunicação serial. A primeira é a comunicação assíncrona que não exige o sincronismo do clock entre o receptor e o transmissor e os caracteres carregam seus próprios sinais de sincronismo. De forma simplificada, "assíncrona" significa que a comunicação não utiliza nenhum tipo de recurso para sincronizar o envio/recebimento e que o sincronismo consiste em o dispositivo receptor saber que esta está na hora de receber um dado.

A segunda forma é a comunicação síncrona cuja transmissão contínua de bits exige que os sinais de Clock do transmissor e receptor precisam estar permanentemente sincronizados.

Uma forma possível para isso é enviar o sinal de Clock por um canal (fio). Mas pode ser feita também através da codificação e transmissão do sinal de clock junto com a informação. A comunicação pode ser orientada a caractere, onde as informações são transmitidas caractere por caractere e quando não há informação a transmitir, são inseridos caracteres chamados SYN para manter o sincronismo. Outra forma é a transmissão orientada a bit, onde as informações são agrupadas em quadros de bits com um código delimitador de início e final de quadro.

São exemplos de interfaces seriais assíncrona a UART-Universal Asynchronous Receiver Transmitter, RS232 e RS485, e de comunicação síncrona SPI-Serial Peripheral Interface e I2C-Inter Integrated Circuit.

• Interface UART

. Uma das interfaces de comunicação mais usadas, a UART-Universal Asynchronous Receiver/Transmitter é um protocolo assíncrono para a troca de dados seriais entre dispositivos que pode ser simplex (os dados são enviados apenas em uma direção), half-duplex (os dois lados transmitem, mas somente um de cada vez) ou full-duplex (ambos os lados podem transmitir simultaneamente).

Utiliza dois fios entre o transmissor e o receptor para transmitir e receber em ambas as direções, além do aterramento nas extremidades.

Os UARTs normalmente são usados com os padrões de comunicação serial, como RS-232C, RS-422 e RS-485. Nos microcontroladores são incorporadas interfaces UART para comunicação pelas portas seriais.

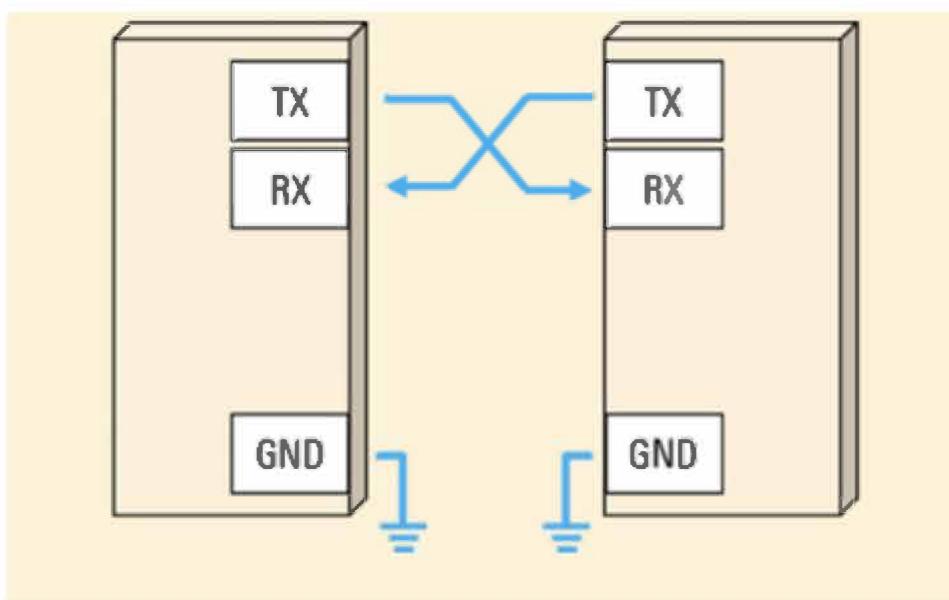


Figura 20 - Diagrama de comunicação serial entre dois dispositivos, mostrando as conexões cruzadas de transmissão e recepção (TX e RX), além da conexão comum de terra (GND).

