Embedded Computing Overview

Dr. Larry Aamodt

Engr355 – Embedded Systems Design

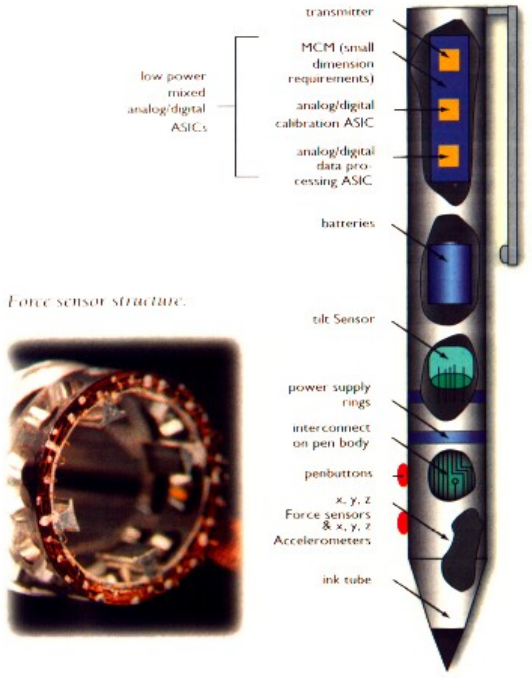
Walla Walla University

Objectives Today

- Provide a general overview of Embedded Systems;
- Show examples of Embedded Systems;
- Show current statistics in the field of Embedded Systems.

Embedded Systems Are Everywhere!

Tire Pressure Sender



SmartPen

What Is an Embedded System?

- An **embedded system** is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is *embedded* as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer, is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

  -- Wikipedia

Examples of Embedded Systems

- Air conditioner
- ATM machine
- Battery charger
- Digital camera
- DVD player
- Fax machine
- Home security system
- Mobile phone
- Modem
- Navigation system
- PDA
- Photocopier
- Printer
- Router
- Scanner
- TV
- Video game console
- Wearable computer

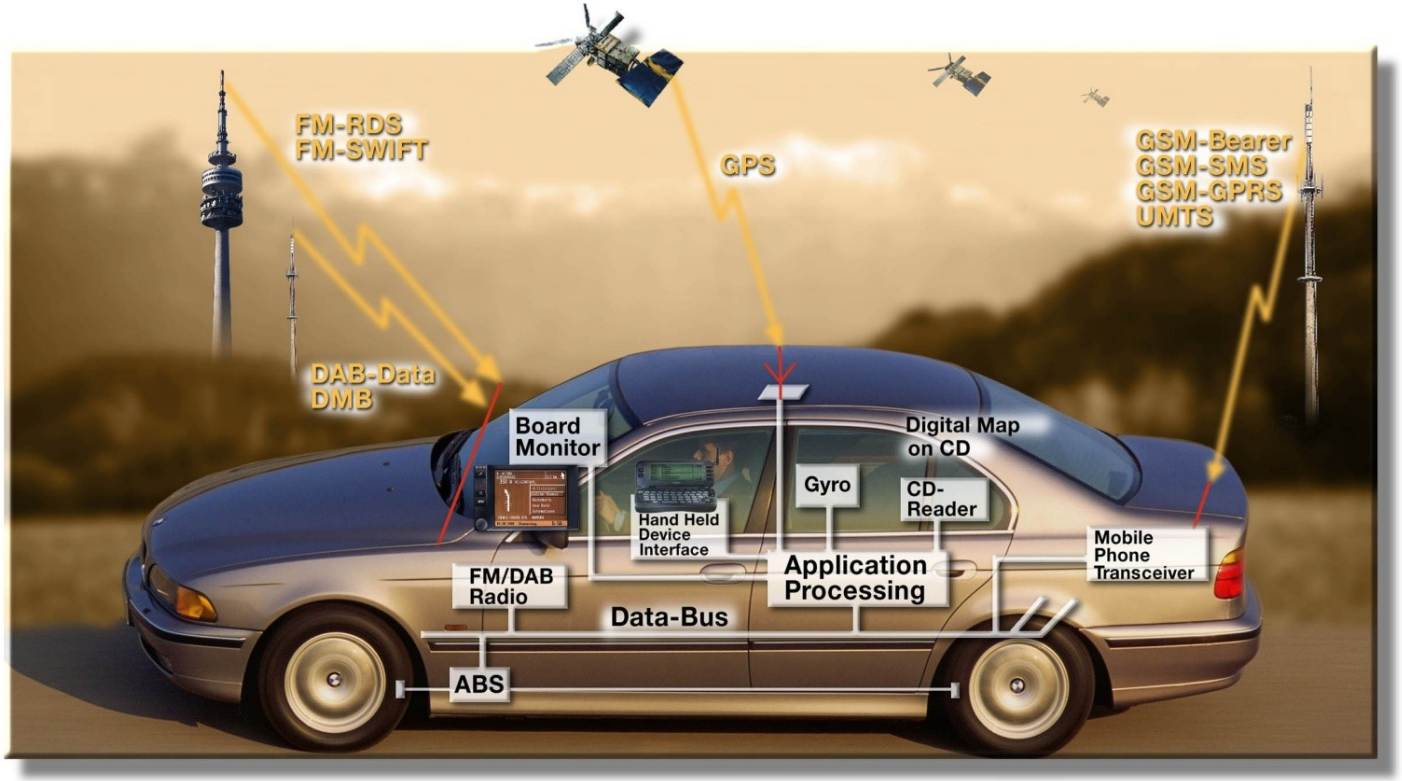
Example Embedded System: Bike Computer

- Functions
  - Speed and distance measurement
- Constraints
  - Size
  - Cost
  - Power and Energy
  - Weight
- Inputs
  - Wheel rotation indicator
  - Mode key
- Output
  - Liquid Crystal Display
- Low performance MCU
  - 8-bit, 10 MIPS

