



Introduction to Embedded Systems, Lecture 1

Basic concepts of Embedded Systems

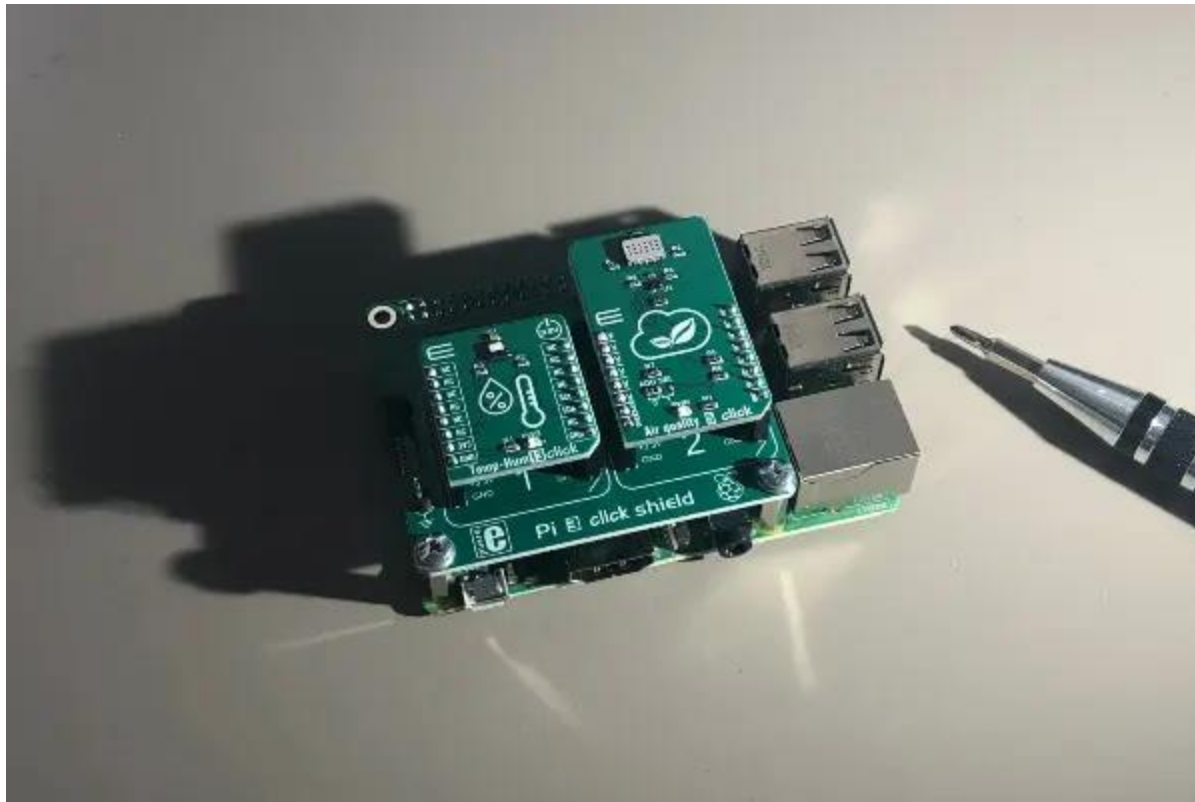
Dmitry Zaitsev

<http://daze.ho.ua>

Embedded System

- An embedded system is a computer (control) system—a combination of a computer processor, computer memory, and input/output peripheral devices—that has a dedicated function within a larger mechanical or electronic system.
- It is embedded as part of a complete device often including electrical or electronic hardware and mechanical parts.

