

Sistemas Embarcados

Sistemas Embarcados

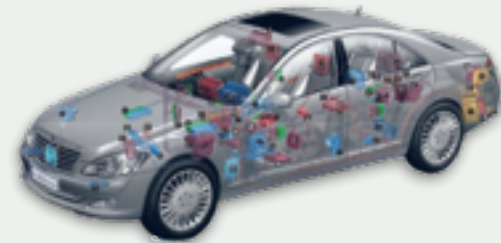
- **Definições:**
 - “Uma combinação de hardware e software, e frequentemente outras partes elétricas e/ou mecânicas, projetado para executar função específica. Em alguns casos, fazem parte de um produto ou sistema maior...” (Barr and Ganssle, 2003)
 - “São sistemas de processamento de informações embarcados em produtos.” (Marwedel, P.; 2011)

Sistemas Embarcados

- **Definições:**
 - "Sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla ... realiza um conjunto de tarefas predefinidas, geralmente com requisitos específicos ... pode-se otimizar o projeto reduzindo tamanho, recursos computacionais e custo do produto." (Heath, 2002)
 - "***Cyber-Physical Systems*** (CPS) are integrations of computation and physical processes". (Lee, 2007)

Aplicações

- Aviação
- Indústria Automotiva
- Telecomunicações
- Robótica
- Automação (Industrial/Comercial/Residencial)
- Eletrodomésticos
- Etc.



Características

- Aplicação bem definida e pré-definida (Contrário aos dispositivos de propósito geral)
- Restrições de memória
- Restrições de tempo-real (Deadline: Hard/Soft real-time)
- Confiabilidade (Tolerância a falhas / Segurança)
- Eficiência energética (operar com baterias)
- Baixo custo (produtos)