(Fonte: <https://www.youtube.com/c/Microtecolog%C3%ADa/videos>

A imagem mostra um diagrama de comunicação serial entre dois dispositivos, cada um com três pinos rotulados como TX (Transmissão), RX (Recepção), e GND (Terra). As conexões entre os dois dispositivos são cruzadas: o pino TX de um dispositivo está conectado ao pino RX do outro, e vice-versa. As conexões de GND (Terra) de ambos os dispositivos também estão ligadas, representadas por símbolos de aterramento.

O formato do frame (pacote) de transmissão de uma interface UART é mostrado na Figura 20 – Fonte: Autor- Formato do frame da UART, abaixo.



Figura 20 - Formato do frame da UART (Fonte: Autor)

A imagem ilustra um processo de transmissão serial de dados entre um transmissor (Transmitter) e um receptor(Receiver). No lado esquerdo, o transmissor envia dados a partir do pino TX. O fluxo de dados começa com um bit de início(start), seguido por uma sequência de bits(1, 1, 0, 0, 1, 0, 1) representando a informação transmitida. Após os bits de dados, há um bit de paridade(parity) para verificação de erros e, em seguida, um bit de parada(stop). O fluxo termina com o estado de inatividade(idle), antes de os dados chegarem ao pino RX do receptor.

No início de cada quadro UART, é transmitido um simples bit de início, sempre em 0. Isso assinala o começo da transmissão e prepara a recepção dos dados.

Os bits de dados transportam a informação (neste caso, 1100101). A quantidade de bits pode variar, mas geralmente são 8 bits, podendo ser também 7 ou 6 bits.

A UART pode incluir um bit de paridade no final do quadro para verificação de erro. A paridade pode ser par ou ímpar, significando uma contagem par ou ímpar de bits "1". Um ou mais stop bits em nível 1 são enviados após o bit de paridade para indicar o fim da transmissão.

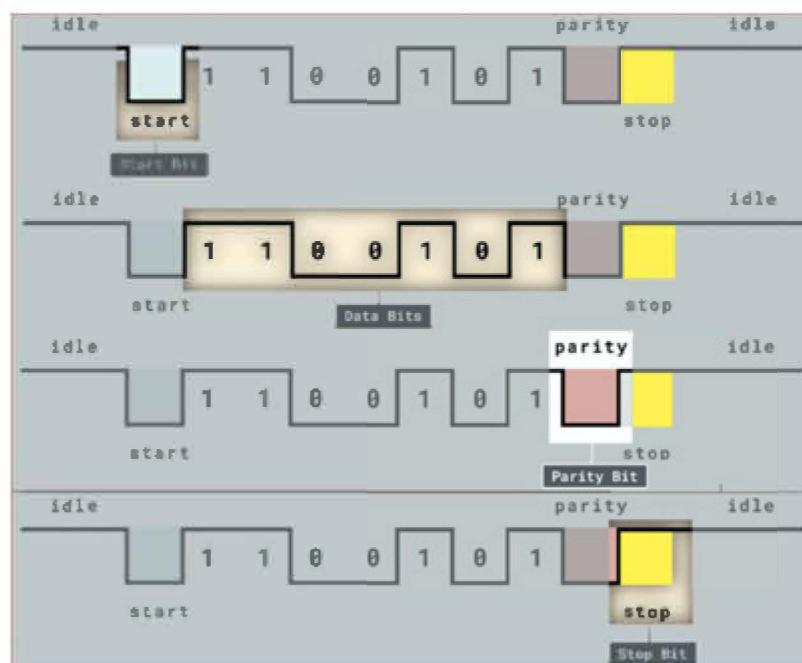


Figura 21 - Transmissão serial na UART (Fonte: Autor)

A imagem mostra um diagrama de comunicação serial em três camadas, destacando o processo de transmissão de bits de dados. Cada linha do diagrama ilustra uma etapa do processo de transmissão.Na primeira linha, o diagrama começa no estado idle(inativo), seguido por um bit de start(início) e uma sequência de bits de dados 11001, e termina com o bit de paridade e o bit de stop(parada), voltando ao estado idle.Na segunda linha, os Data Bits(bits de dados)são destacados no centro, com a sequência 11001. Na terceira linha, o Parity Bit(bit de paridade)é realizado, e o diagrama finaliza com o Stop Bit.

