Sistemas embarcados e de tempo real

Análise de Sistemas e Requisitos de Software II

Aula 5

Allan Rodrigo Leite

Sistemas embarcados e de tempo real

- Sistema embarcado é um sistema dedicado e especialista constituído de hardware, software e periféricos

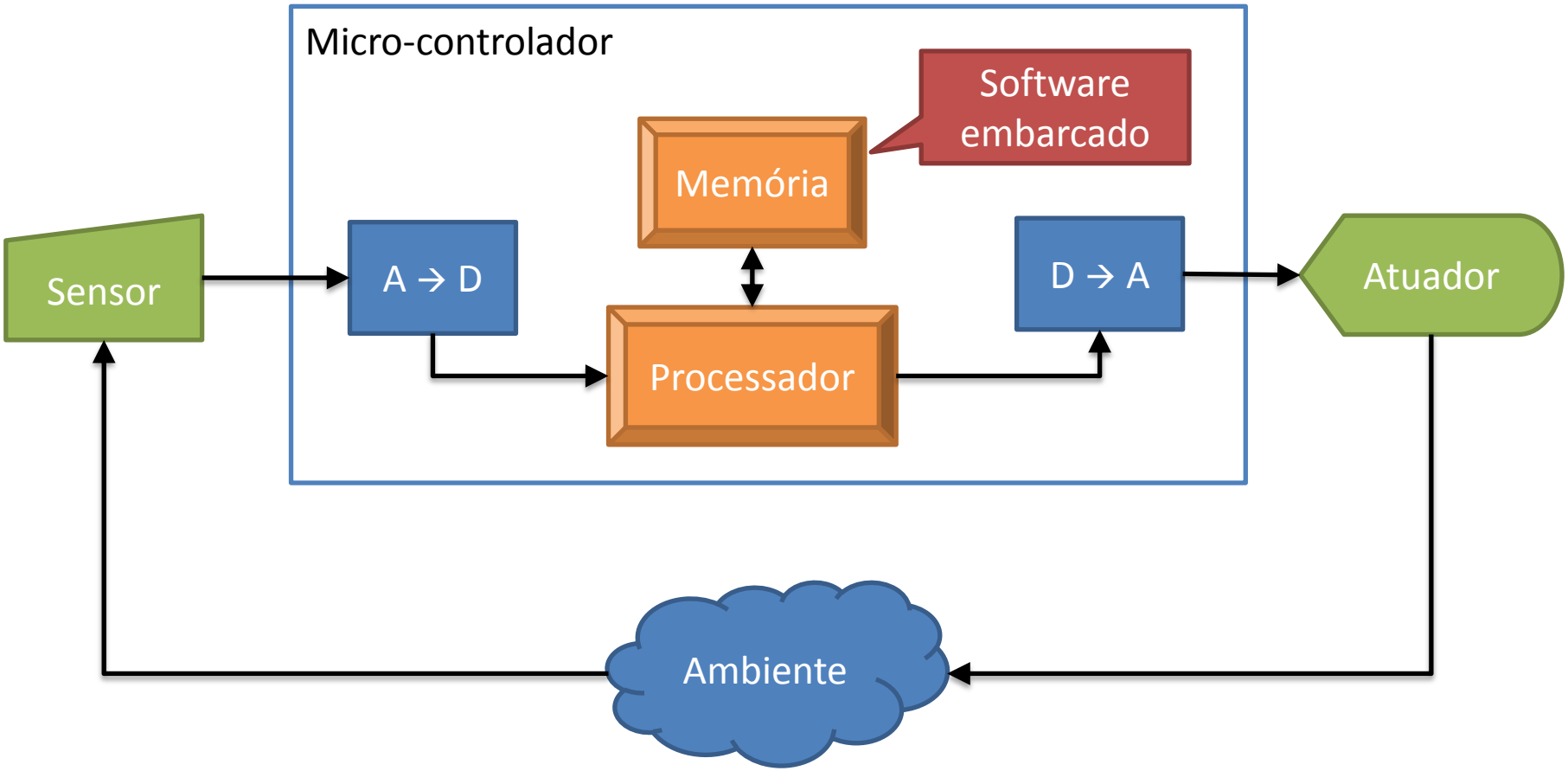
- Sistemas tradicionais são chamados de propósito geral por serem projetados para uma vasta finalidade de uso

Sistemas embarcados e de tempo real

- Um sistema embarcado normalmente é composto por:
  - Microprocessador
  - Memória
  - Interfaces de comunicação
  - Circuito integrado

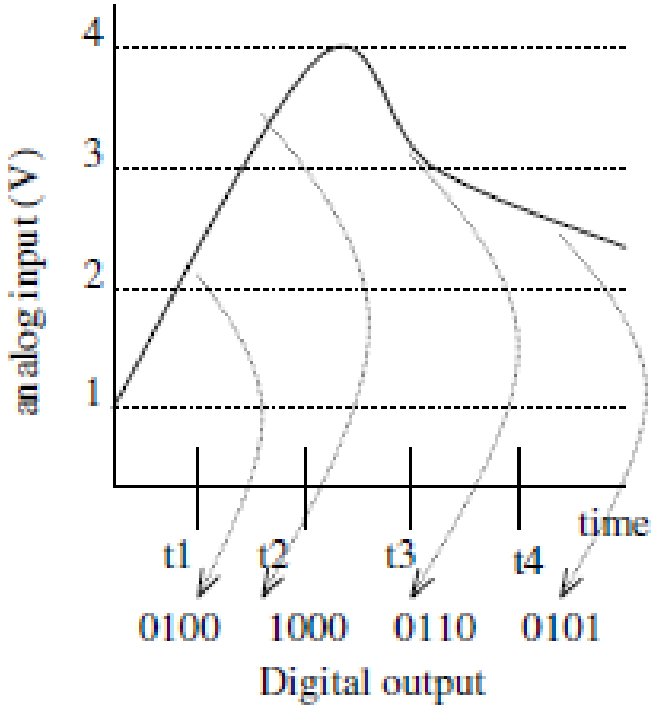
Sistemas embarcados e de tempo real

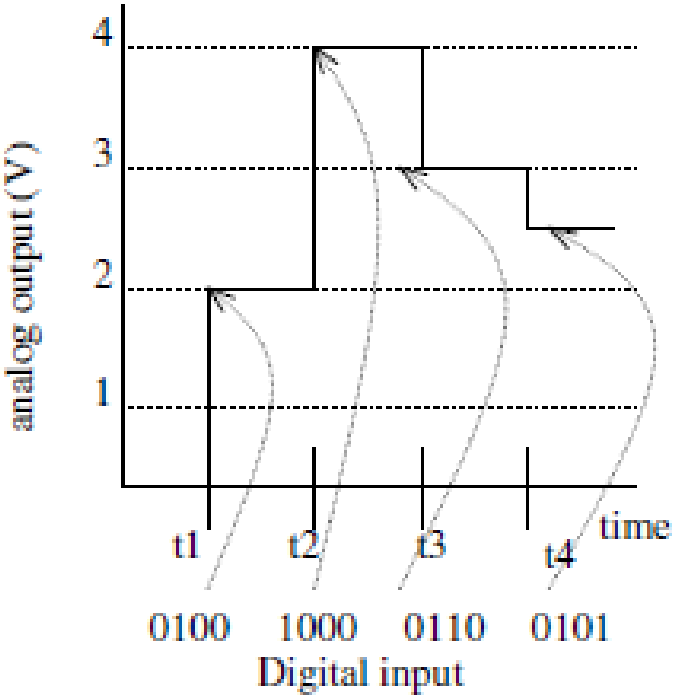
- Arquitetura tradicional de um micro-controlador



Sistemas embarcados e de tempo real

- Arquitetura tradicional de um micro-controlador
  - Conversão sinal analógico para digital