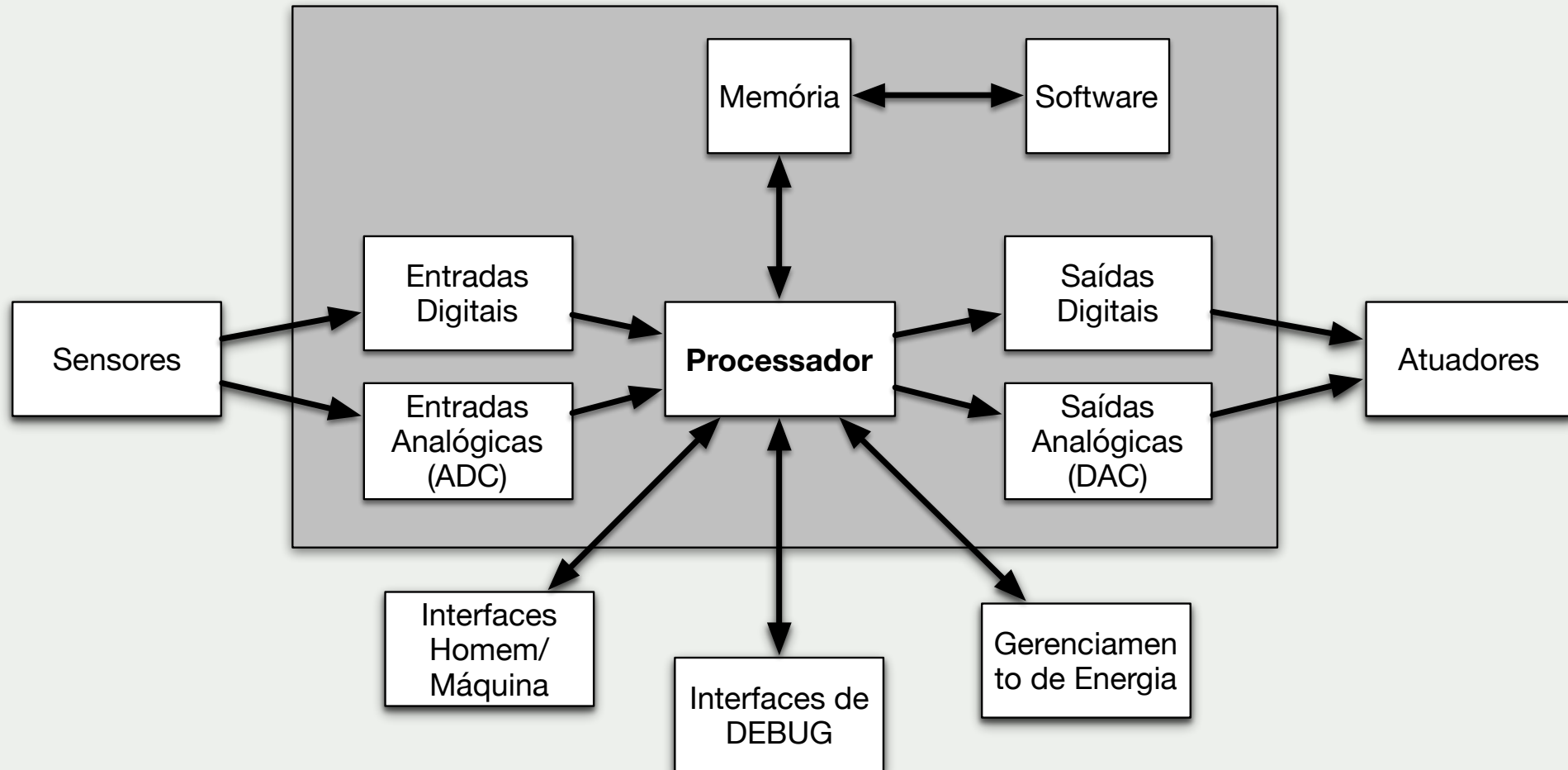
Classificação de Sistemas Embarcados

- **Sistemas similares à Computadores de propósito geral**
 - Video games, Tablets*, Smartphones*
- **Sistemas de Controle**
 - Controle em malha fechada de sistemas em tempo-real
 - Controle de vôo, controle de combustão do motor, reator nuclear
- **Processamento de Sinais**
 - Decodificador de TV digital, Radar, Sonar
- **Comunicação e Redes**
 - Telefone celular*, Roteadores

Funções Gerais do Sistema Embarcado

- Algoritmos de controle
- Lógica sequencial
- Processamento de Sinais
- Interfaces específicas baseadas na aplicação
- Resposta a falhas

Componentes da Arquitetura



Componentes da Arquitetura

- **Hardware**
 - Elemento(s) de Processamento
 - Periféricos
 - Dispositivos de Entrada e Saída
 - Interfaces com Sensores e Atuadores
 - Interfaces de comunicação (Protocolos)
 - Memória
 - Barramento
- **Software**
 - Sistema (OS, RTOS, Compiladores, Simuladores, Emuladores, Ferramentas de Debug, etc)
 - Aplicação

Arquiteturas de Hardware

- Microcontroladores (Ex: PIC, AVR, MSP)
- Processadores de Propósito Geral - GPPs (Ex: Intel, ARM, Athlon)
- DSPs - *Digital Signal Processors* (Ex: TMS___)
- FPGAs - *Field Programmable Gate Arrays*
- ASICs - *Application Specific Integrated Circuits*
- SoCs - *System-on-Chip*
- MPSoCs - *Multiple Processor System-on-Chip*

Software Embarcado

- Correção Lógica
- Correção Temporal
 - Algo correto no tempo errado não tem validade
- Tratar concorrência física (conexão com o mundo externo)
- Confiabilidade e tolerância a falhas são críticas
- Funcionalidade específica para um fim

Software Embarcado

- Multi-tarefa e Concorrência
 - Tratar de várias entradas e saídas e múltiplos eventos que ocorrem independentemente uns dos outros
 - A separação de tarefas simplifica a programação mas exige a mudança constante (multi-tarefa) para que todos os casos sejam tratados
 - Na concorrência, temos a necessidade de executar várias tarefas ao mesmo tempo

Desafios no desenvolvimento de Sistemas Embarcados

- Que tipo de hardware é necessário para o sistema?
 - Que tamanho de palavra do processador?
 - Quanta memória é necessário?
- Como podemos atender aos requisitos de tempo?
 - Hardware mais rápido?
 - Software mais otimizado?
- Como minimizar o consumo de energia?
 - Gerenciar módulos de hardware
 - Combinar tarefas para minimizar o acesso à memória

Projeto de um Sistema Embarcado

<i>Multi-Objetivo</i>
Confiabilidade
Preço Acessível
Segurança
Escalabilidade
Vida Útil

<i>Multidisciplinar</i>
Hardware eletrônico
Software
Hardware Mecânico
Algoritmos de Controle
Interação humana