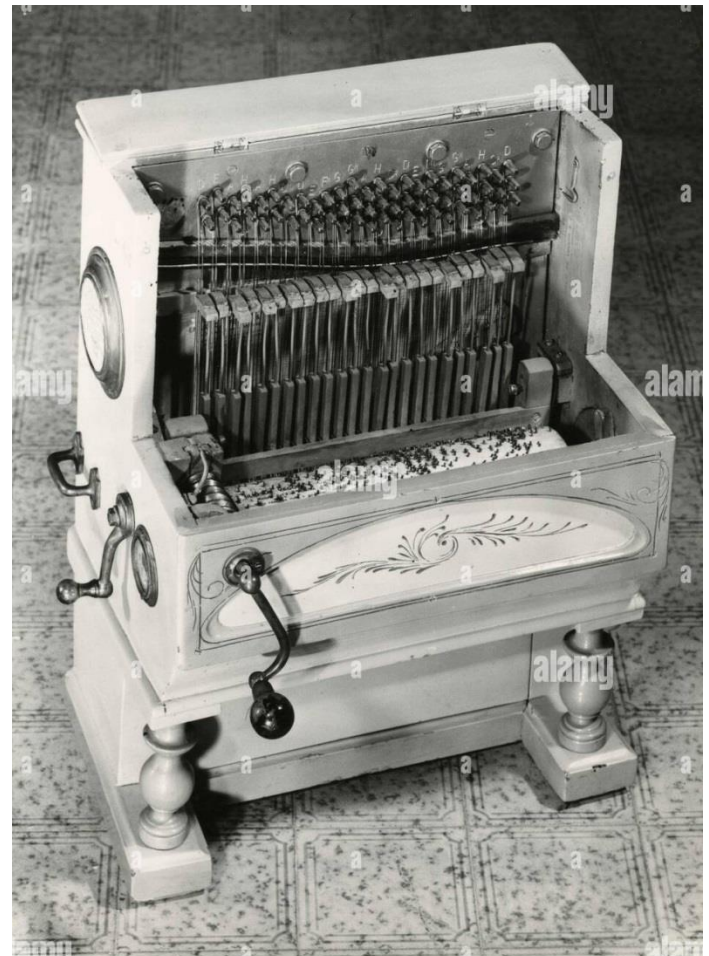
Example of ES



Control Systems

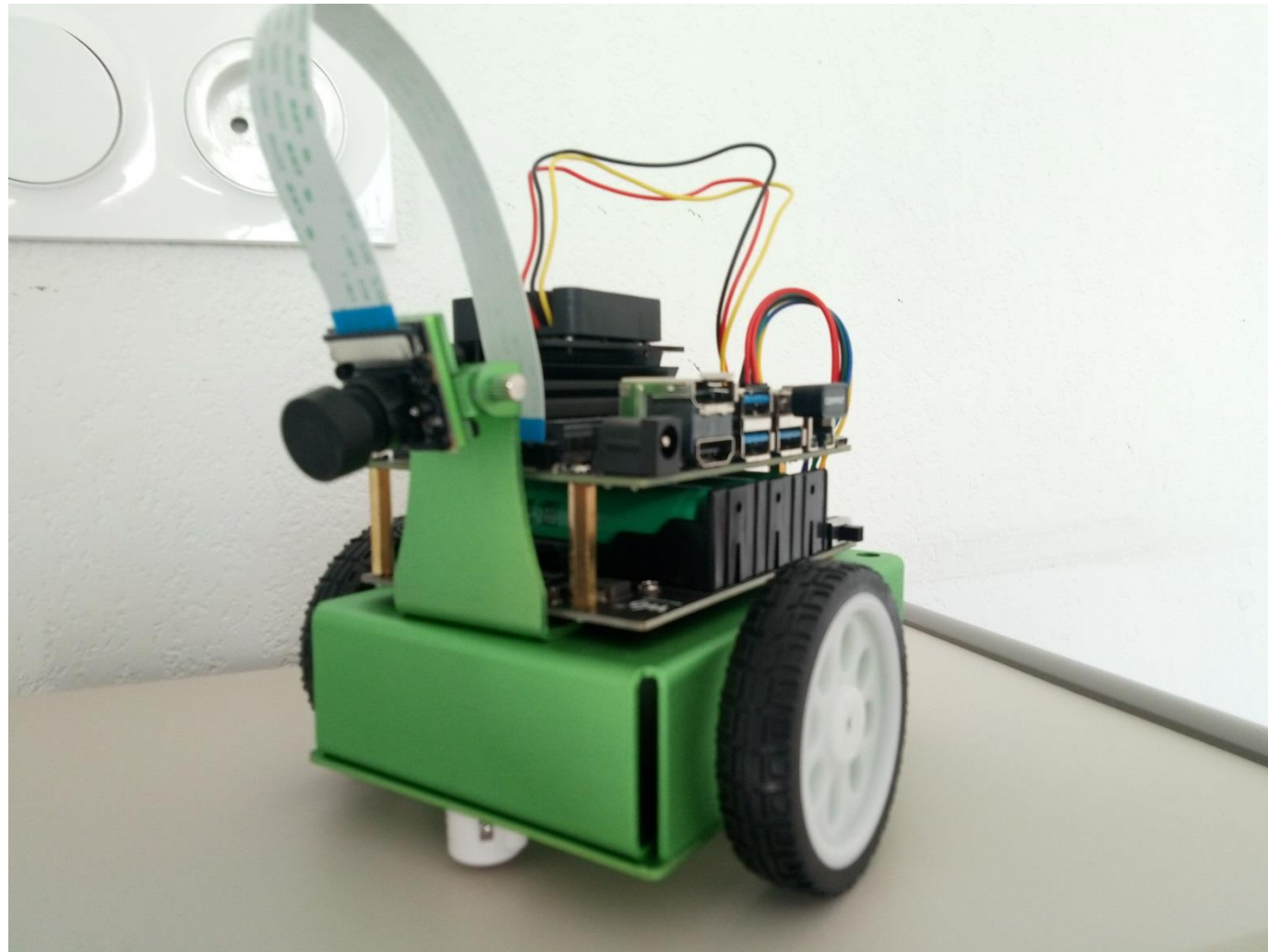
- Mechanical
- Electro-Mechanical
- Electronic Analog
- **Electronic Digital**
 - Logic and automata
 - Microcontrollers
 - Artificial Intelligence – Machine Learning