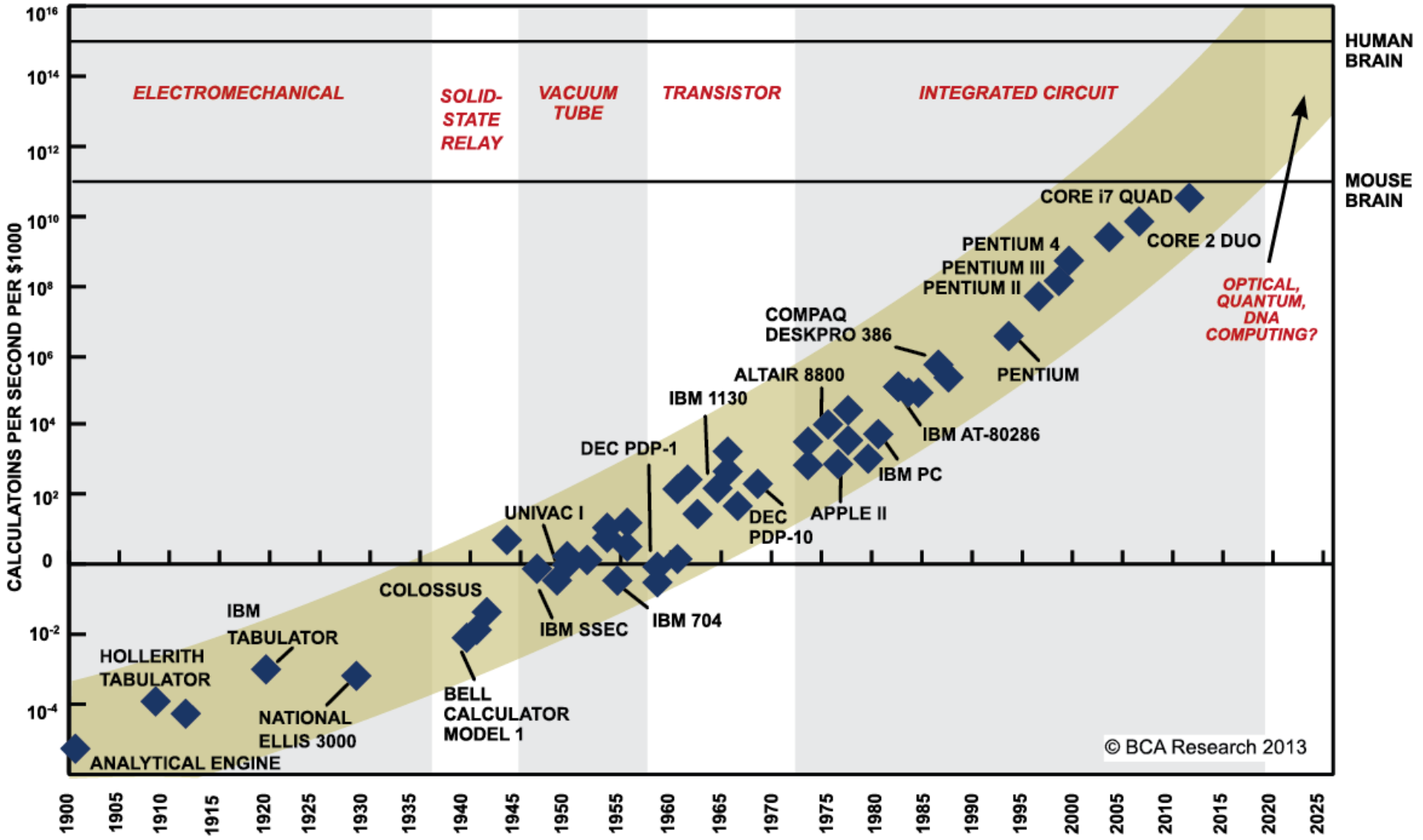
Sistemas embarcados e de tempo real

- Sistemas embarcados atuais seguem a arquitetura System on Chip (SoC)
  - São mais robustos em termos de capacidade física
- Esta arquitetura pode ser estendida para múltiplos processadores
  - Necessidade cada vez maior para processamento em tempo tolerável
- Multipreprocessor System on Chip (MPSoC) não trata apenas de múltiplos processadores
  - Este conceito define que os processadores devem ser otimizados para a aplicação alvo e vice-versa

Sistemas embarcados e de tempo real

- Desafios recentes de sistemas embarcados
  - Lei de Moore: em 18 meses dobra a capacidade de integração de transistores em um circuito
  - Crescimento acelerado de hardware gera um descompasso com software
  - Necessidade de softwares cada vez mais complexos
  - Poucas ferramentas especializadas de apoio

Sistemas embarcados e de tempo real



SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, *THE VIKING PRESS*, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

Sistemas embarcados e de tempo real

- Sistemas de tempo real
  - Adicionam restrições de tempo à execução das tarefas
  - Tanto o hardware quanto o software básico devem estar adaptados a esta necessidade
    - Além do próprio sistema em questão

Sistemas embarcados e de tempo real

- Interfaces e periféricos
  - Maneira como o sistema embarcado captura as percepções e atua sobre o ambiente
- Processamento
  - Sistemas embarcados normalmente utilizam microcontroladores, pela redução de custo e pelos periféricos estarem integrados no mesmo componente

Sistemas embarcados e de tempo real

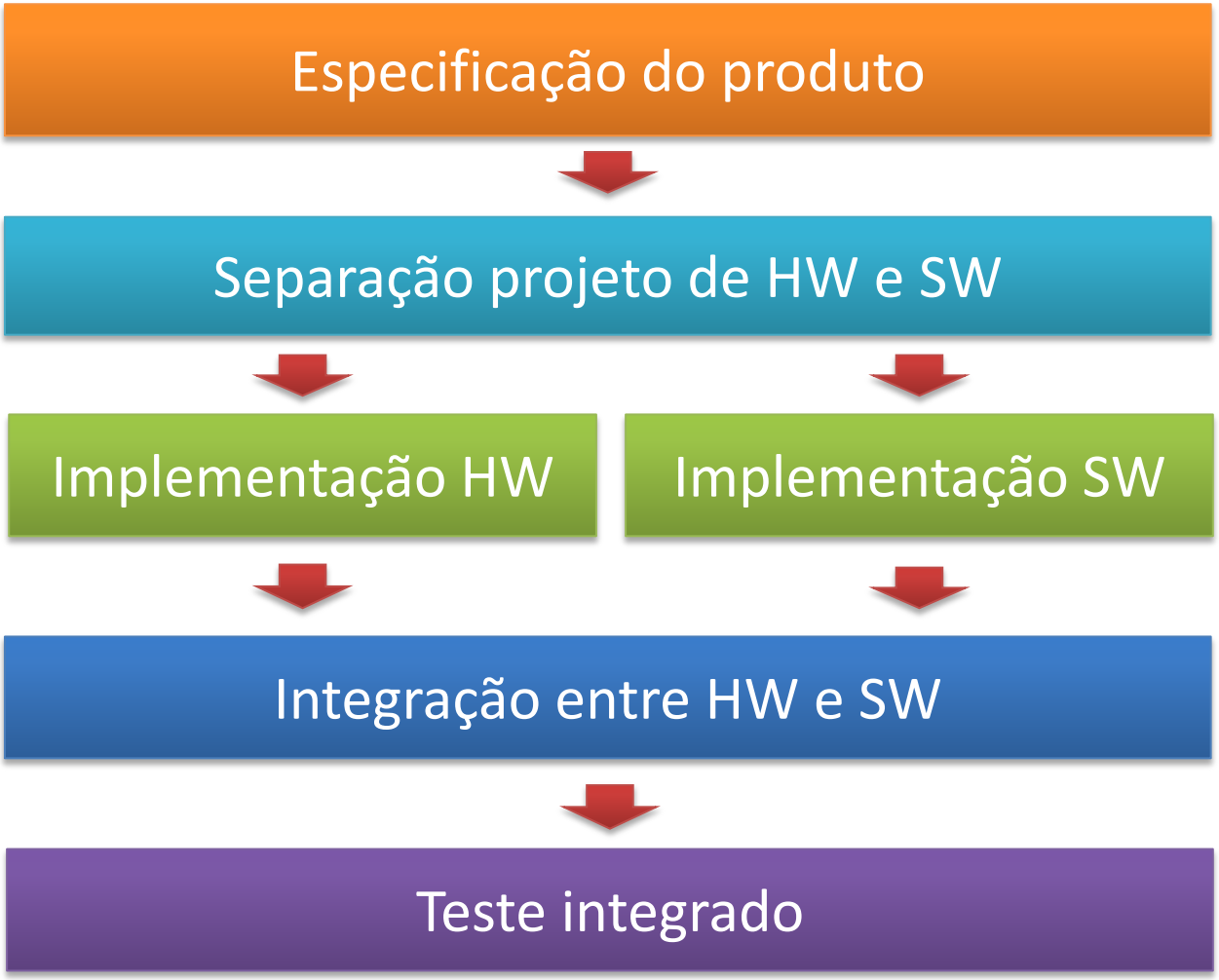
- Projeto de hardware reconfigurável
  - É possível adequar o hardware existente para as necessidades do software
  - FPGA (Field Programmable Gate Array)
- Projeto de hardware específico
  - O hardware é projetado para as características do software
  - ASIC (Application Specific Integrated Circuit)