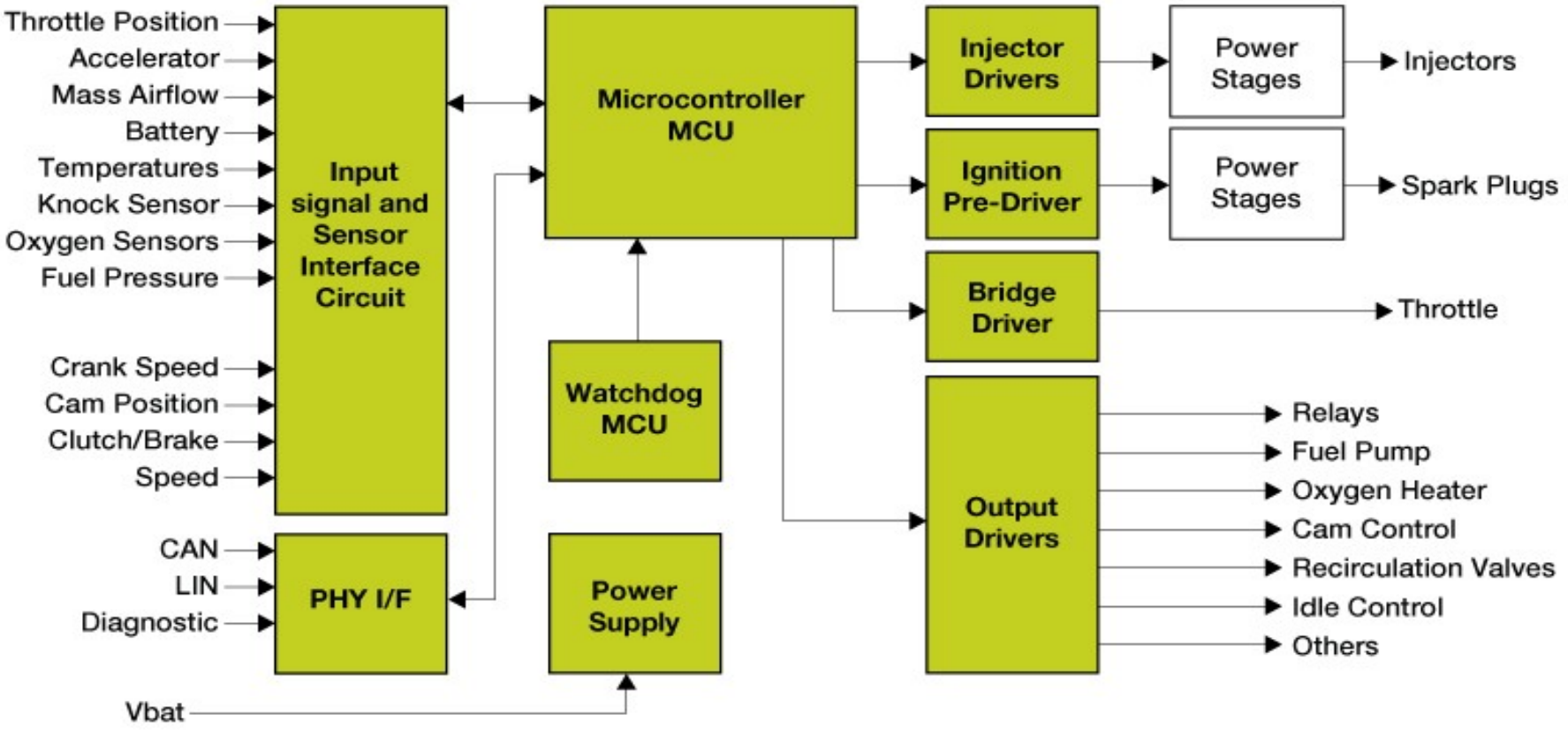
ARM

Embedded Automotive

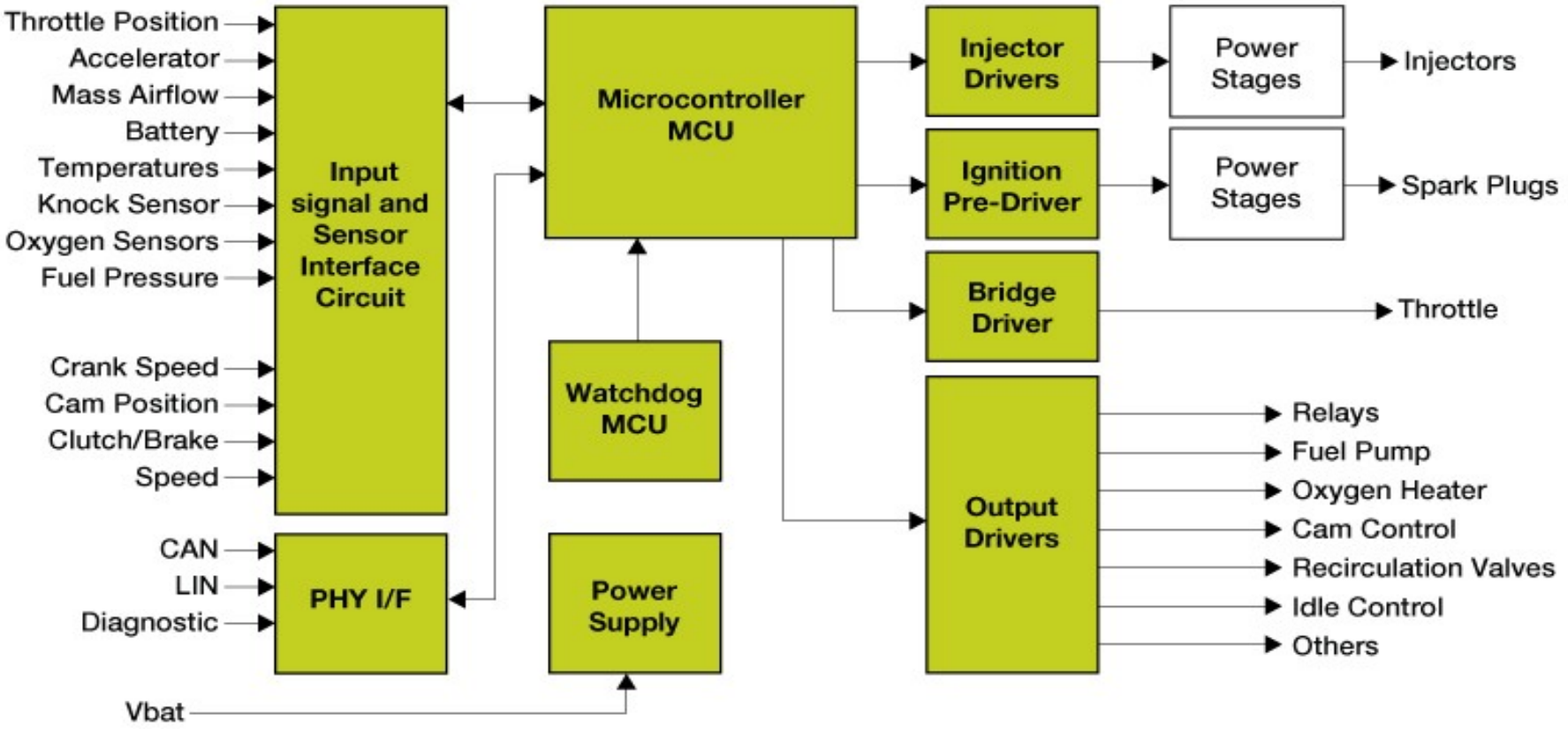
- More than 30% of the cost of a car is now in Electronics.
- 90% of all innovations in the near future will be based on electronic systems.