Mechanical music box

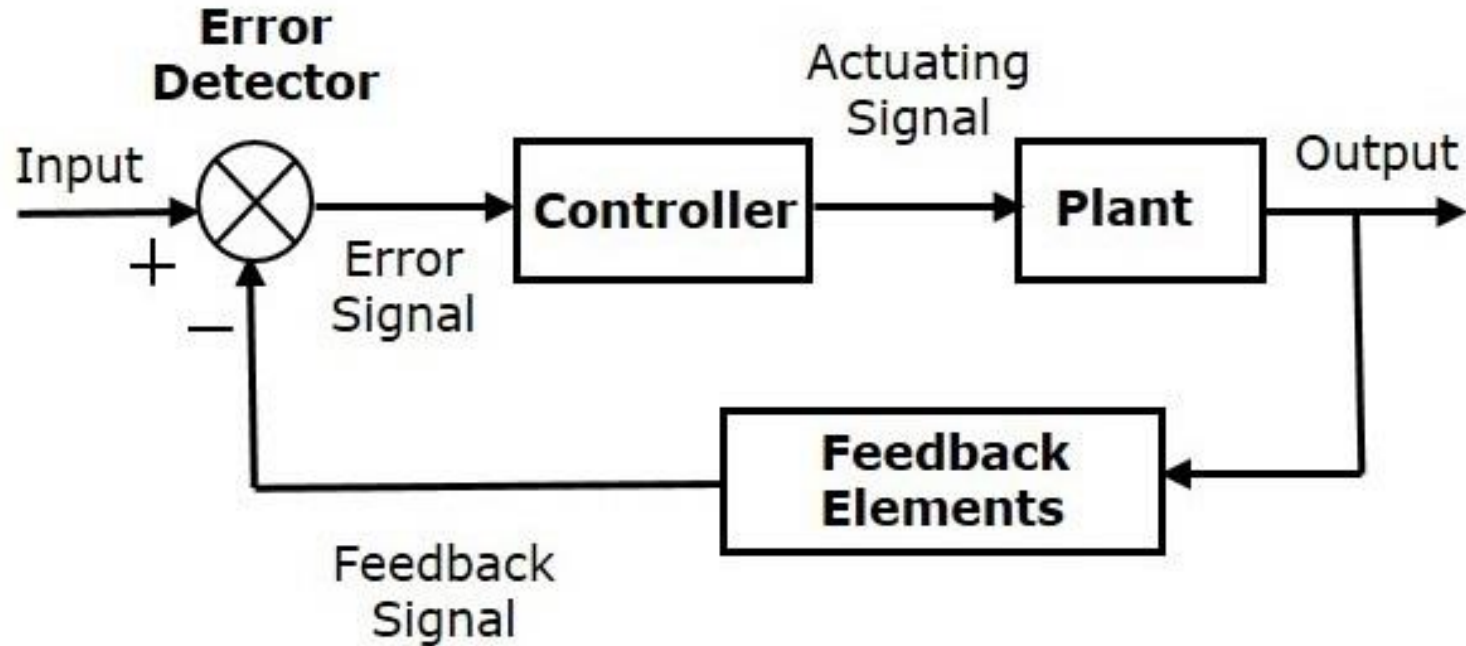


AI&ML powered ES

JetBot with
NVIDIA
Jetson Nano
(on the top)



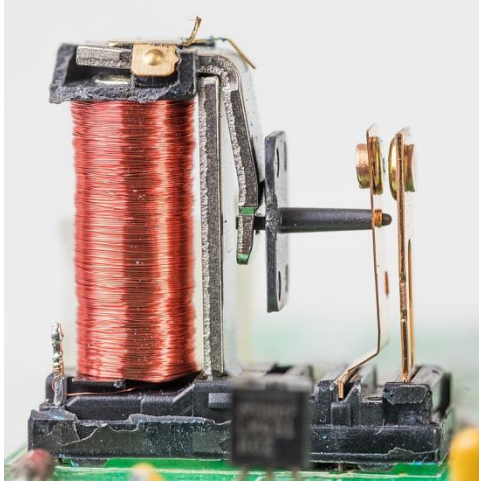
Scheme of Control System



Generations of control systems

- Relay logic systems – hard-wired
- Programmable logic controllers
- Fuzzy logic controllers
- Microcontrollers – based on a microprocessor and dedicated software
- AI & ML based controllers