<i>Ciclo de Vida</i>
Requisitos
Design
Fabricação
Distribuição
Logística
Suporte

Exemplos de Kits de Desenvolvimento

Microcontroladores

Arduino Uno

- Microcontrolador ATmega328
- Clock Speed 16 MHz
- Tensão: 5V
- 14 Pinos Digitais I/O Pins (6 PWM)
- 6 Entradas Analógicas
- 32 KB Flash Memory (ATmega328)
- 2 KB SRAM
- 1 KB EEPROM



Microcontrolador + GPP

Arduino Yún

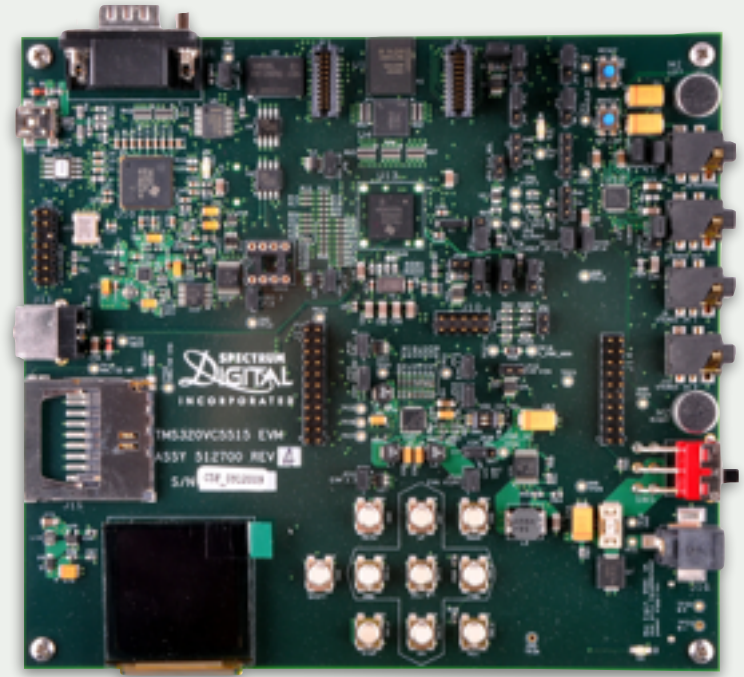
- **Microcontroller ATmega32u4**
 - Operating Voltage 5V
 - Input Voltage (recommended) 5V via microUSB or PoE 802.3af
 - Input Voltage (limits) 6-20V
 - Digital I/O Pins 14
 - PWM Channels 7
 - Analog Input Channels: 6 (plus 6 multiplexed on 6 digital pins)
 - DC Current per I/O Pin 40 mA
 - DC Current for 3.3V Pin 50 mA
 - Flash Memory: 32 KB (ATmega32u4) of which 4 KB used by bootlo
 - SRAM: 2.5 KB (ATmega32u4)
 - EEPROM 1 KB (ATmega32u4)
 - Clock Speed: 16 MHz
- **Embedded Linux machine**
 - MIPS 24K processor operating at up to 400 MHz
 - DDR2 64MB Ram and 16MB SPI Flash
 - Complete IEEE 802.11bgn 1x1 AP or router
 - USB 2.0 host/device
 - poE compatible 802.3af
 - MicroSD card support



DSP

TMS320C5515 DSP Evaluation Module

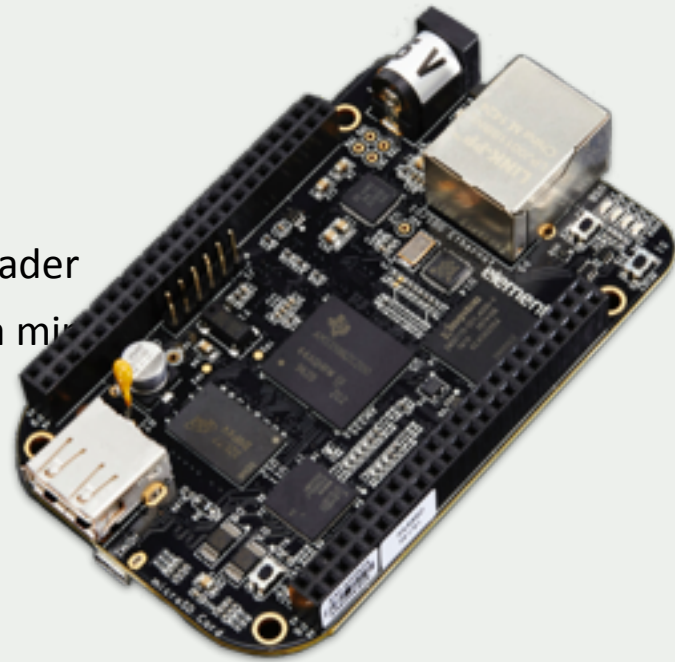
- TMS320C5515 fixed point low power DSP
- On board embedded JTAG emulation to enable the true plug-and-play functionality through just an A-to-mini B USB cable and compatibility of external JTAG emulation interface
- TLV320AIC3204 32-bit programmable low power stereo codec
- OLED color LCD display (128x128 pixels)
- Stereo line in (2) /out (1), headphone out (1) and microphone in (L/R)
- Integrated Flash and mobile SDRAM
- I2C and SPI EEPROMs
- High speed USB 2.0 slave port
- MMC/SD slot, CE-ATA connector, RS232 interface
- 10 user defined push button switches
- Analog front end connectors
- Two expansion connectors for memory cards
- External oscillator socket
- Battery Holder (For 2 AAA, not included)
- Jog dial
- +5V universal power supply