Slide courtesy of Alberto Sangiovanni-Vincentelli

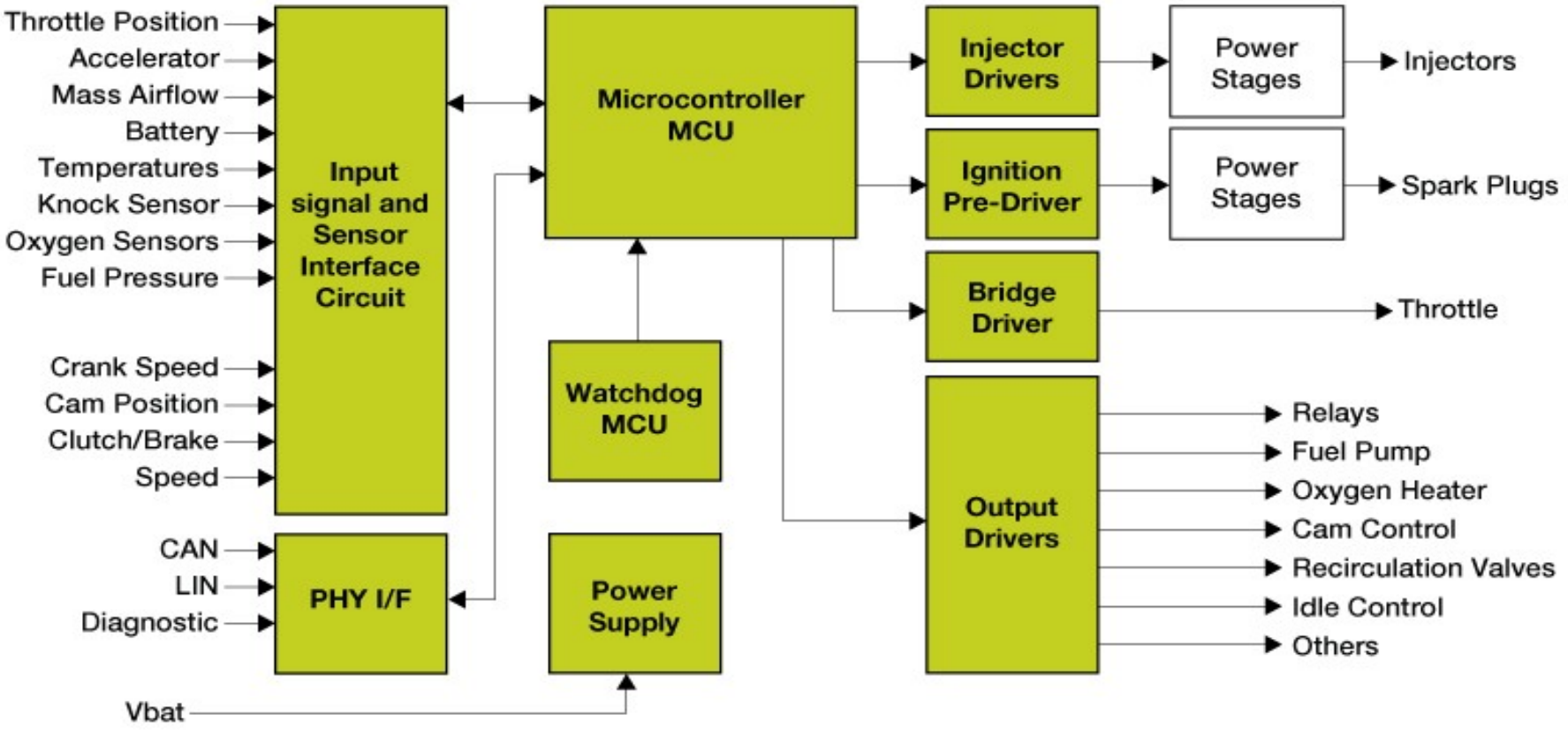


Functions: Fuel injection
Spark timing
Electronic throttle control
Air intake setting
Exhaust gas circulation
Knock control



Characteristics: Many inputs and outputs
Discrete sensors and actuators
Network interfaces to the rest of the car

High performance MCU 32bit 150-300 Mhz



Constraints: Reliability in a harsh environment
Cost
Weight

Attributes of Embedded Computing

- Real-time operation;
- Sensing & control;
- Low cost;
- Low power consumption;
- Dependable;
- Autonomous.