• Interface SPI

Serial Peripheral Interface é uma interface síncrona que permite a comunicação do microcontrolador com diversos dispositivos em rede, sendo a comunicação em modo "full duplex" usando uma arquitetura "master-slave" com um único mestre que é responsável por originar a trama para a leitura e a escrita.

Múltiplos dispositivos escravos são suportados e selecionados por linhas de seleção chamadas SS.

Quando em modo "escravo", o microcontrolador comporta-se como um componente da rede, recebendo o sinal de Clock.

Em modo "mestre", o microcontrolador gera um sinal de relógio e deve ter um pino de entrada/saída para habilitação de cada periférico.

Os pinos de comunicação são MOSI (Master Output Slave Input), MISO (Master In Slave Out), SS (Slave Select) e CLK (sinal de sincronismo Clock).

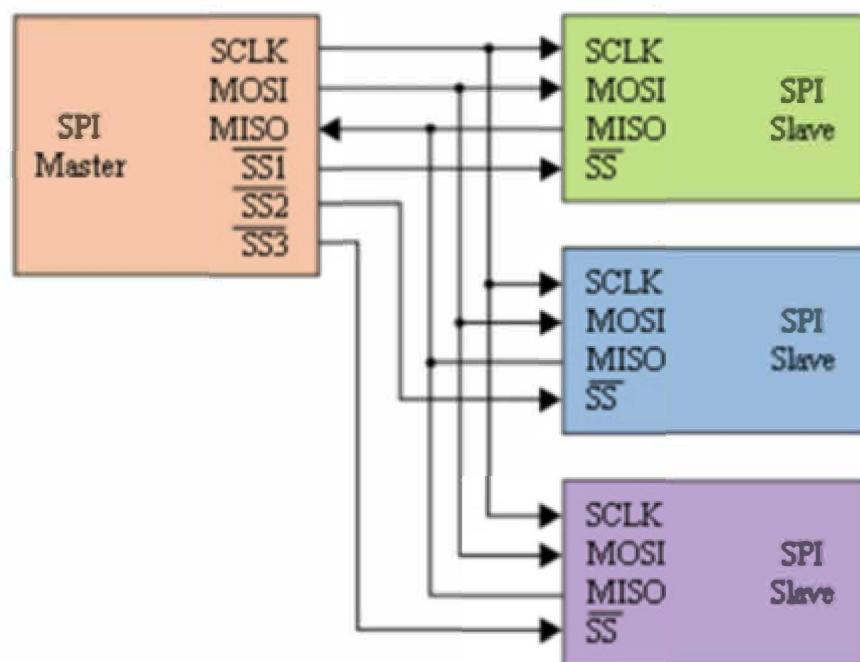


Figura 22 - Master-slave na interface SPI (Fonte: Autor)

A imagem mostra um diagrama de comunicação SPI (Serial Peripheral Interface) entre um SPI Master(Mestre SPI), à esquerda, e três SPI Slaves (Escravos SPI), à direita. O SPI Master possui os seguintes pinos: SCLK(Serial Clock), MOSI (Master Out Slave In), MISO (Master In Slave Out), e três sinal de seleção de escravo: SS1, SS2, SS3. Cada um desses pinos está conectado a pinos correspondentes nos três SPI Slaves, que também possuem pinos SCLK, MOSI, MISO, e SS. O diagrama mostra como o mestre SPI controla os três escravos SPI por meio desses sinal de controle.

• RS485

O padrão RS-485 é uma interface de comunicação serial utilizada para a transmissão de dados em rede seriais cabeados de longa distância (até 1,2Km). Ele é amplamente utilizado em aplicações industriais, comerciais e de automação devido à sua robustez pois é adequada para condições adversas de operação. O canal de comunicação é constituído de par de fio trançados chegando a velocidades superiores a 100Mbps, permitindo a conexão de vários dispositivos na mesma linha.