System-on-Chip (GPP + GPU)

BeagleBone Black

- Processor: Sitara AM3358BZCZ100 1GHz, 2000 MIPS
- Graphics Engine: SGX530 3D, 20M Polygons/S
- SDRAM Memory: 512MB DDR3L 606MHZ
- Onboard Flash: 4GB, 8bit Embedded MMC
- Debug Support: Optional Onboard 20-pin CTI JTAG, Serial Header
- HiSpeed USB 2.0 Client Port: Access to USB0, Client mode via mini
- Serial Port UART0 access via 6 pin 3.3V TTL Header.
- Ethernet 10/100, RJ45
- SD/MMC Connector microSD , 3.3V
- User Input: Reset Button, Boot Button, Power Button
- Video Out: 16b HDMI
- Audio Via HDMI Interface, Stereo
- Expansion Connectors:
 - McASP0, SPI1, I2C, GPIO(65), LCD, GPMC, MMC1, MMC2, 7 AIN(1.8V MAX), 4 Timers, 3 Serial Ports, CAN0, EHRPWM(0,2),XDMA Interrupt, Power button, Expansion Board ID



System-on-Chip (GPP + GPU)

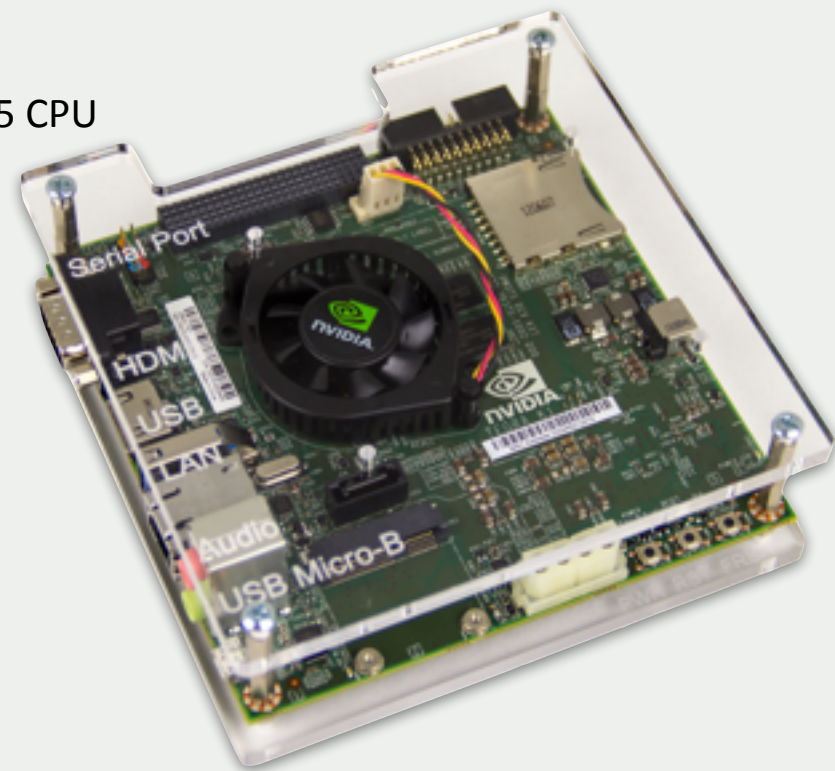
Raspberry Pi 2 B

- Processador: Quad-Core ARM Cortex-A7
- Clock 900 MHz
- GPU: Dual Core VideoCore IV Multimedia Co-Processor (OpenGL ES 2.0), OpenVG, H.264 Decoder
- Memory 1GB LPDDR2
- OS: Linux / Windows (SD Card)
- Ethernet 10/100
- Video: HDMI (1.3 e 1.4)
- Áudio out
- USB 2.0 x 4
- GPIO (40 pin) (+3.3V e +5V)
- Camera and Display Connector