Characteristics (1/4)

- Special-purpose
  - Typically designed to execute a single program, repeatedly
  - It used to be single-purpose
  - Now, multi-functioned, but single-purpose
  - Low cost and efficient

Characteristics (2/4)

- Tightly constrained
  - Low cost
  - Simple systems
  - Fewer components
  - Performs functions as slow as allowable
  - Minimum power

Characteristics (3/4)

- Reactive and real-time
  - Reactive: Continually reacts to external events
  - Real-time: Must compute certain results in real-time

Characteristics (4/4)

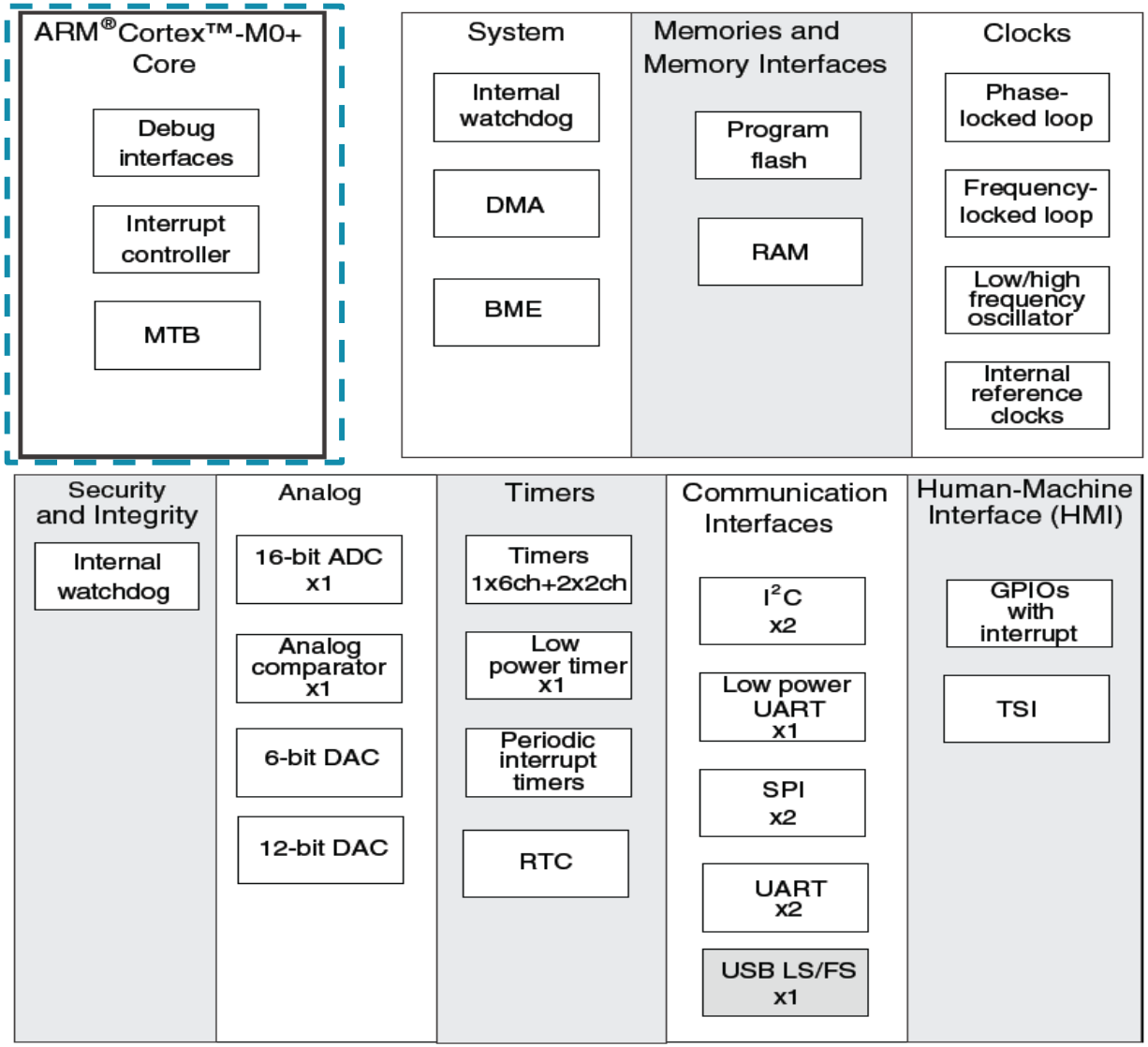
- Hardware and software co-exist
  - The software written for embedded systems is often called firmware
  - It is stored in read-only memory or Flash memory chips rather than a disk drive

Benefits of Embedded Computer Systems

- Greater performance and efficiency
  - Software makes it possible to provide **sophisticated control**
- Lower costs
  - Less expensive components can be used
  - Manufacturing costs reduced
  - Operating costs reduced
  - Maintenance costs reduced
- More features
  - Many not possible or practical with other approaches
- Better dependability
  - Adaptive system which can compensate for failures