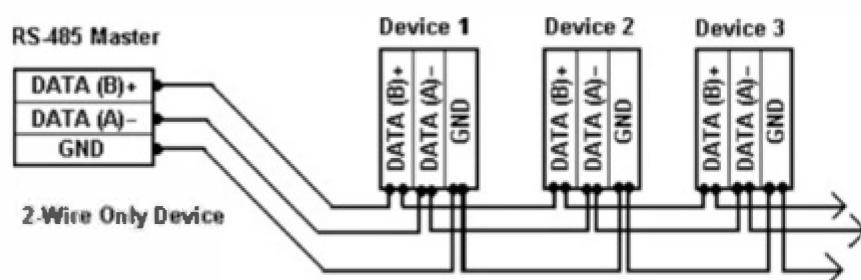


Figura 23 - Interligação de uma rede 485 (Fonte: Autor)

A imagem mostra um diagrama de comunicação em um barramento RS-485. À esquerda, o RS-485 Master tem três conexões principais: DATA (B)+, DATA (A)-, e GND. Esses fios são conectados em paralelo a três dispositivos identificados como Device 1, Device 2 e Device 3, que também possuem os pinos DATA (B)+, DATA (A)-, e GND. A comunicação segue um esquema de duas vias, onde os dados são transmitidos e recebidos pelos dispositivos através dos fios DATA e GND.

• USB

Universal Serial Bus (USB) é um tipo de dispositivo de conexão que foi desenvolvido por companhias do ramo de telecomunicações e de computadores pessoais. O USB, basicamente, tem como objetivo conectar periféricos externos ao sistema embarcado sem que se tenha a necessidade de instalar placas em slots e ainda permite alternar entre periféricos sem ter que desligar o PC (hot plugging, Plug-and-Play). O USB traz ao usuário diversas vantagens como a versatilidade e a facilidade no uso.

A interface USB é mais comumente usada em conexões de periféricos de baixa à média banda, como câmeras digitais, leitores de cartões de memória Flash, joysticks, impressoras e scanners.

Outro benefício que traz o USB é o fato de que permite que o dispositivo tenha tanto alimentação elétrica própria como alimentação via barramento.

São exemplos de padrão a Usb 1.1 - 12 Mbps, a Usb 2.0 - 480Mbps, a Usb 3.1 - 10Gbps e a Usb 4.0 - 20Gbps.

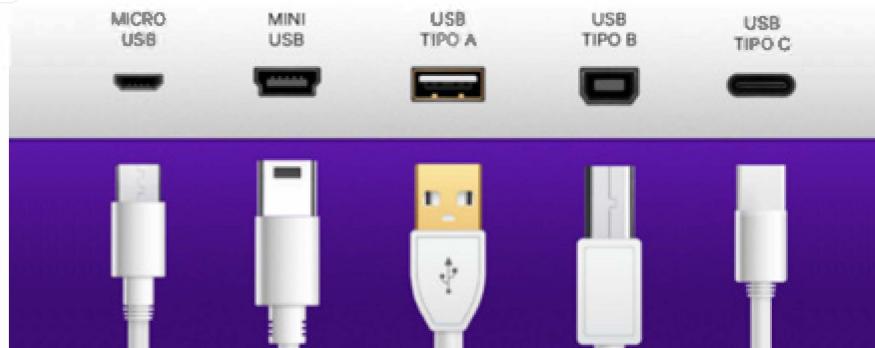


Figura 24 - Conectores utilizados na USB (Fonte: Autor)

A imagem mostra diferentes tipos de conectores USB, dispostos em duas linhas. Na linha superior estão as entradas dos conectores, e na linha inferior, os cabos correspondentes. Da esquerda para a direita, são: Micro USB (pequeno e retangular com bordas arredondadas), Mini USB (maior e trapezoidal), USB Tipo A (retangular e plano, com contatos visíveis), USB Tipo B (quadrado com uma borda superior chanfrada) e USB Tipo C (ovalado e reversível, sem orientação fixa)

• Infravermelho

IrDA - Infrared Data Association, é uma interface de comunicação serial que é feita a partir do envio de pacotes de dados por meio de luz infravermelha. É preciso que o emissor e o receptor estejam em certa proximidade, para que esse feixe de luz seja reconhecido entre os dispositivos. O transmissor precisa fazer o processo de emissão da luz e o receptor precisa ter um sensor sensível a luz.

A segurança é sua grande vantagem, pois é exigida a proximidade dos dispositivos, a transmissão é direcional e não ultrapassa paredes.