Hard-wired control components



Relay



7 contact cam timer

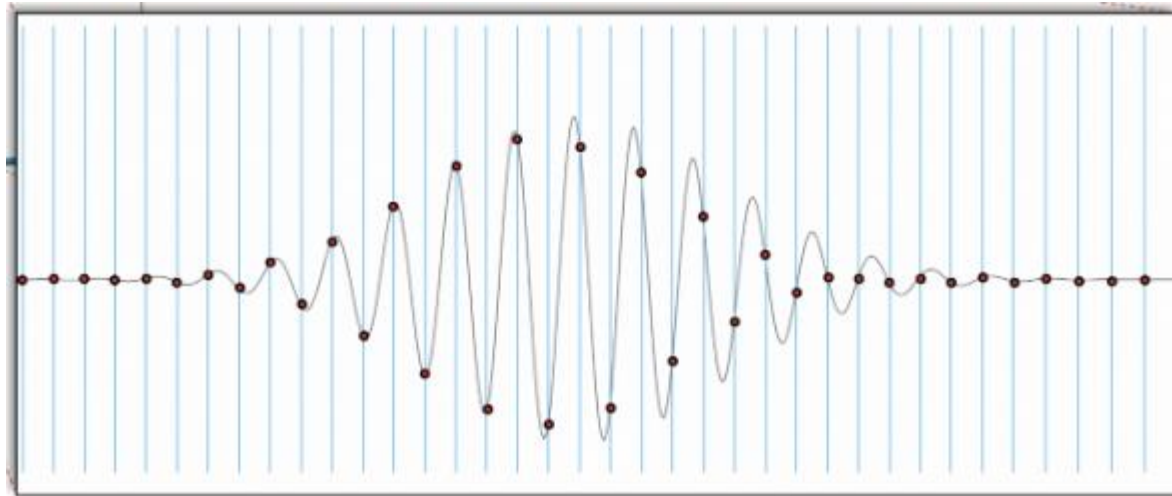


Electro mechanical
timer

Digital vs Analog

- Analog control system – transformation of analog signals according to required control scheme (ODE, PDE)
- Digital control system – digitalization (discretization) of input and output signals; processing of digital info by the controller

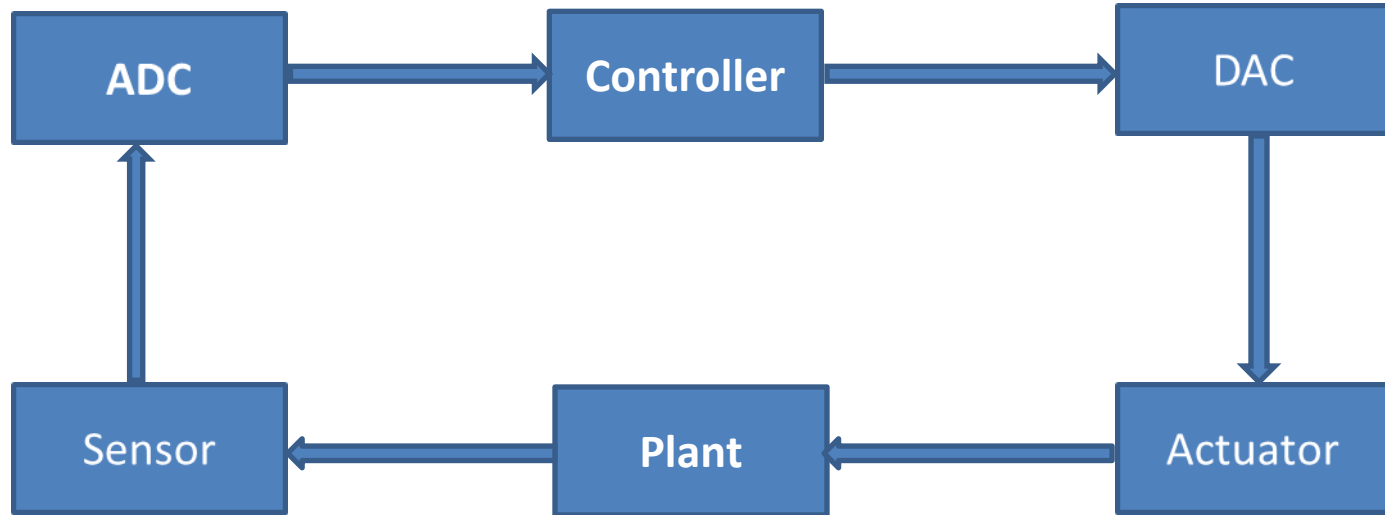
Digitizing analog signals



Digital Control

- Object of control – **plant**
- Subject of control – **controller**
- **Sensors** – to read the plant state
- **Actuators** – to correct the plant state
- **AD** – analog to digital convertor
- **DA** – digital to analog convertor
- **Electric Relays** – to control big electric devices

Scheme of Digital Control System



Types of ES

- Real-Time: required output function within a specified time interval (driving a vehicle)
- Network: connected via network (avionics)
- Mobile: connected via wireless network (IoT)
- Independent – stand-alone (microwave stove)