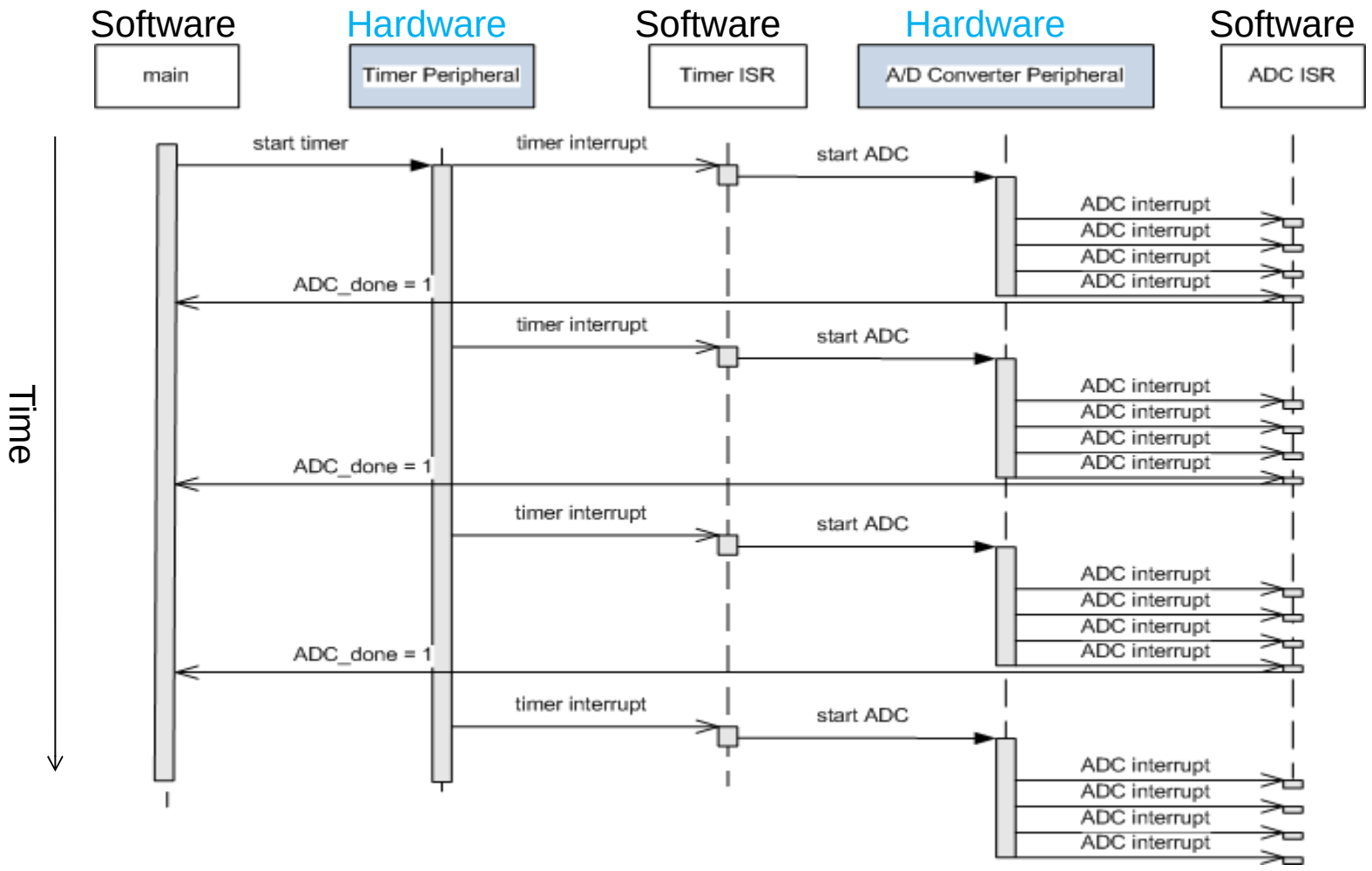
Better diagnostics to improve repair time

Microcontroller vs Microprocessor



Microcontroller (MCU)	Microprocessor (uP)
CPU – executes instructions	CPU – executes instructions
Memory - Program - Data	- no program memory - no data memory
Peripherals for input and output - Analog in/out - Logic type signals in/out (GPIO) - Timers - Clock generators - Communications between chips or subsystems I^2C Inter Chip Communications SPI Serial Peripheral Interface - USB Universal Serial Bus	- no peripherals