Possui baixo consumo energético em seu uso, não sofre interferências eletromagnéticas, garante maior privacidade na troca de dados, baixo custo de implementação, facilidade de compatibilidade entre dispositivos.

Como desvantagem podemos citar comunicação ponto-a-ponto, necessidade de visada direta, cone de incidência estreito (30°), distância máxima de 1 metro e taxa de 600 bps a 115 Kbps.

• Bluetooth

A tecnologia Bluetooth conecta sistemas embarcados e periféricos por meio de transmissões de rádio de curto alcance (10cm a 10m) na faixa de frequência de 2,4 GHz, com vários canais que os dispositivos podem aproveitar para comunicação.

Atualmente, a tecnologia Bluetooth é oferecida nas versões Bluetooth Classic e Bluetooth Low Energy (LE). O Bluetooth Low Energy é otimizado para comunicações intermitentes que preservam a duração da bateria, enquanto o Bluetooth Classic é usado para aplicações que exigem transferência de dados mais frequente e persistente.

O Bluetooth Classic tem até 79 canais de 2,4 GHz à sua disposição, enquanto o Bluetooth Low Energy tem até 40 canais para alternar.

A interface Bluetooth permite a comunicação de voz e dados, de ponto-a-multiponto (broadcast), suportando rede de até 8 dispositivos, fácil integração com protocolo TCP/IP.



Figura 25 – Ilustração de um smartphone utilizando Bluetooth BLE para se conectar com diversos dispositivos inteligentes, como relógio, óculos, tênis, monitor de batimentos cardíacos, balança e carro, mostrando a versatilidade das conexões sem fio em dispositivos do cotidiano.

(Fonte: Internet <https://www.google.com/search?q=interface+bluetooth+ble>)

A imagem apresenta um smartphone no centro com o ícone de Bluetooth BLE na tela, cercado por diversos dispositivos que se comunicam com ele através de ondas azuis que representam a conexão sem fio. À esquerda, estão ícones de um corredor, um relógio inteligente, um óculos inteligente e um tênis conectado. À direita, aparecem ícones de uma mesa digitalizadora, um monitor de batimentos cardíacos, uma balança e um carro. Todos os dispositivos estão interconectados via Bluetooth com o smartphone.

• LCD

LCD(liquid crystal display) é um painel fino usado para exibir informações como texto, imagens e vídeos. Seu uso inclui monitores para computadores, televisores, painéis de instrumentos e outros dispositivos, que vão desde cockpit de aeronaves, displays em computadores de bordo de automóveis, a dispositivos de utilização diárias, tais como leitores de vídeo, dispositivos de jogos, relógios, calculadoras e telefones.

Temos vários modelos como alfanuméricos: apresentam letras, números e caracteres de pontuação e gráficos que permitem o acesso a cada ponto independentemente.

Possui vários tamanhos com ou sem iluminação (backlight) podendo ser acessada via interface serial.

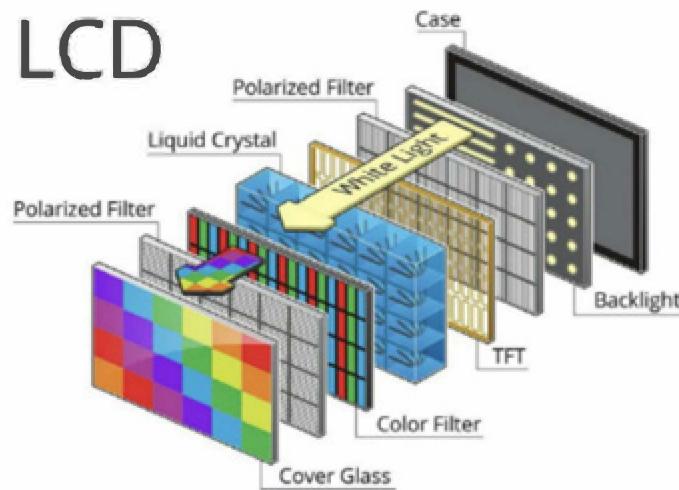


Figura 26 – Estrutura interna de uma tela LCD, destacando as camadas de cristal líquido, filtros polarizados, transistores de filme fino, filtros de cor e luz de fundo, responsáveis pela formação das imagens.