Sistemas embarcados e de tempo real

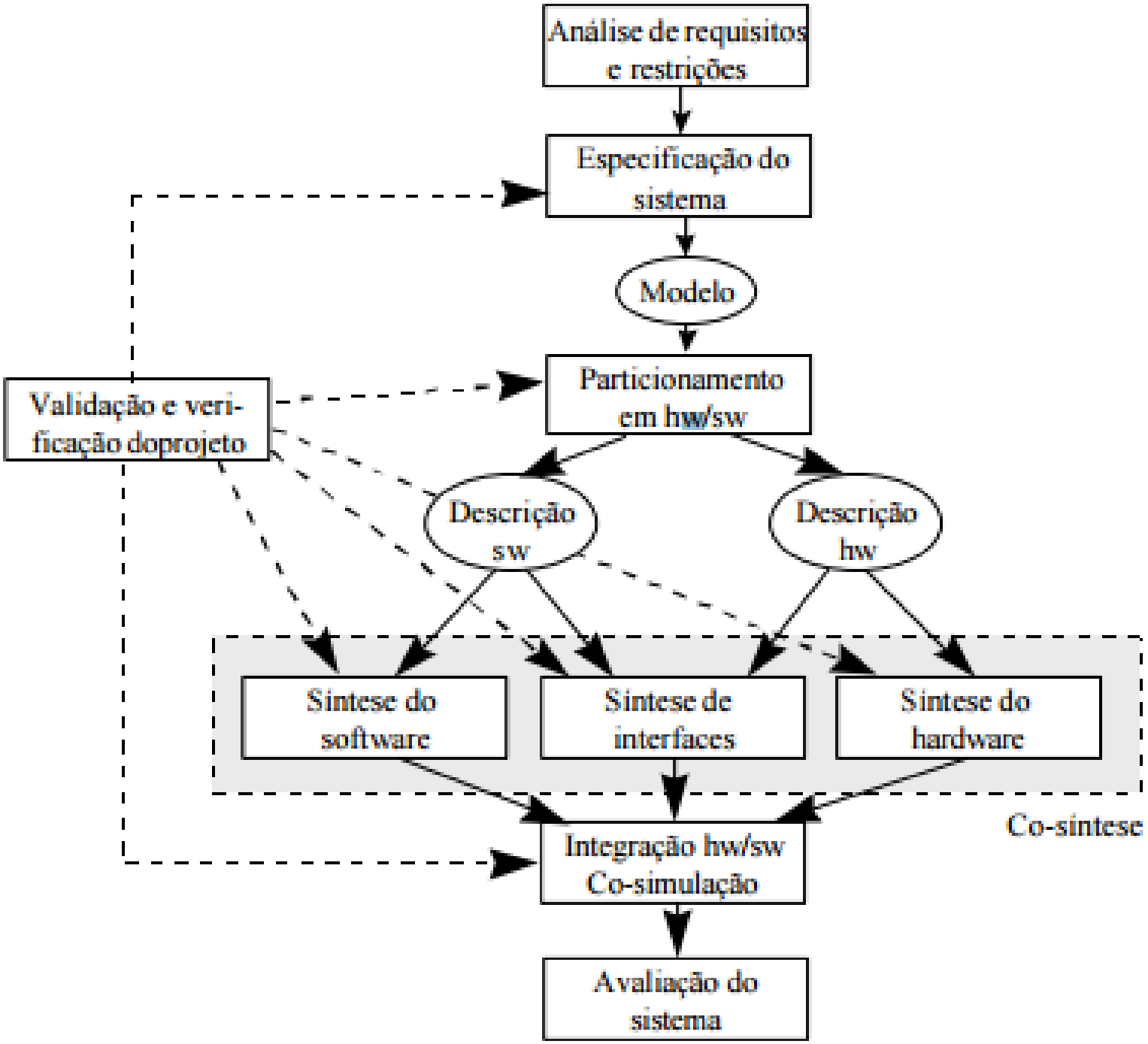
- Disponibilidade e tolerância a falhas
  - Capacidade do software se recuperar de uma falha após ocorrer um erro no sistema
  - Técnicas de notificação são as mais comuns

- Tempo real
  - Capacidade para determinar o tempo de execução de uma determinada operação
  - Normalmente requer que o software seja projetado com esta característica

Sistemas embarcados e de tempo real

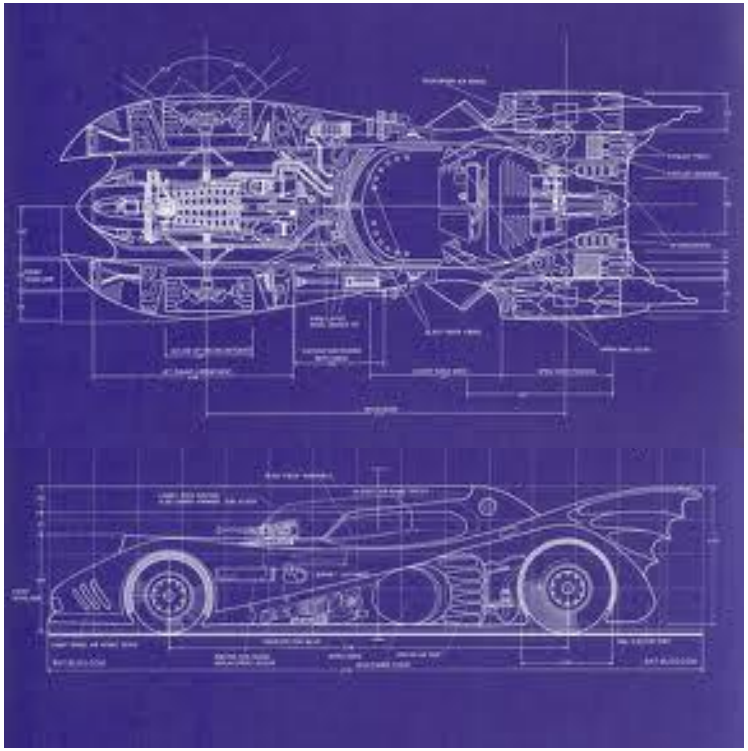


Modelo de co-design



Exemplos de projetos reais de software para sistemas embarcados

- Em um veículo
  - Anti-blocking system (ABS)
  - Central eletrônica
  - Transmissão automática
  - Controle de tração
  - Computador de bordo
  - Aparelhos de som
  - Sensor de estacionamento
  - Sensor de chuva
  - Demais dispositivos



Exemplos de projetos reais de software para sistemas embarcados

- Em uma cozinha
  - Refrigerador
  - Micro-ondas
  - Forno elétrico
  - Lava-louças
  - Demais eletrônicos

Exemplos de projetos reais de software para sistemas embarcados

- Em uma sala de estar
  - Televisão
  - Receptores de TV a cabo
  - Leitores de DVD/Blue-ray
  - Controles remotos
  - Vídeo games
  - Condicionador de ar
  - Demais eletrônicos

Exemplos de projetos reais de software para sistemas embarcados

- Domótica → IOT – Internet of Things
  - Segurança
  - Conforto
  - Economia de energia
  - Comunicação

Gerenciados por um conjunto de sistemas forma descentralizada

Exemplos de projetos reais de software para sistemas embarcados

- IOT não é somente isto
  - Qualquer dispositivo físico utilizado para coletar informações e compartilhá-las
  - Utiliza técnicas de M2M – Machine to Machine communication
    - Fornece um meio de comunicação entre sistemas
    - Suporte a diferentes protocolos de comunicação
      - Rádio frequência (RFID)
      - TCP/UDP e suas variações (HTTP)
      - EDI (arquivos)
  - Restrito quanto a capacidade de processamento e autonomia (energia)

Características de sistemas embarcados

- Usualmente as aplicações em um sistema embarcado são monoprocesso
  - O comportamento do software é definido por meio de iterações

- Cada iteração define uma tarefa específica
  - Determinística
    - Leitura de dados de um sensor
    - Processamento local
  - Não determinística
    - Comunicação (sinal digital/analógico, wifi)
    - E/S de dados (arquivo, serial, USB)