Concurrent Hardware and Software Operation



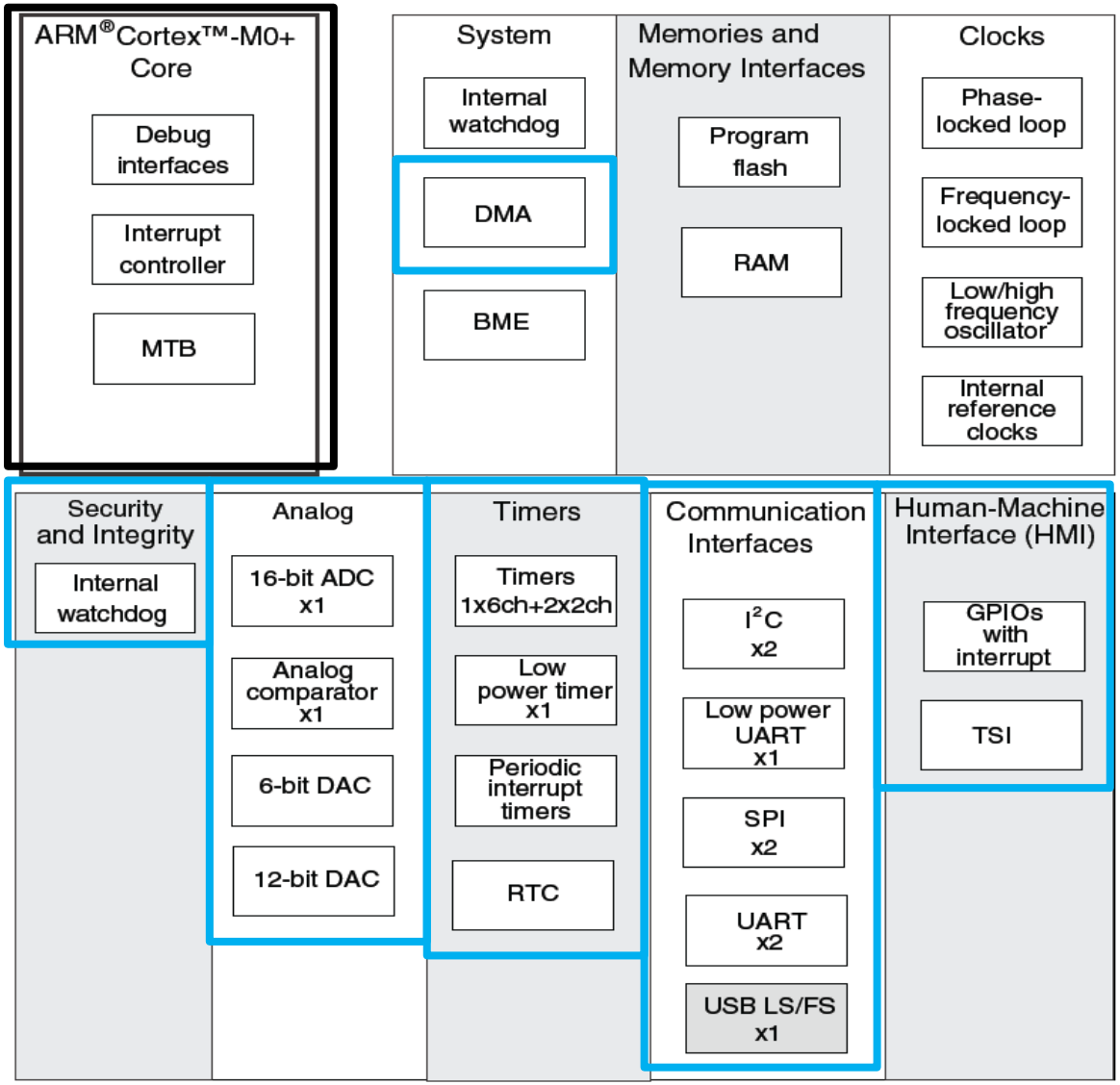
Attributes of typical embedded systems

- Concurrent, reactive behaviors
  - Must respond to sequences and combinations of events
  - Real-time systems have deadlines on responses
  - Typically must perform multiple separate activities concurrently

MCU Hardware and Software Concurrency

Specialized hardware peripherals add dedicated concurrent processing
- DMA
- Watchdog timer
- Analog interfacing
- Timers
- Communication with other devices
- Detecting external signal events

Peripherals use interrupts to notify the CPU of events



Additional attributes of embedded systems

- Fault handling
  - Many systems must operate independently for long periods of time, requiring system to handle likely faults without crashing
  - Often fault-handling code is larger and more complex than the normal-case code
- Diagnostics
  - Help service personnel determine problem quickly

Constraints

- Cost
  - Competitive markets penalize products which don’t deliver adequate value for the cost
- Size and weight limits
  - Mobile (aviation, automotive) and portable (e.g. handheld) systems
- Power and energy limits
  - Battery capacity
  - Cooling limits
- Environment
  - Temperatures may range from -40°C to 125°C, or even more

Impact of Constraints

- Microcontrollers used (rather than microprocessors)
  - Include peripherals to interface with other devices, respond efficiently
  - On-chip RAM, ROM reduce circuit board complexity and cost
- Programming language
  - Programmed in C rather than Java (smaller and faster code, so less expensive MCU)
  - Some performance-critical code may be in assembly language
- Operating system
  - Typically no OS, but instead simple scheduler (or even just interrupts + main code) (foreground/background system)
  - If OS is used, likely to be a lean RTOS

System on a Chip (SoC)

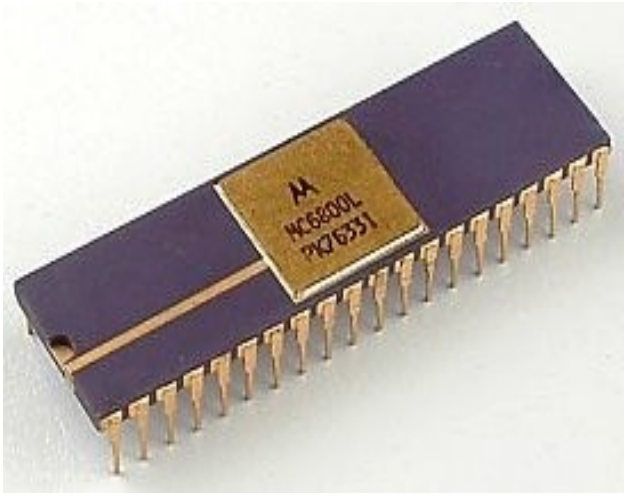
- Integrating all components of a computer or other electronic systems into a single integrated circuit (chip).
- May contain digital, analog, mixed-signal, and often radio-frequency functions – all on one chip.
- A typical application is in a cell or smart phone.

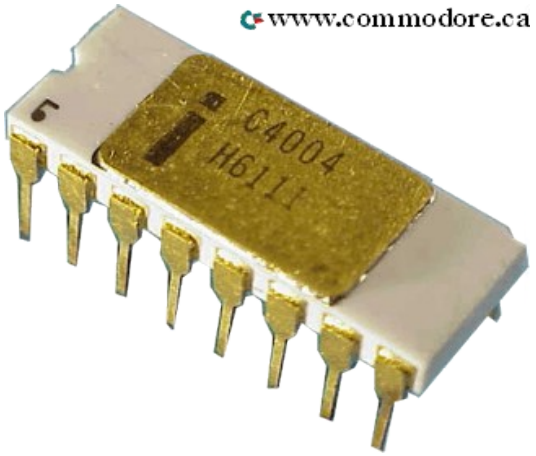
History (1/4)

- In the earliest years of computers in 1930 – 40s, computers were sometimes dedicated to a single purpose task.
- One of the first recognizably modern embedded systems was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory.

History (2/4)

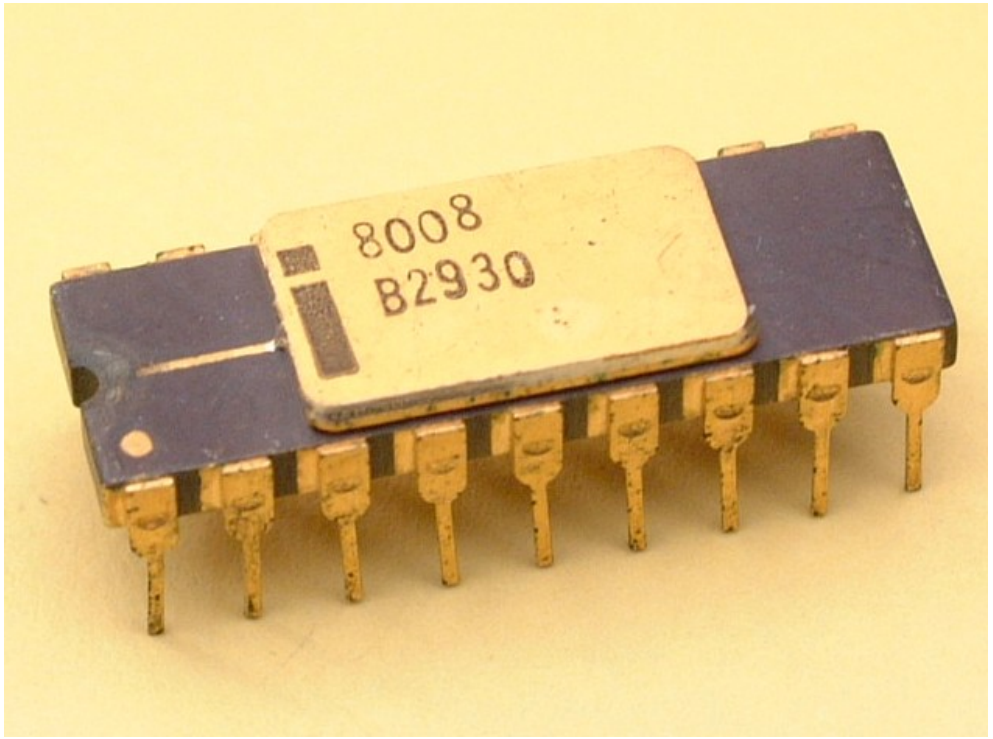
- Since early applications in the 1960s, embedded systems have come down in price and there has been a dramatic rise in processing power and functionality.
- The first microprocessor, the Intel 4004, shipped in 1971 and was designed for calculators and other small systems but still required many external memory and support chips.