(Fonte: <https://www.dakenchem.com/everything-you-ever-wanted-to-know-about-lcd-materials/>)

A imagem ilustra a estrutura interna de uma tela LCD (Liquid Crystal Display), mostrando suas várias camadas. Da frente para trás, temos o Cover Glass (vídeo de proteção), seguido pelo Color Filter(filtro de cor), o TFT (transistor de filme fino), o Liquid Crystal (cristal líquido), e um Polarized Filter(filtro polarizado). Atrás dessas camadas está a Backlight (luz de fundo), que emite luz branca através das camadas, controlada por um segundo filtro polarizado. A estrutura é envolta em uma Case (capa) que mantém todos os componentes juntos.

• Transdutores

Os transdutores são dispositivos que convertem tipos de energia e são muito utilizados como periféricos.

Como exemplo destes, temos os microfones de carvão, dispositivo onde a pressão do ar (onda sonora) desloca o diafragma que muda a densidade de partículas de carvão, variando a resistência elétrica que produz uma onda elétrica similar à sonora.

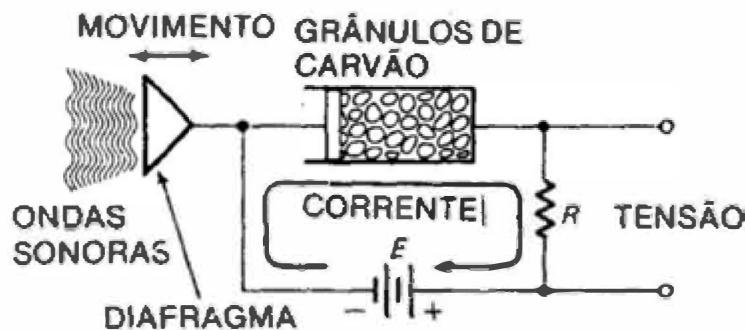


Figura 27 – Esquema de funcionamento de um microfone de carvão, mostrando a conversão de ondas sonoras em variações de corrente elétrica através do movimento de grânulos de carvão em um circuito elétrico.

(Fonte: Internet <https://www.google.com/search?q=microfone>)

A imagem mostra um diagrama simplificado de um sistema de microfone de carvão. À esquerda, o diafragma vibra em resposta às ondas sonoras, e esse movimento é transmitido para uma câmara com grânulos de carvão. À medida que os grânulos de carvão se movimentam, alteram a resistência elétrica do circuito, modulando a corrente elétrica que flui através dele. O circuito é alimentado por uma fonte de energia representada por uma bateria (E), e a variação de resistência causa uma alteração na tensão medida sobre o resistor R, localizada à direita.

• PWM

A interface PWM (Pulse Width Modulation) é bastante aplicada em sistemas embarcados no controle de velocidade dos motores, controle de luminosidade, controle de servo motores e em outras aplicações.

Outros transdutores são os alto falantes, os materiais especiais piezoelétricos, são capazes de se expandir ou contrair de acordo com a voltagem (e vice-versa), os sensores de pressão e peso, termopares que geram voltagens proporcionais à sua temperatura, as células fotoelétricas que geram voltagens ou modificam a resistência proporcionalmente à iluminação sobre elas.

O funcionamento é baseado nos parâmetros período e a largura do pulso (Duty Cycle) de uma onda quadrada que é entregue pelo microcontrolador ao periférico. Variando a largura do pulso, é possível variar a quantidade de potência entregue ao periférico.

Os valores da potência e tensão são disponibilizados conforme a necessidade do circuito. Definimos duty cycle como o percentual do tempo em que o sinal permanece no valor máximo de tensão V_p . A figura abaixo ilustra o funcionamento.