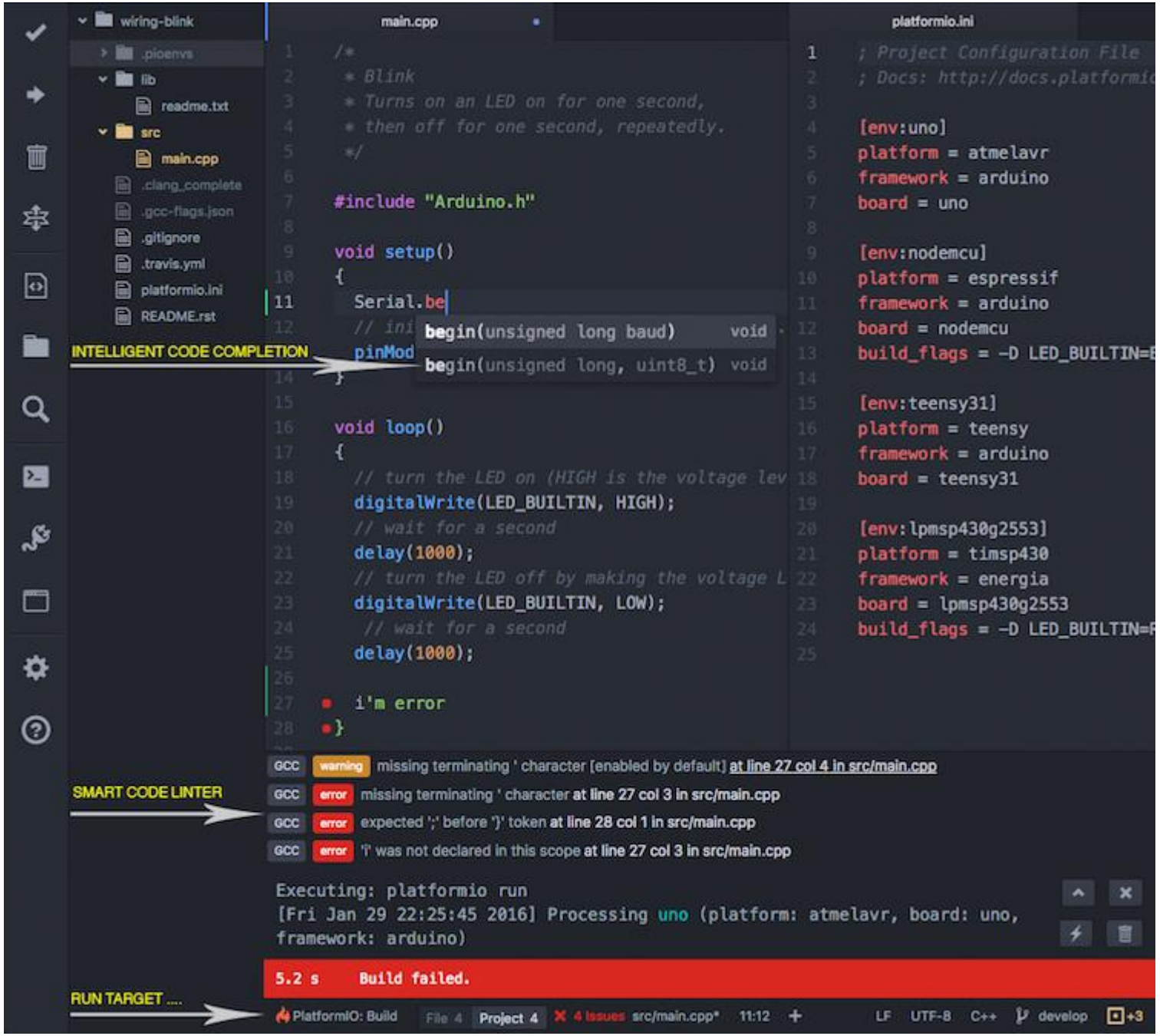
Características de sistemas embarcados

- Temporizadores e contadores
  - Utilizado para definir o fim de cada ciclo de iteração (tempo real)
  - Número de ciclos (clock)
  - Pulsos sobre os dispositivos de E/S

- Watchdog - sistema emergencial
  - Constituído de um dispositivo eletrônico ou temporizador que realiza um reset quando
    - Identificada alguma condição de erro
    - O temporizador alcançou um limite

Exemplos de projetos reais de software para sistemas embarcados

- IDEs para IOT
  - PlatformIO
    - Suporte a várias arquiteturas e plataformas embarcadas
      - Arduino
      - ARM, CMSIS
      - Raspberri, WiringPI
      - Simba RTOS
    - Baseada em CLI – command line interface


wiring-blink
.pioenvs
lib
readme.txt
src
main.cpp
.clang_complete
.gcc-flags.json
.gitignore
.travis.yml
platformio.ini
README.rst
main.cpp
platformio.ini
INTELLIGENT CODE COMPLETION
SMART CODE LINTER
RUN TARGET ....
GCC warning missing terminating ' character [enabled by default] at line 27 col 4 in src/main.cpp
GCC error missing terminating ' character at line 27 col 3 in src/main.cpp
GCC error expected ';' before '}' token at line 28 col 1 in src/main.cpp
GCC error 'i' was not declared in this scope at line 27 col 3 in src/main.cpp
Executing: platformio run
[Fri Jan 29 22:25:45 2016] Processing uno (platform: atmelavr, board: uno, framework: arduino)
5.2 s Build failed.
PlatformIO: Build
Project 4
4 issues
src/main.cpp*
11:12
LF
UTF-8
C++
develop

Exemplos de projetos reais de software para sistemas embarcados

- IDEs para IOT
  - IOT Eclipse
    - Inúmeros plug-ins no Eclipse para desenvolvimento em sistemas embarcados
    - Propósitos mais específicos e bastante voltados para M2M e protocolos de comunicação

Mosquitto

Eclipse Mosquitto provides a lightweight server implementation of the MQTT and MQTT-SN protocols, written in C. The reason for writing it in C is to enable the server to run on machines which do not even have capacity for running a JVM. Sensors and [...]

Read more...

Paho

The Paho project provides reliable open-source implementations of open and standard messaging protocols aimed at new, existing, and emerging applications for Machine-to-Machine (M2M) and Internet of Things (IoT). Paho reflects the inherent physical [...]

Read more...

Eclipse SmartHome

The Eclipse SmartHome project is a framework that allows building smart home solutions that have a strong focus on heterogeneous environments, i.e. solutions that deal with the integration of different protocols or standards. Its purpose is to [...]

Read more...

Leshan

Leshan is an OMA Lightweight M2M (LWM2M) implementation in Java. Eclipse Leshan relies on the Eclipse IoT Californium project for the CoAP and DTLS implementation. It is tested against the LWM2M C client provided by the Eclipse IoT [...]

Read more...

Kura

Kura offers a Java/OSGi-based container for M2M applications running in service gateways. Kura provides or, when available, aggregates open source implementations for the most common services needed by M2M applications. Kura components are designed [...]

Read more...

Eclipse NeoSCADA

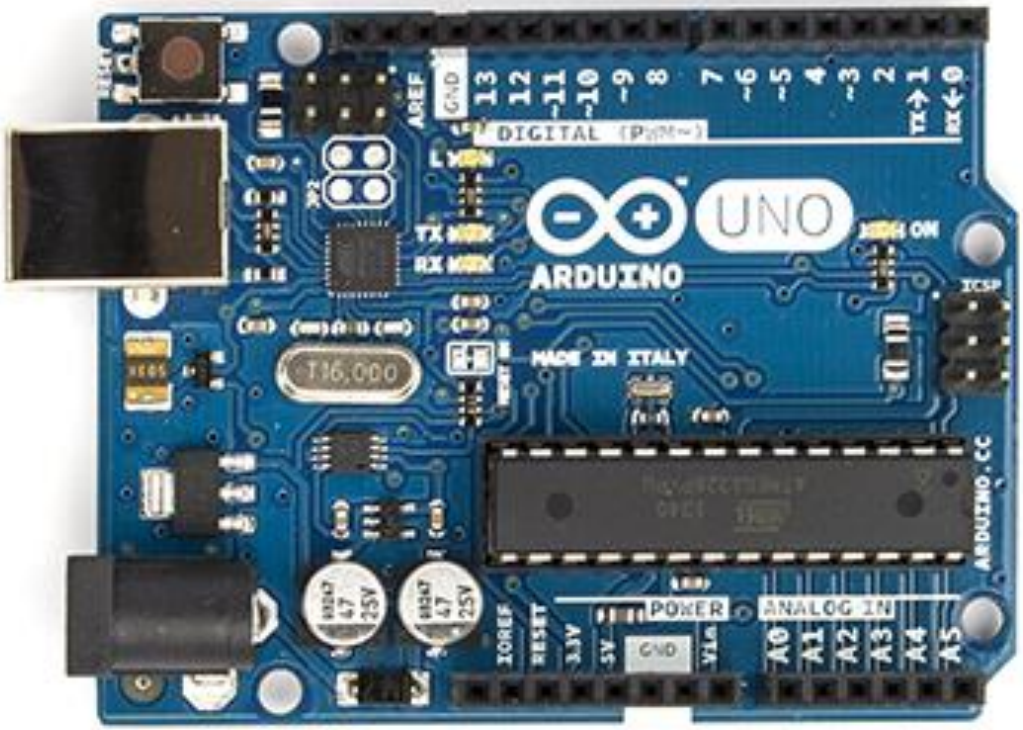
SCADA (supervisory control and data acquisition) is a type of industrial control system (ICS). Industrial control systems are computer controlled systems that monitor and control industrial processes that exist in the physical world. SCADA systems [...]

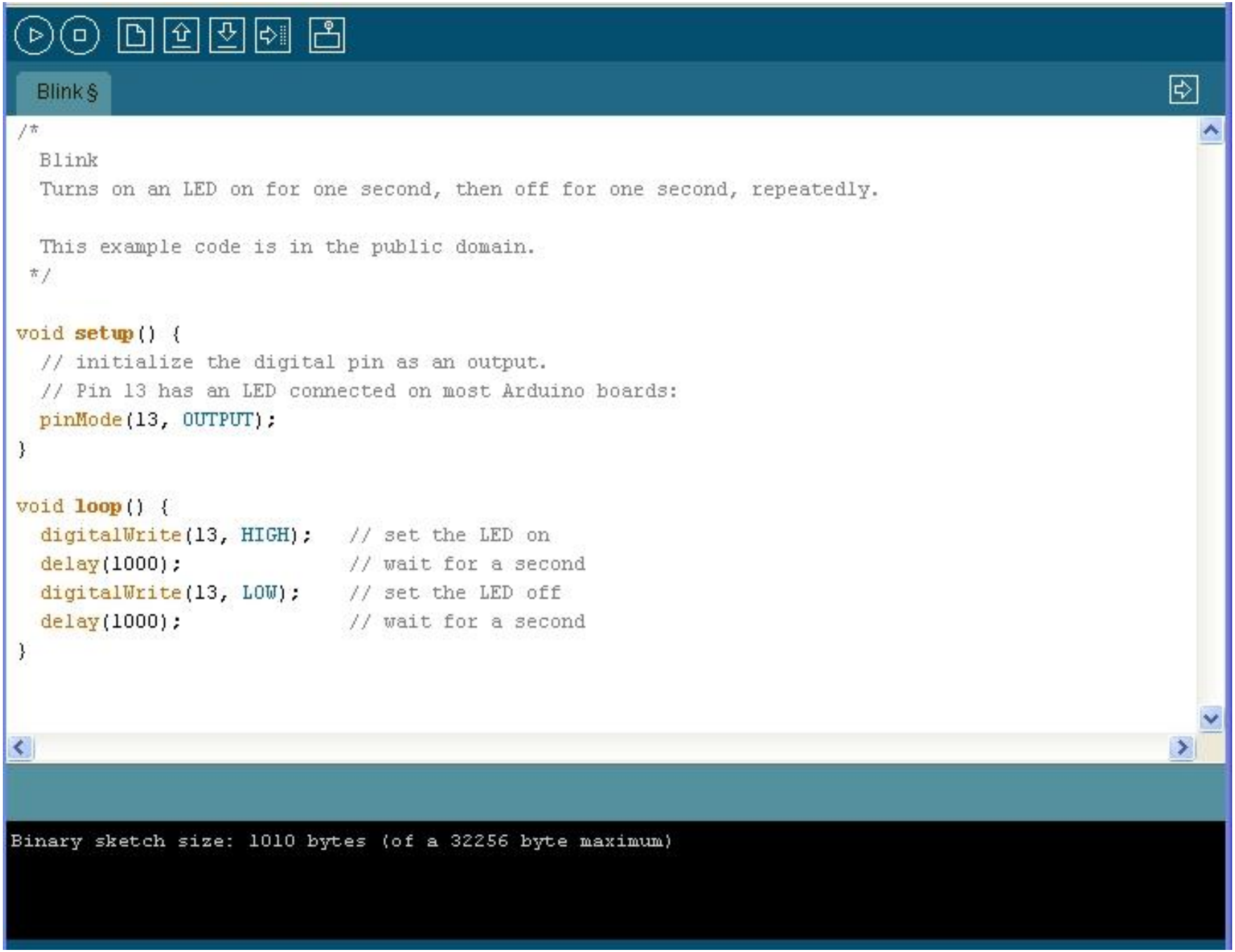
Read more...