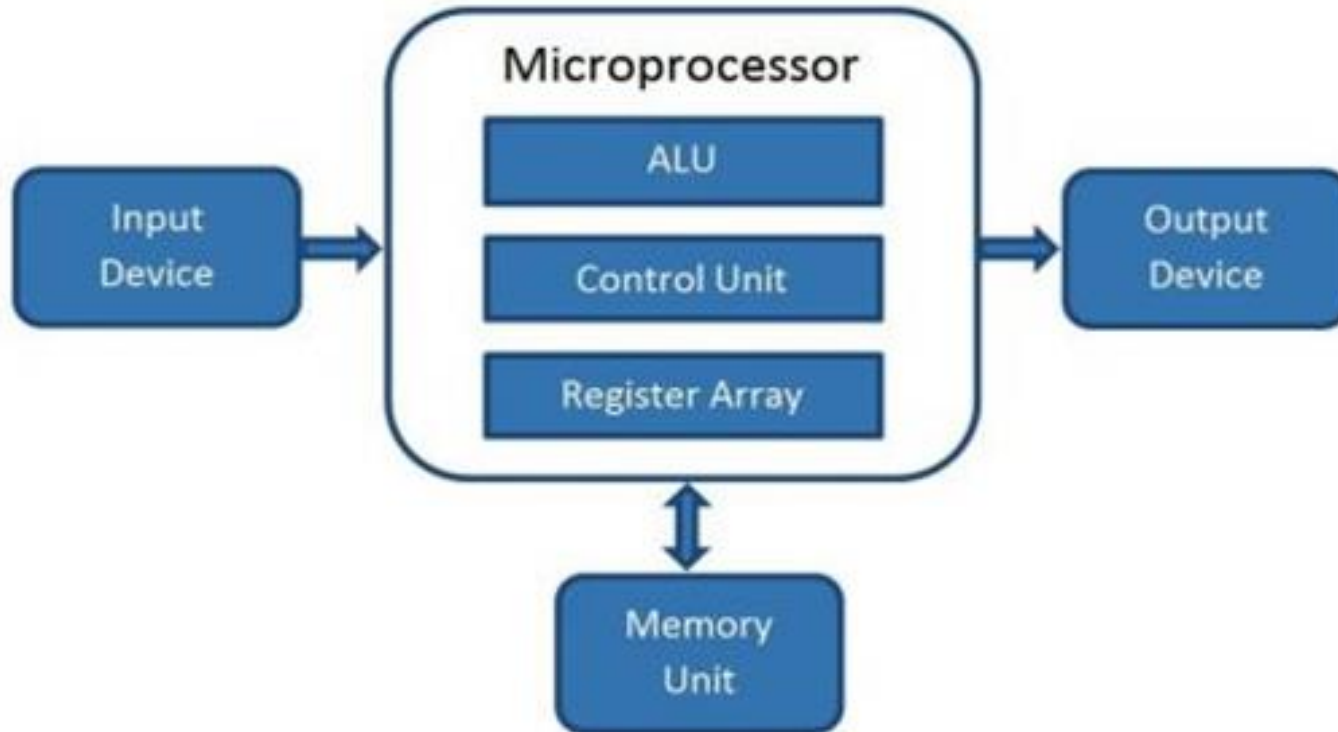
Application Domains of ES

- Automotive Vehicles
- Personal Devices, Home Appliances
- Medical Equipment
- Industrial Machines
- Payment Systems
- Security Systems

Microcontroller-based ES

- Micro-processor (ALU, registers, PC, PSW)
- ROM stores a program
- RAM stores data, including runtime context
- Controllers of peripheral devices
- Microcontroller-on-chip

Microprocessor Block Diagram



Microprocessor



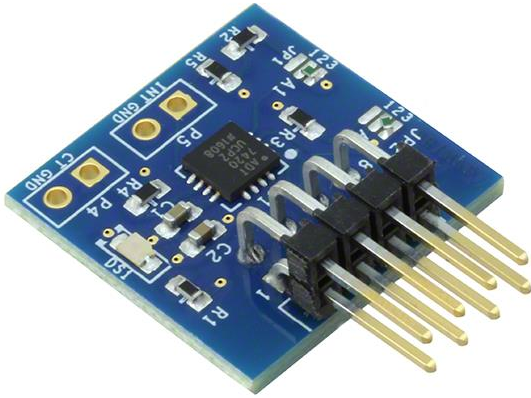
Microcontroller board



MICROCHIP EV64G19A

Development Board,
dsPIC33CK512MP608-I/PT,
Motor Control, Power
Management - Motor Control