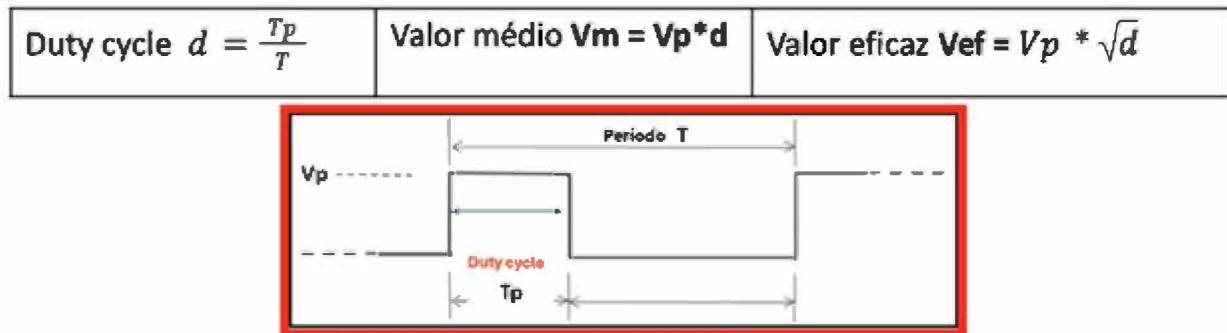


Figura 28 – Fórmulas e gráfico ilustrando o conceito de Duty Cycle, mostrando a relação entre o tempo de pulso, o período total e os valores médio e eficaz de um sinal retangular.
 (Fonte: Autor)

A imagem mostra três fórmulas e um gráfico. A primeira fórmula define o Duty Cycle como a fração do tempo em que um sinal está ativo, dada pela relação entre o tempo ativo (chamado de T_p) e o período total (T). A segunda fórmula apresenta o Valor médio (V_m), que é o valor de pico (V_p) multiplicado pelo Duty Cycle (d). A terceira fórmula calcula o Valor eficaz (V_{ef}), ou valor RMS, como o valor de pico (V_p) multiplicado pela raiz quadrada do Duty Cycle (d). Abaixo das fórmulas, há um gráfico que mostra um sinal retangular, com um valor máximo chamado V_p . O tempo que o sinal permanece em nível alto é denominado T_p , e o intervalo completo de um ciclo é chamado de T . O gráfico também destaca a proporção em que o sinal está ativo, representando o conceito de Duty Cycle.

7. Aplicação dos Sistemas Embarcados

Como você já viu, os sistemas embarcados estão sendo utilizados cada vez mais nas diversas áreas da vida. Como são muitas áreas, relacionamos alguns exemplos de aplicações nas áreas da agricultura, da indústria, da segurança, da logística e da medicina.

• Indústria

- Sistema de Automação Residencial Controlado por Sistemas Embarcados;
- Sistema embarcado para berço inteligente para monitoramento de bebês;
- Acompanhamento dos deslocamentos de pessoas por todo o chão da fábrica;
- Visão da distribuição das temperaturas por todo o chão da fábrica utilizando câmeras térmicas;
- Acompanhamento do comportamento dos colaboradores que trabalham ao longo de uma cadeia de produção;
- Monitoramento de visitantes da fábrica;
- Monitoramento e acompanhamento do funcionamento das máquinas na fábrica;
- Reconhecimento de gestos para operar equipamentos;
- Identificação de pessoas através do reconhecimento de íris;
- Nova Telemetria Inteligente Escalável para Sistemas Industriais;
- Medidor de Energia Inteligente para Utilização Eficiente de Energia.

• Segurança

- Uma arquitetura de casa inteligente com sistemas embarcados para cuidados de longo prazo de pessoas com necessidades especiais.

• Agricultura

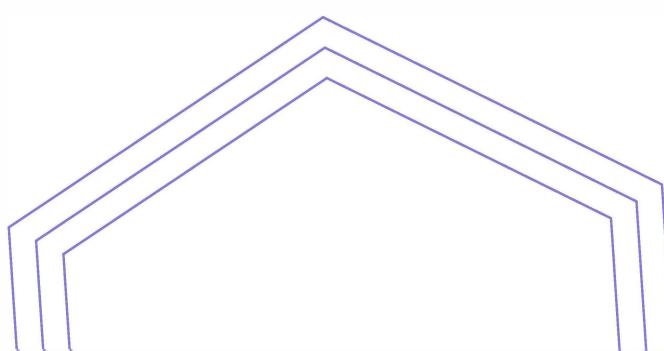
- Sistema embarcado de irrigação automatizado;
- Sistema Embarcado para monitorar cultivo de cogumelos;
- Fábrica inteligente de plantas medicinais.