Exemplos de projetos reais de software para sistemas embarcados

- Frameworks para IOT
  - Arduino
    - Plataforma aberta
    - Fornece um conjunto de interfaces bem definido para acesso aos componentes conectados ao circuito integrado
    - Utiliza como base linguagem de programação C

Exemplos de projetos reais de software para sistemas embarcados




Blink §
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
This example code is in the public domain.
*/
void setup() {
// initialize the digital pin as an output.
// Pin 13 has an LED connected on most Arduino boards:
pinMode(13, OUTPUT);
}
void loop() {
digitalWrite(13, HIGH); // set the LED on
delay(1000); // wait for a second
digitalWrite(13, LOW); // set the LED off
delay(1000); // wait for a second
}
Binary sketch size: 1010 bytes (of a 32256 byte maximum)

Exemplos de projetos reais de software para sistemas embarcados

- Frameworks para IOT
  - Raspberry PI
    - Mini computador em uma única placa integrada
    - Além dos tradicionais dispositivos de E/S, conta também com pinos GPIO
      - General Purpose Input/Output
    - QEMU para emulação do Raspberry PI

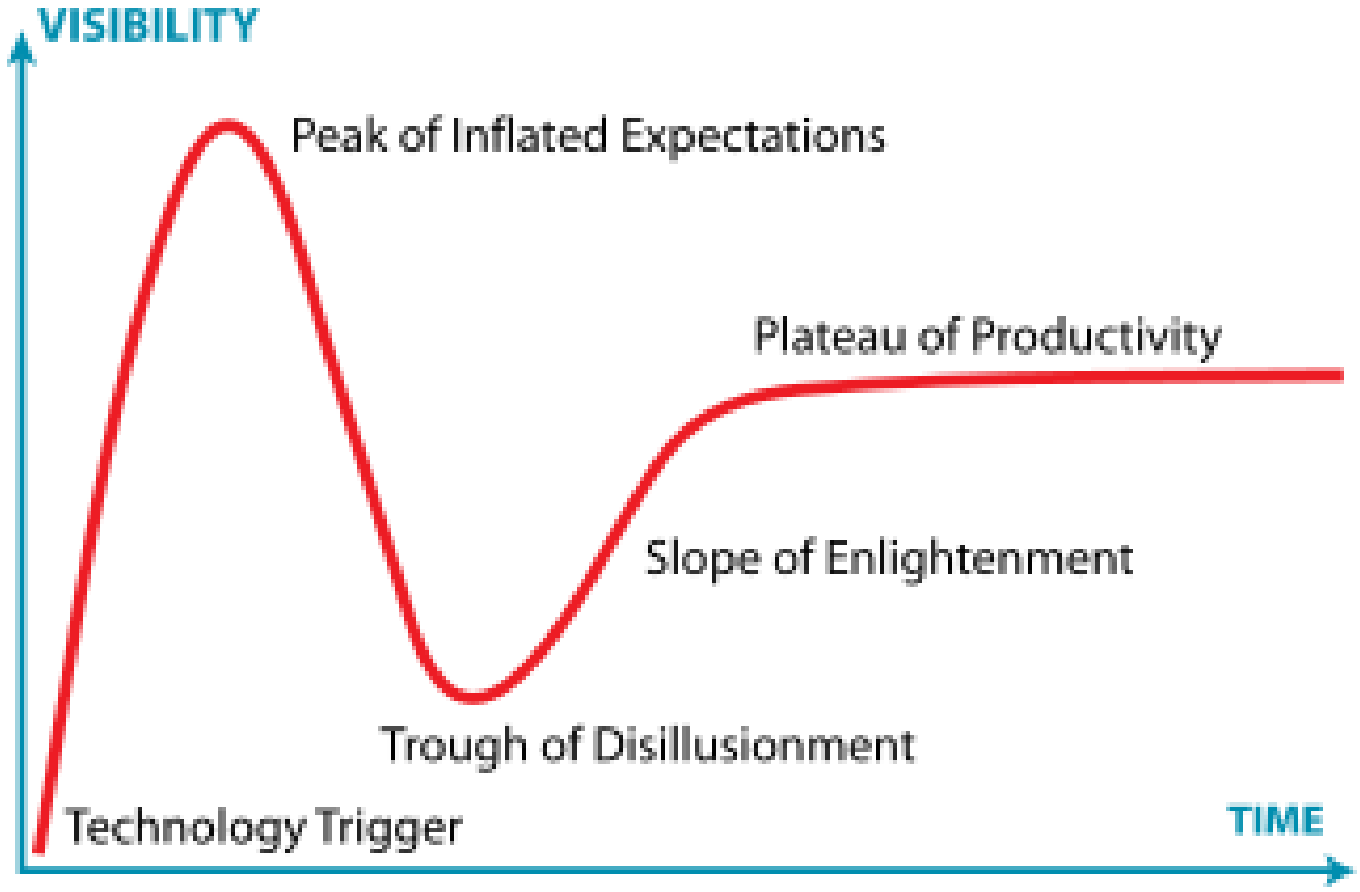
Exemplos de projetos reais de software para sistemas embarcados

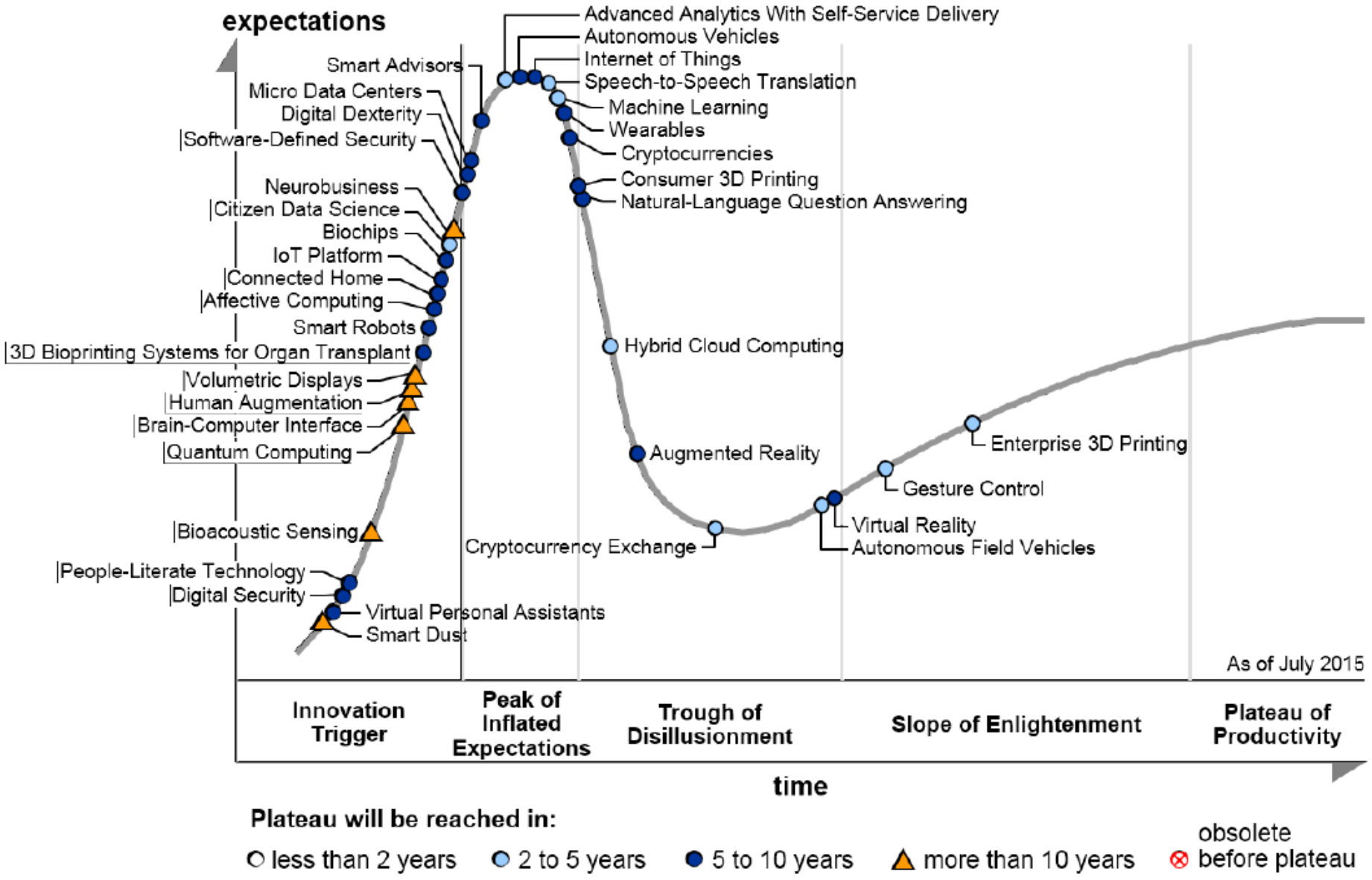
Exemplos de projetos reais de software para sistemas embarcados