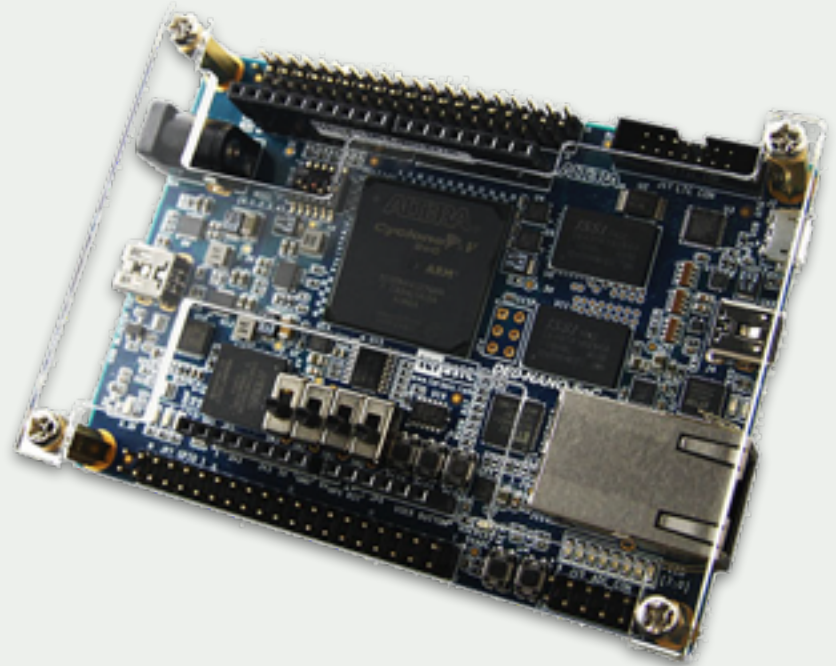
System-on-Chip (GPP + GPU)

- **NVIDIA JETSON TK1**
- Tegra K1 SOC
 - NVIDIA Kepler GPU with 192 CUDA cores
 - NVIDIA 4-Plus-1™ quad-core ARM® Cortex-A15 CPU
- 2 GB memory
- 16 GB eMMC
- Gigabit Ethernet
- USB 3.0
- SD/MMC
- miniPCle
- HDMI 1.4
- SATA
- Line out/Mic in
- RS232 serial port
- Expansion ports for additional display, GPIOs, and high-bandwidth camera interface
- Power supply and cables
- Micro USB-USB



FPGA + GPP + Microcontrolador (SoC)

- **DE0-Nano-SoC**
- **FPGA Device**
 - Altera Cyclone® V SE 5CSEMA4U23C6N device
 - Serial configuration device – EPCS128
 - USB-Blaster II onboard for programming; JTAG Mode
 - 2 push-buttons, 4 slide switches, 8 green user LEDs
 - Three 50MHz clock sources from the clock generator
 - Two 40-pin expansion header
 - One Arduino expansion header (Uno R3 compatibility)
 - One 10-pin Analog input expansion header.
 - A/D converter, 4-pin SPI interface with FPGA
- **HPS (Hard Processor System)**
 - 925MHz Dual-core ARM Cortex-A9 processor
 - 1GB DDR3 SDRAM (32-bit data bus)
 - 1 Gigabit Ethernet PHY with RJ45 connector
 - USB OTG Port, USB Micro-AB connector
 - Micro SD card socket
 - Accelerometer (I2C interface + interrupt)
 - UART to USB, USB Mini-B connector
 - Warm reset button and cold reset button
 - One user button and one user LED



Bibliografia

- Heath, S. (2002). **Embedded Systems Design**, Newnes.
- Barr, M. and Ganssle, J. (2003). **Embedded Systems Dictionary**, CMPBooks
- Lee, E. A. (2007). **Computing foundations and practice for cyber-physical systems: A preliminary report**.
Technical Report UCB/EECS-2007-72, EECS Department, University of California, Berkeley.
- Mardwedel, P. (2011). **Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems**. 2nd Edition, Springer.