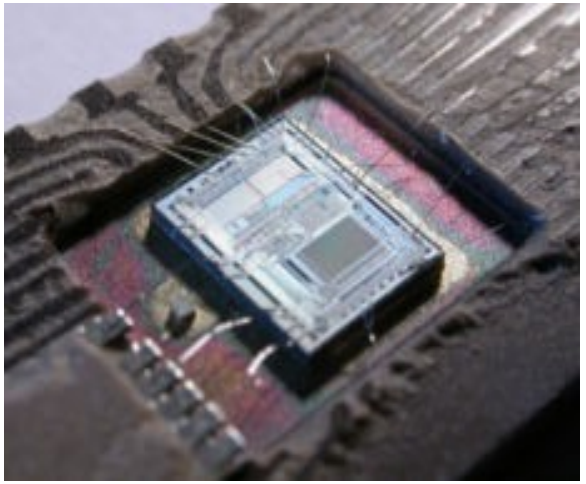
History (3/4)

- The first 8-bit microprocessor:
  Intel 8008 (1972)

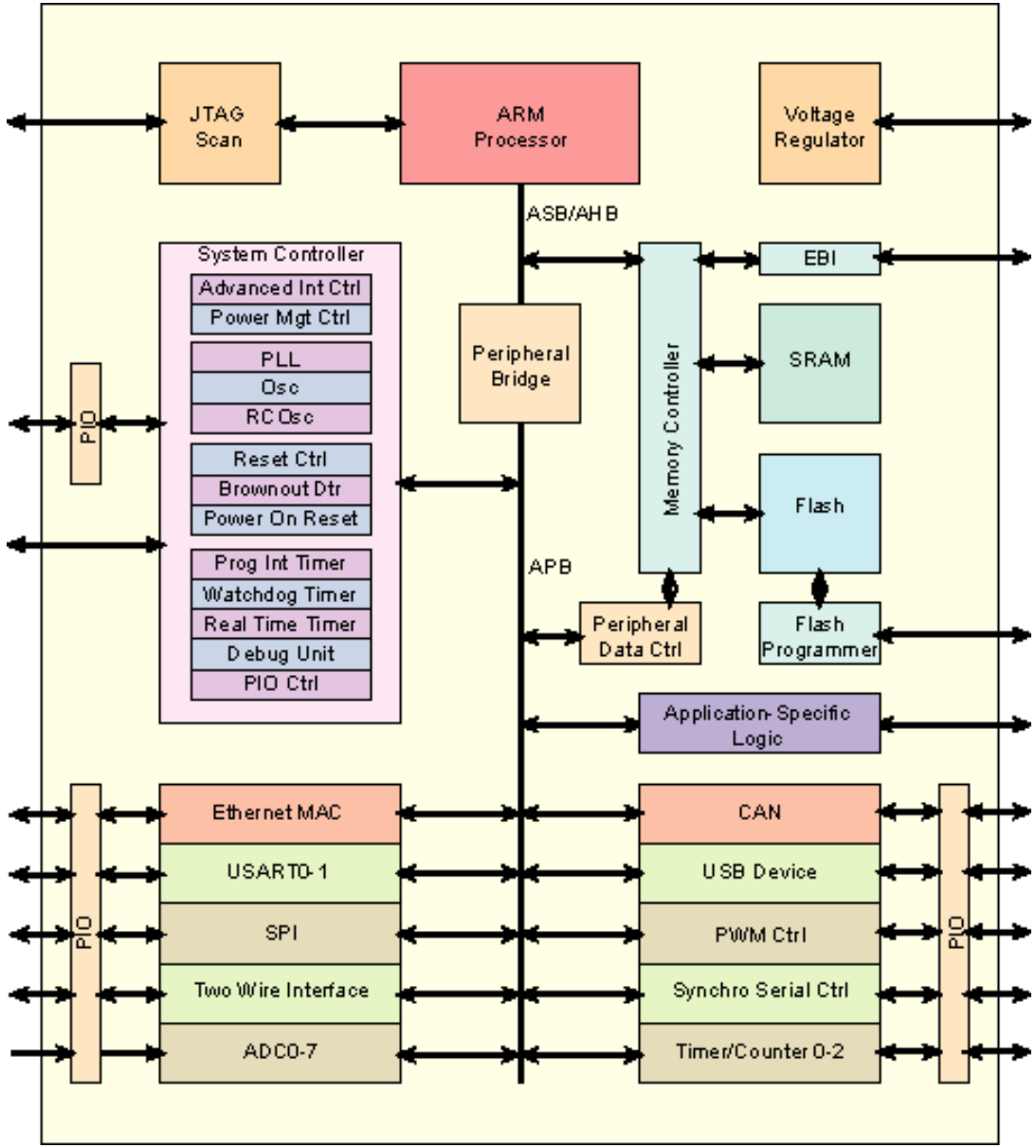


History (4/4)

- In 1977 Intel shipped the 8048 which had most of the common previously external system components integrated into the same chip as the processor to create what is called a ***microcontroller.*** This allowed even more widespread use and for small to mid size embedded applications has become the norm rather than the exception.