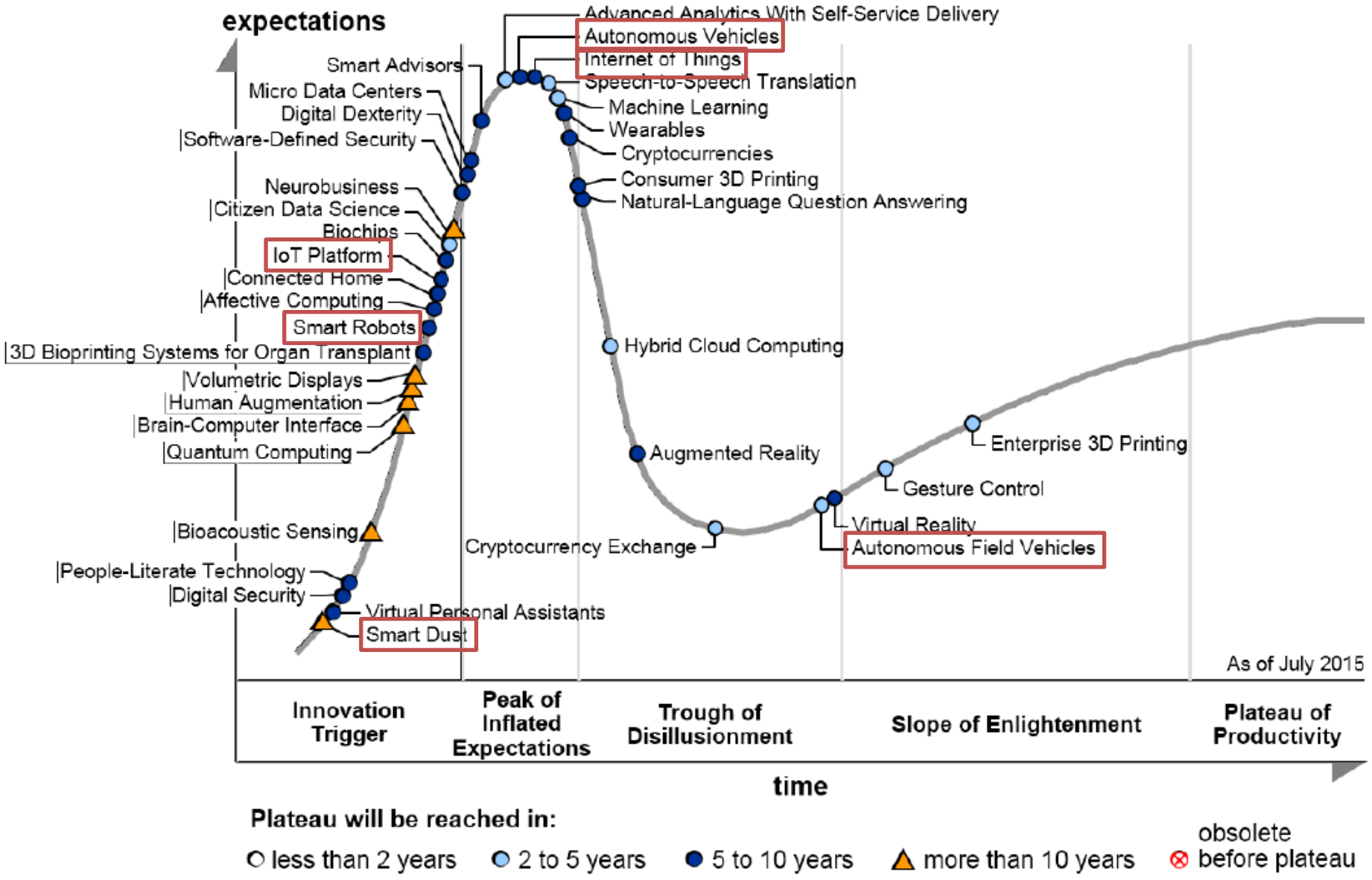
- **Hype Cycle** – Emerging Technologies
  - Estudo anual sobre a **relevância** das **tecnologias** em diversas áreas de pesquisa
  - Representa a **maturidade** das principais tecnologias
    - **Entusiasmo** inicial
    - **Desapontamento** devido a **exagerada expectativa**
    - **Potencial** e **benefícios** práticos
    - **Aceitação** abrangente no mercardo e **estabilização** da tecnologia

Gartner

Exemplos de projetos reais de software para sistemas embarcados




expectations
Advanced Analytics With Self-Service Delivery
Autonomous Vehicles
Internet of Things
Speech-to-Speech Translation
Machine Learning
Wearables
Cryptocurrencies
Consumer 3D Printing
Natural-Language Question Answering
Smart Advisors
Micro Data Centers
Digital Dexterity
Software-Defined Security
Neurobusiness
Citizen Data Science
Biochips
IoT Platform
Connected Home
Affective Computing
Smart Robots
3D Bioprinting Systems for Organ Transplant
Volumetric Displays
Human Augmentation
Brain-Computer Interface
Quantum Computing
Hybrid Cloud Computing
Augmented Reality
Enterprise 3D Printing
Gesture Control
Virtual Reality
Autonomous Field Vehicles
Cryptocurrency Exchange
Bioacoustic Sensing
People-Literate Technology
Digital Security
Virtual Personal Assistants
Smart Dust
As of July 2015
Innovation Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
time
Plateau will be reached in:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau


expectations
Advanced Analytics With Self-Service Delivery
Autonomous Vehicles
Internet of Things
Speech-to-Speech Translation
Machine Learning
Wearables
Cryptocurrencies
Consumer 3D Printing
Natural-Language Question Answering
Smart Advisors
Micro Data Centers
Digital Dexterity
Software-Defined Security
Neurobusiness
Citizen Data Science
Biochips
IoT Platform
Connected Home
Affective Computing
Smart Robots
3D Bioprinting Systems for Organ Transplant
Volumetric Displays
Human Augmentation
Brain-Computer Interface
Quantum Computing
Hybrid Cloud Computing
Augmented Reality
Enterprise 3D Printing
Gesture Control
Virtual Reality
Autonomous Field Vehicles
Cryptocurrency Exchange
Bioacoustic Sensing
People-Literate Technology
Digital Security
Virtual Personal Assistants
Smart Dust
As of July 2015
Innovation Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
time
Plateau will be reached in:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau

Sistemas embarcados e Sistemas tradicionais

- Sistemas embarcados estão evoluindo para:
  - Interface para entrada de dados de sistemas legado
  - Coletam informações do ambiente e as repassam para outros sistemas processarem os dados

- Sistemas tradicionais estão evoluindo para:
  - Arquitetura orientada a serviços
    - SOA, remote objects, Microservices, etc.
  - Tratam sobre o que fazem com os dados coletados
    - Como interpretá-los, processá-los e quais ações serão tomadas

O desafio é: como unir estas abordagens?