Examples of peripherals

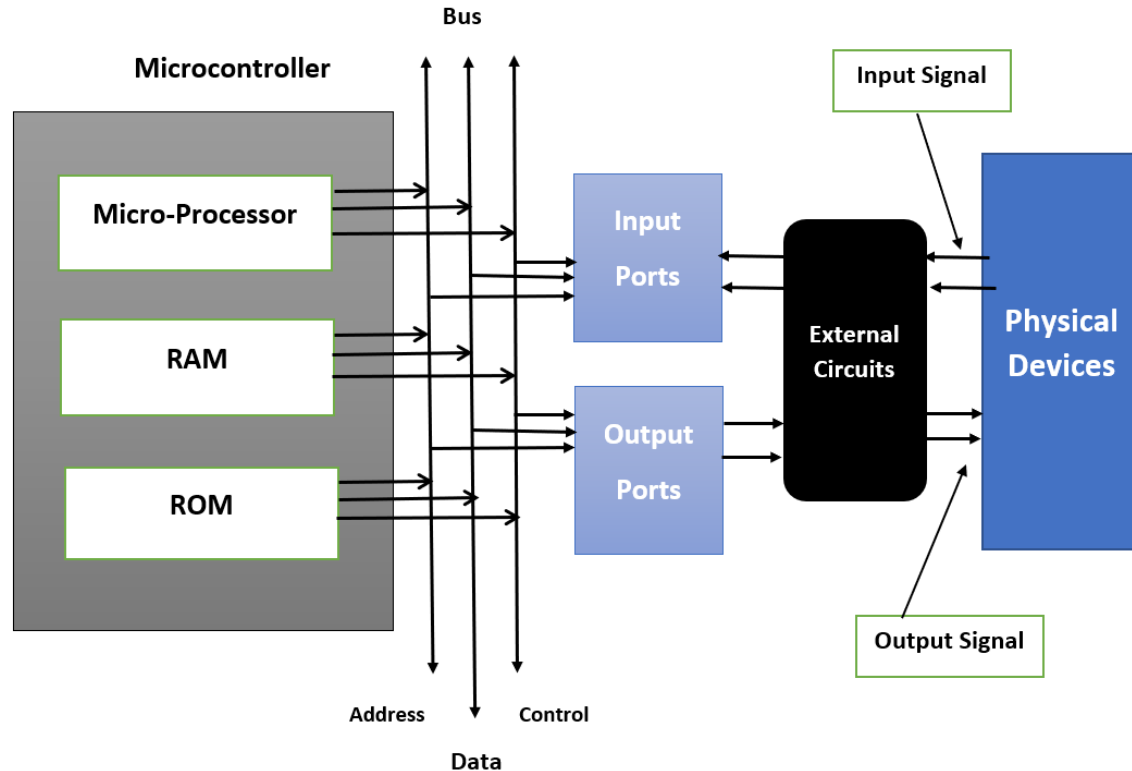


Temperature Sensor



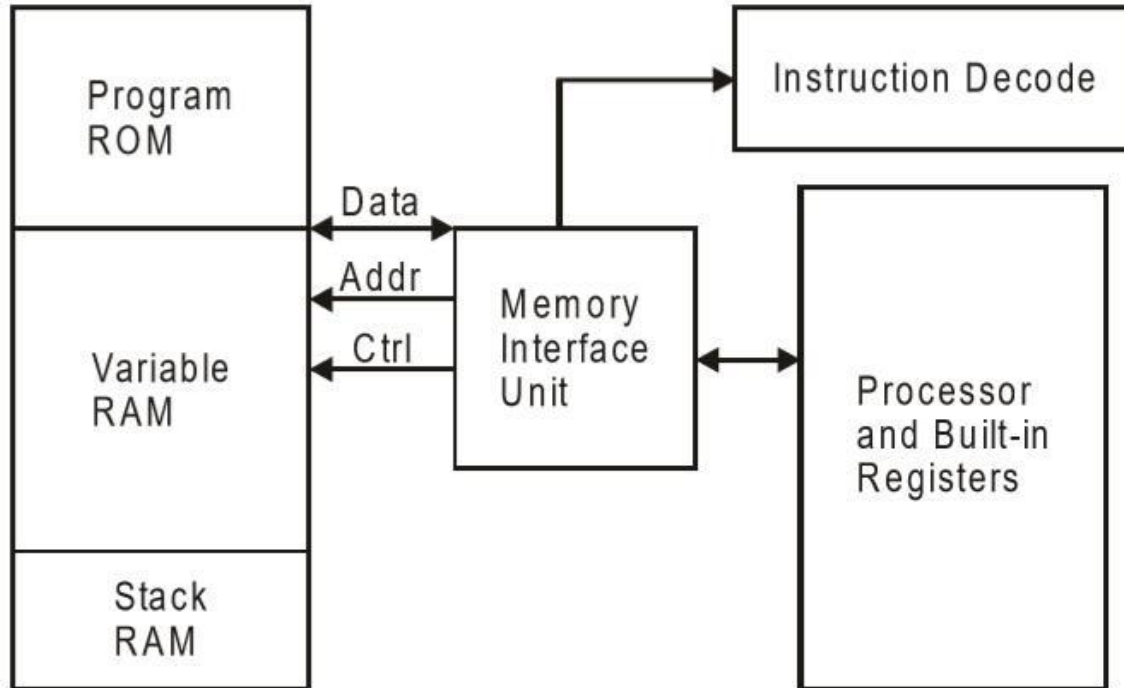
High Performance Cooling Fan, 4-Pin

Embedded system architecture

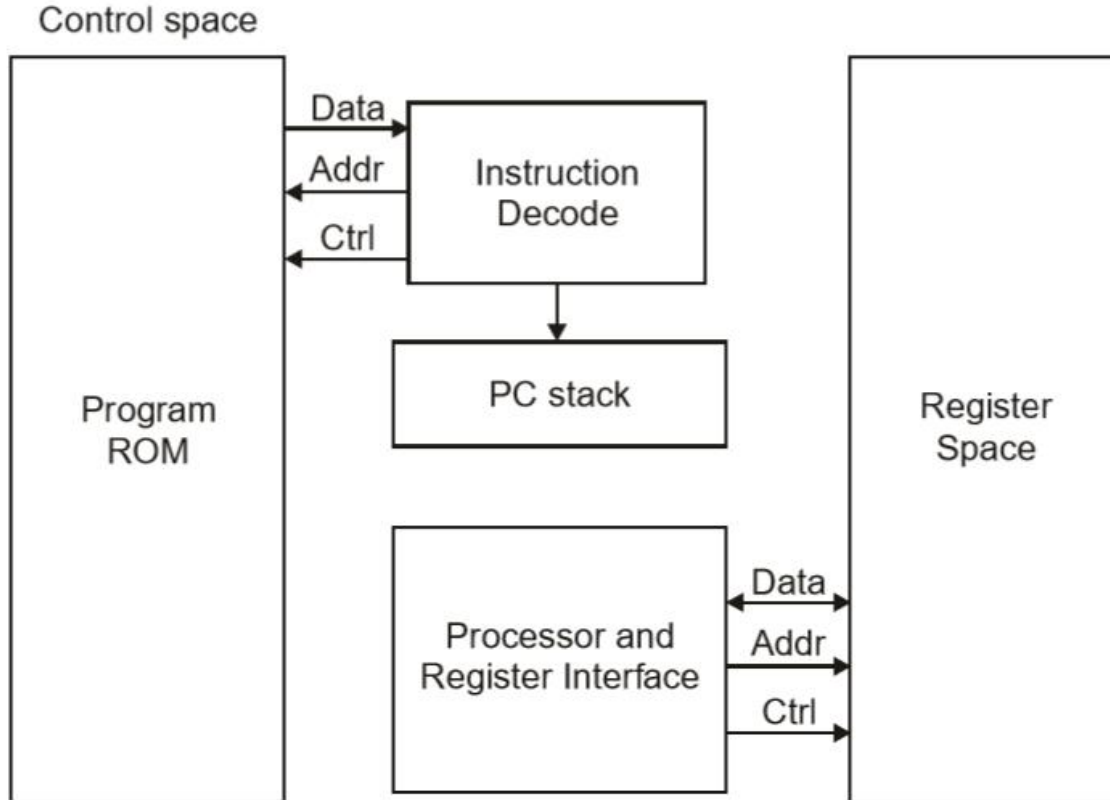


Von Neumann Architecture

Memory space



Harvard Architecture



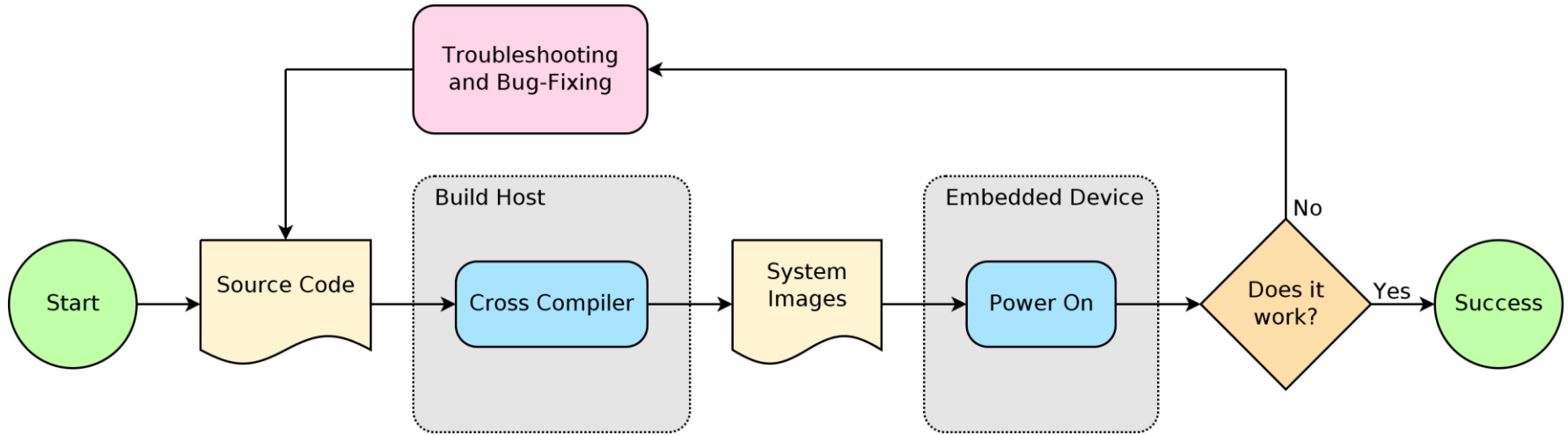
Specific features of embedded systems

- Limited resources
- Restricted operating systems or no operating system
- Low energy consumption for stand-alone systems
- Aggressive environment

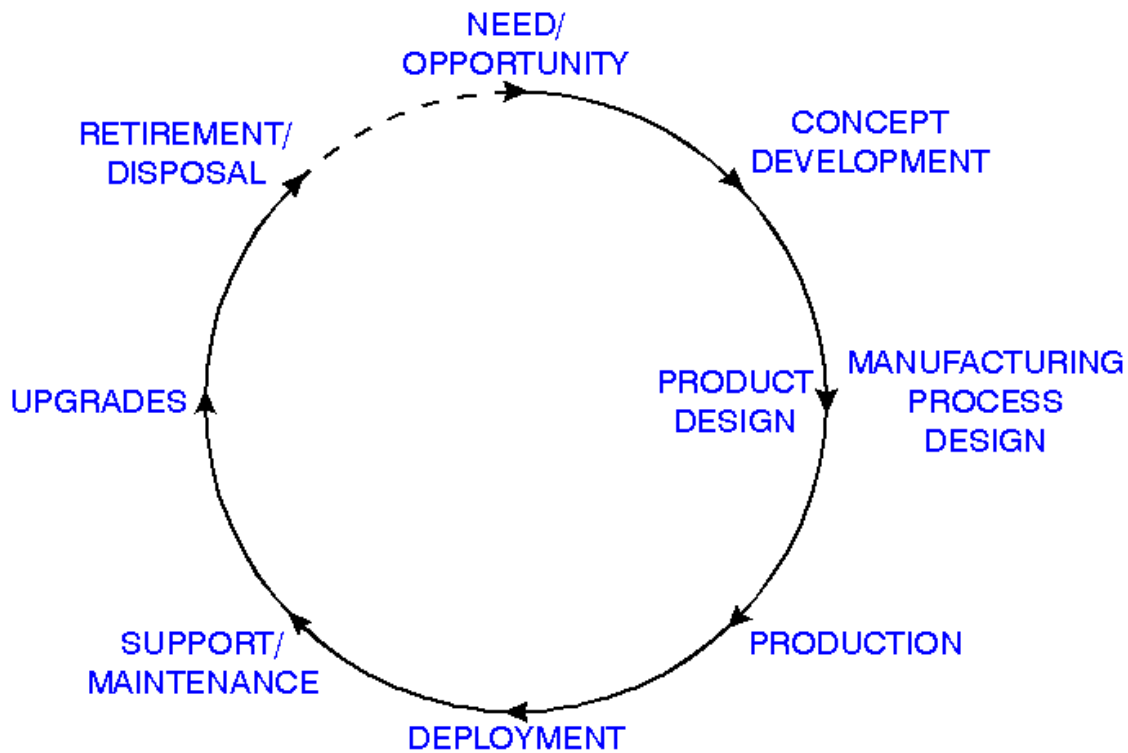
Cross-platform design

- General purpose computer tools – design and debug hardware-software within the modeled (simulated) environment
- Compose ES hardware
- Upload ES software
- Debug on plant

Cross-platform approach (simplified)



Lifecycle of ES



Reliability of ES

- Testing – run on correct combinations of (input,output) data; difficulty for processes
- Verification – formal proof of correct work – at least correspondence to specifications
- Advantages of using formal systems for ES design: logic, sequential logic, ladder diagrams, Petri nets, etc.

Real-time systems

- Time constraints (hard and soft)
- Real time clock
- The tasks assigned to real-time systems need to be completed in given time interval
- Performance evaluation techniques
- Optimization of algorithms
- Fast quasi-optimal solutions

Operational environment

- Bare Metal embedded system – application software running directly on hardware
- Real Time Operating System – PX5 RTOS, FreeRTOS, QNX, Embedded Linux, HarmonyOS, ThreadX, VxWorks
- OS brings in hardware abstraction level

Organization of control

- Basic cycle – sensors' poll, computation of optimal control, generation of output data for actuators
- Event (interrupt) driven – components of ES software responding to particular events

Model based Control

- Formal model of plant
- Formal model of control
- Formal Verification and Optimization
- Model: ODE, PDE, logic, automata, CSP, Petri net