• **Medicina**

- Monitoramento cardíaco em tempo real usando um sistema embarcado multisensorial;
- Sistema embarcado para monitoramento de saúde com Raspberry Pi;
- Sistema embarcado para detecção de queda humana.

• **Logistica**

- Um sistema de compras inteligente baseado em localização com sistema embarcado;
- Um sistema embarcado para monitoramento de água para edifícios inteligentes;
- Sistema de controle digital de registro de acesso usando RFID e Raspberry Pi;
- Sistema de estacionamento de aeroporto baseado em sistemas embarcados.



8. Conclusão

Caro aluno, aprendemos que um sistema embarcado é composto por hardware e software em camadas, incluindo hardware, software básico e firmware. O hardware inclui dispositivos físicos interconectados (MPU, timer, Uart, etc) e pode ser programado em dispositivos lógicos programáveis.

Componentes de Sistemas embarcados

Componentes Digitais (Microcontroladores, coprocessadores, memórias, barramentos, microprocessadores, Application Specific Hardwares).

Sensores (Fotoelétrico, Detectores de sons, Velocidade, Temperatura, Magnéticos, Pressão, Acelerômetros, Giroscópios, Gás, fumaça), Atuadores (Relés , Drivers de Motores, Drivers de LEDs, Emissores de IR, SCR, TRIAC),

Atuadores (Relés , Drivers de Motores, Drivers de LEDs, Emissores de IR, SCR, TRIAC).

Software (Sistema Operacional, Device Drivers, Aplicações (MPEG-x, GSM-kernel)).

Conversores (Conversor Analógico-Digital (ADC), Digital-Analógico).

Por último, mostramos alguns exemplos de projetos de sistemas embarcados propostos para área como agricultura, logísticas, medicina, segurança e indústria.

Agora, procure aprofundar esses conceitos, pois você entrará em uma fase mais avançada do programa e quanto mais domínio desse conteúdo você tiver, mais você progredirá no seu objetivo. Estude através de artigos, livros e vídeos que você pode encontrar na Internet e refaça de forma constantes revisões do material na plataforma.

Continue em frente e obrigado pelo seu empenho até aqui!

9. Referências

- [1] BALL, Stuart R. **Embedded microprocessor systems: real world design.** 3rd ed. Boston: Newnes, 2002.
- [2] WOLF, Wayne. **Computers as components: principles of embedded computing system design.** 2nd ed. San Francisco: Morgan Kaufmann Publishers, 2008.
- [3] TONILOLO, Cristiano Marçal. **Sistemas embarcados.** Londrina: Editora e Distribuidora Educacional S.A., 2018.
- [4] CHASE, Otavio. **Sistemas embarcados.** [S.l.: s.n.], 2007. Disponível em: <http://www.lyfreitas.com.br/ant/pdf/Embarcados.pdf>. Acesso em: 12 ago. 2024.
- UNIVERSIDADE ESTADUAL DE CAMPINAS. **Capítulo 8: Comunicação serial.** Disponível em: https://www.dca.unicamp.br/courses/EA078/1s2004/arquivos/turma_ab/cap8.pdf. Acesso em: 12 ago. 2024.
- WIKIPÉDIA. **Serial Peripheral Interface.** Disponível em: https://pt.wikipedia.org/wiki/Serial_Peripheral_Interface. Acesso em: 12 ago. 2024.
- INTEL CORPORATION. **Como funciona o Bluetooth.** Disponível em: <https://www.intel.com.br/content/www/br/pt/products/docs/wireless/how-does-bluetooth-work.html>. Acesso em: 12 ago. 2024.
- WIKIPÉDIA. **LCD.** Disponível em: <https://pt.wikipedia.org/wiki/LCD>. Acesso em: 12 ago. 2024.



Residência
Tecnológica
em Sistemas
Embarcados

Executores:



INSTITUTO FEDERAL
DO RIO GRANDE DO SUL



INSTITUTO FEDERAL
DO RIO GRANDE DO NORTE



INSTITUTO FEDERAL
DE SANTA CATARINA



INSTITUTO FEDERAL
DO ESPÍRITO SANTO



cepedi



INSTITUTO
HARDWARE BR

Coordenação:



Financiamento:



INSTITUTO FEDERAL
DO RIO GRANDE DO SUL