Exemplos de projetos reais de software para sistemas embarcados

- CAT – Condução Automática de Trens
  - Treinamento simulado de maquinistas
  - Auxiliar maquinistas em tomadas de decisão durante uma viagem

PUCPR

ALL

DAIKEN

FINEP

CAPES

CAT – Condução Automática para Trens

- Em geral, o projeto de um sistema embarcado requer atenção em:
  - Hardware
    - Dimensões físicas do hardware
    - Consumo de energia
    - Resistência, durabilidade e custo
  - Software
    - Tolerância a falhas
    - Concorrência de recursos
    - Tarefas de tempo real

CAT – Condução Automática para Trens

- Em geral, o projeto de um sistema embarcado requer atenção em (cont.):
  - Hardware será construído para ser dedicado a um determinado algoritmo?
    - ASIC (Application Specific Integrated Circuit)
  - Software utilizará uma plataforma de hardware já definida?
    - FPGA (Field Programmable Gate Array)

CAT – Condução Automática para Trens

- Para o projeto CAT, a arquitetura de hardware já era conhecida
  - CBL 2 (computador de bordo para locomotivas)

- Próximo passo

  Projetar um software compatível com as Características do hardware capaz de auxiliar o maquinista em tomadas de decisão

CAT – Condução Automática para Trens

- Como um trem de carga funciona?
  - Trens são meios de transporte guiados por trilhos
  - Rodas e trilhos são compostos de material metálico para reduzir a resistência de atrito
  - Cada locomotiva possui um motor a combustível que gera energia a motores elétricos de tração
    - Produz esforço trator e potência contínuos

CAT – Condução Automática para Trens

- Como um trem de carga funciona? (cont.)
  - Consumo x potência para uma locomotiva C30

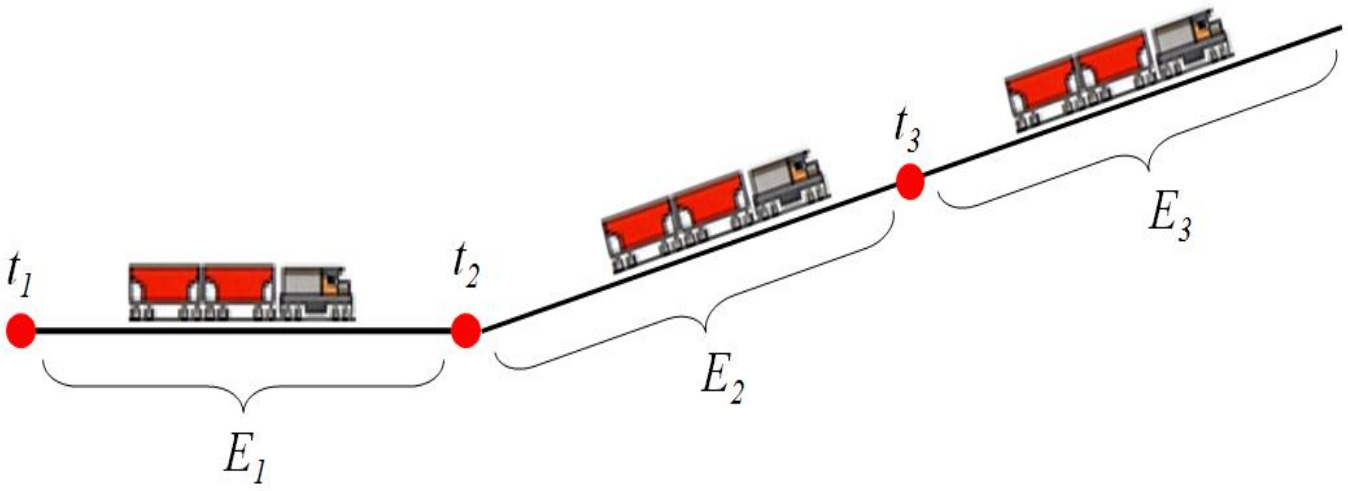
Ponto de aceleração	Potência (HP)	Consumo (l/min)
Marcha lenta	0	0,3168
Freio dinâmico	0	1,767
1	100	0,567
2	275	1,0668
3	575	1,95
4	960	3,033
5	1440	4,533
6	1930	6,183
7	2500	7,6998
8	2940	9,4002

CAT – Condução Automática para Trens

- Como auxiliar o maquinista na tomada de decisão?
  - Indicar qual ponto de aceleração é o mais indicado em uma determinada situação na ferrovia
  - Devido ao fato dos trens serem transportes guiados e das ferrovias serem estáticas, portanto, é possível mapear a geografia da via

CAT – Condução Automática para Trens

- A estratégia adotada consiste de:
  - Sensores do trem indicam a posição na via férrea
  - Computador planeja a melhor ação com base no percurso a ser realizado e evitar ótimo local

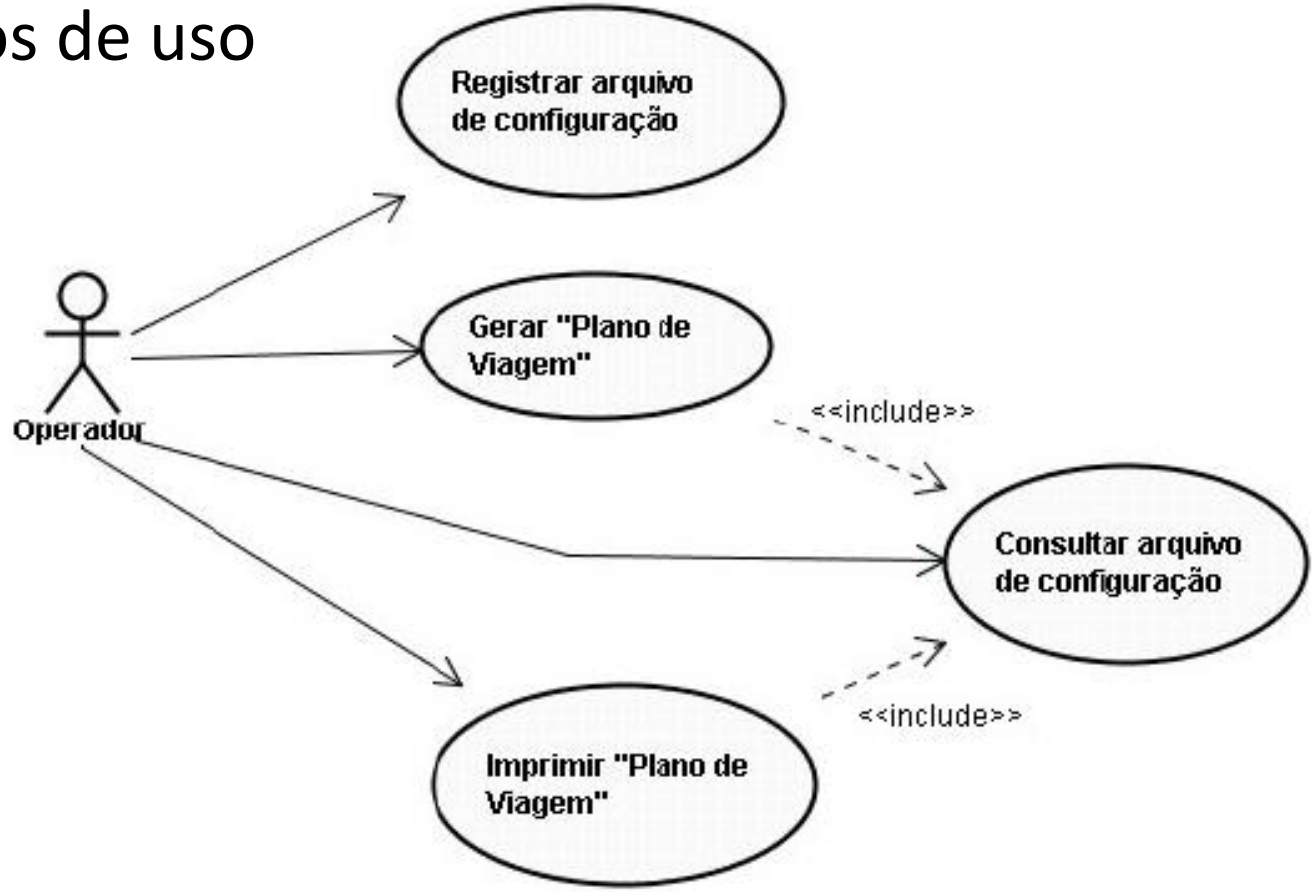


CAT – Condução Automática para Trens

- Alguns dos desafios
  - Simulação da dinâmica do movimento
  - Planejamento das ações
  - Restrição computacional de hardware

CAT – Condução Automática para Trens

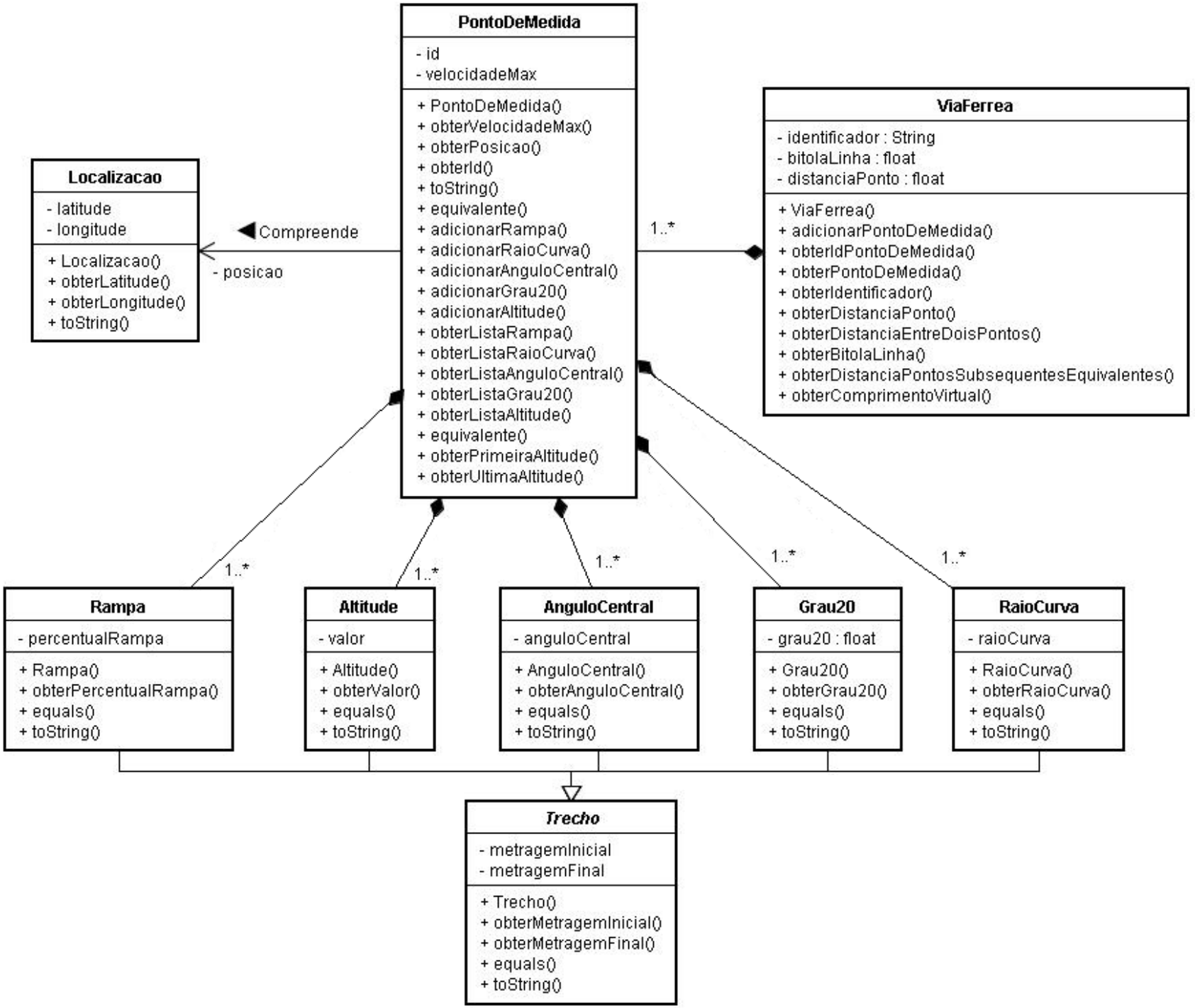
- Levantamento de requisitos
  - Casos de uso



CAT – Condução Automática para Trens

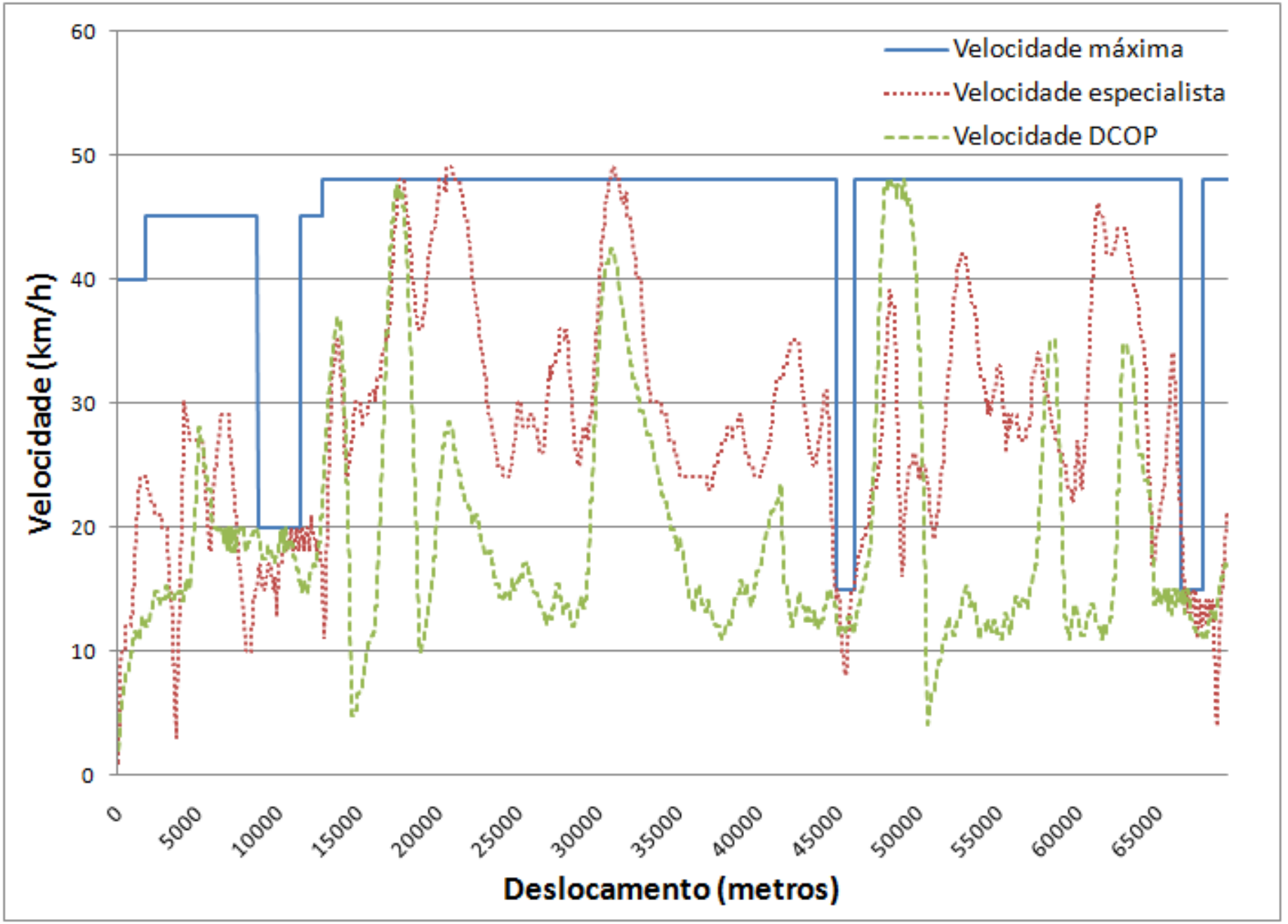
- Especificação
  - Diagrama de classes
  - Diagramas de sequencia
  - Diagramas de atividades

  - Representação da geografia da ferrovia
  - Representação da composição férrea
  - Comportamento da dinâmica do movimento
  - Planejamento das ações futuras


PontoDeMedida
- id
- velocidadeMax
+ PontoDeMedida()
+ obterVelocidadeMax()
+ obterPosicao()
+ obterId()
+ toString()
+ equivalente()
+ adicionarRampa()
+ adicionarRaioCurva()
+ adicionarAnguloCentral()
+ adicionarGrau20()
+ adicionarAltitude()
+ obterListaRampa()
+ obterListaRaioCurva()
+ obterListaAnguloCentral()
+ obterListaGrau20()
+ obterListaAltitude()
+ equivalente()
+ obterPrimeiraAltitude()
+ obterUltimaAltitude()
Localizacao
- latitude
- longitude
+ Localizacao()
+ obterLatitude()
+ obterLongitude()
+ toString()
Compreende
- posicao
1..*
ViaFerrea
- identificador : String
- bitolaLinha : float
- distanciaPonto : float
+ ViaFerrea()
+ adicionarPontoDeMedida()
+ obterIdPontoDeMedida()
+ obterPontoDeMedida()
+ obterIdentificador()
+ obterDistanciaPonto()
+ obterDistanciaEntreDoisPontos()
+ obterBitolaLinha()
+ obterDistanciaPontosSubsequentesEquivalentes()
+ obterComprimentoVirtual()
1..*
1..*
1..*
1..*
1..*
Rampa
- percentualRampa
+ Rampa()
+ obterPercentualRampa()
+ equals()
+ toString()
Altitude
- valor
+ Altitude()
+ obterValor()
+ equals()
+ toString()
AnguloCentral
- anguloCentral
+ AnguloCentral()
+ obterAnguloCentral()
+ equals()
+ toString()
Grau20
- grau20 : float
+ Grau20()
+ obterGrau20()
+ equals()
+ toString()
RaioCurva
- raioCurva
+ RaioCurva()
+ obterRaioCurva()
+ equals()
+ toString()
Trecho
- metragemInicial
- metragemFinal
+ Trecho()
+ obterMetragemInicial()
+ obterMetragemFinal()
+ equals()
+ toString()

CAT – Condução Automática para Trens

- Resultados obtidos
  - Redução na ordem de 20% do consumo de combustível
  - Quanto maior o tamanho dos sub-planos, maior é a economia de combustível, porém, o tempo de resposta também é maior


Velocidade máxima
Velocidade especialista
Velocidade DCOP
Velocidade (km/h)
0
10
20
30
40
50
60
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000
55000
60000
65000
Deslocamento (metros)

Sistemas embarcados e de tempo real

Análise de Sistemas e Requisitos de Software II

Aula 5

Allan Rodrigo Leite