Typical Microcontroller Architecture



Software Development Tools

- Embedded system designers use *compilers*, *assemblers*, and *debuggers* to develop embedded system software. However, they may also use some more specific tools:
  - In circuit debuggers or emulators.
  - Utilities to add a checksum or *CRC* to a program, so the embedded system can check if the program is valid.
  - For systems using *digital signal processing*, developers may use a math workbench such as MATLAB, MathCad, or Mathematica to simulate the mathematics. They might also use libraries for both the host and target which eliminates developing DSP routines.
  - An embedded system may have its own special language or design tool, or add enhancements to an existing language such as Forth, C, or Basic.
  - Another alternative is to add a *real-time operating system* or embedded operating system.

User Interface

- Embedded systems range from no user interface at all — dedicated only to one task — to full user interfaces similar to desktop operating systems in devices such as tablets or smart phones.

Global Embedded System Market Trends

Asia Pacific is expected to witness high growth in the near future as it houses major global semiconductor and electronics companies.

ASIA PACIFIC

86.5 USD Billion 2020-e

116.2 USD Billion 2025-p

CAGR of 6.1%

The global embedded system market size is estimated to be USD 86.5 billion in 2020 and projected to reach USD 116.2 billion by 2025, at a CAGR of 6.1%.

The increasing global demand for communication equipment and consumer electronics equipped with embedded systems to drive growth of embedded systems market.

New product developments would offer lucrative opportunities for market players in the next five years.

COVID-19 is expected to adversely impact the demand for consumer electronics and automotive vehicles globally, thus, in turn, hampering the market sales for embedded system and related components.

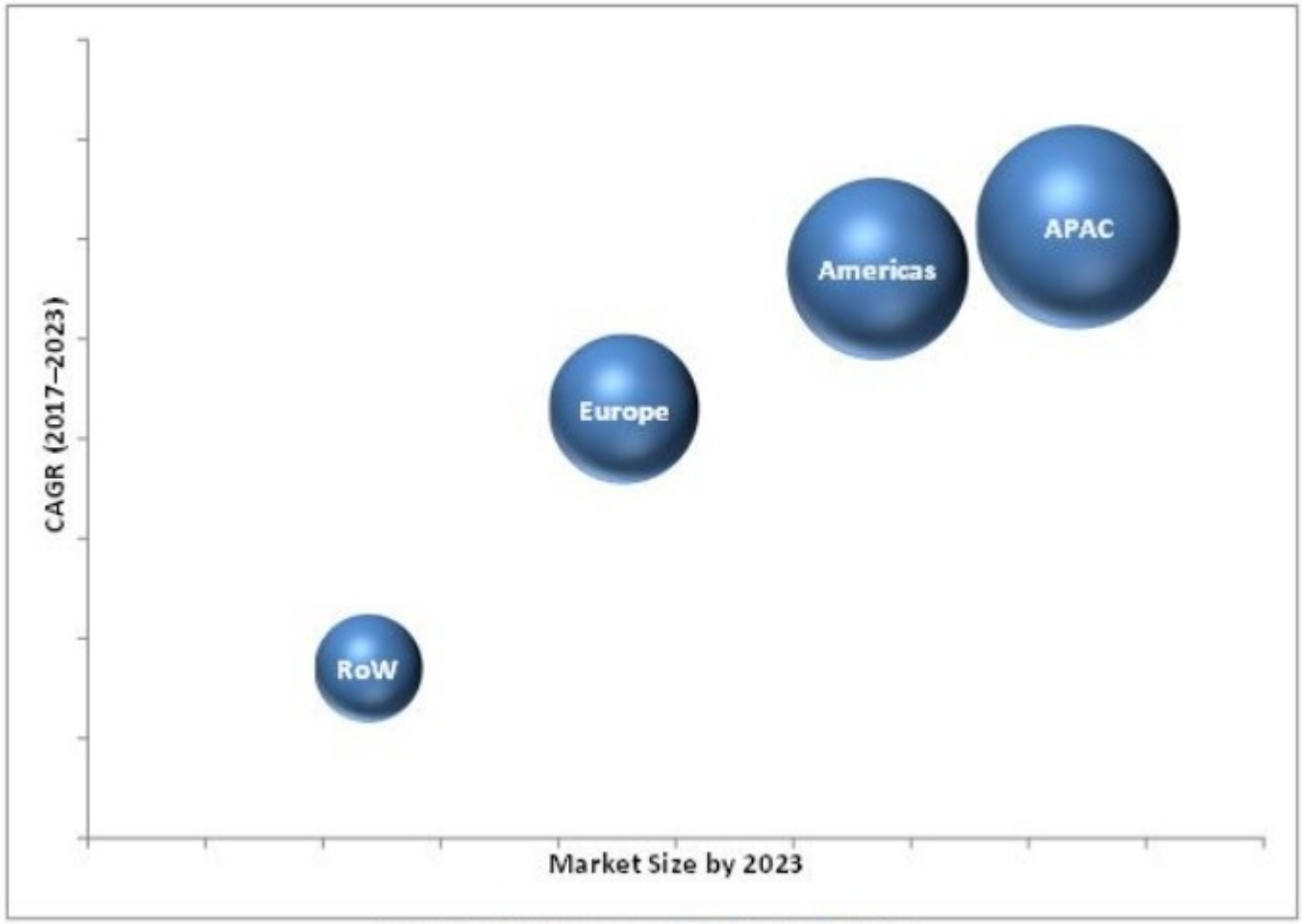
China is expected to offer lucrative growth opportunities for companies offering embedded components during the forecast period.

e-estimated, p-projected

Market drivers

- Automotive (2014, >20% of embedded market)
- Multicore processors in military applications
- Growing market for wearables
- Rising demand for embedded in health care equip
- Internet of Things (IoT)

Embedded Systems Market, by Geography, 2023 (USD Billion)



Source: MarketsandMarkets Analysis

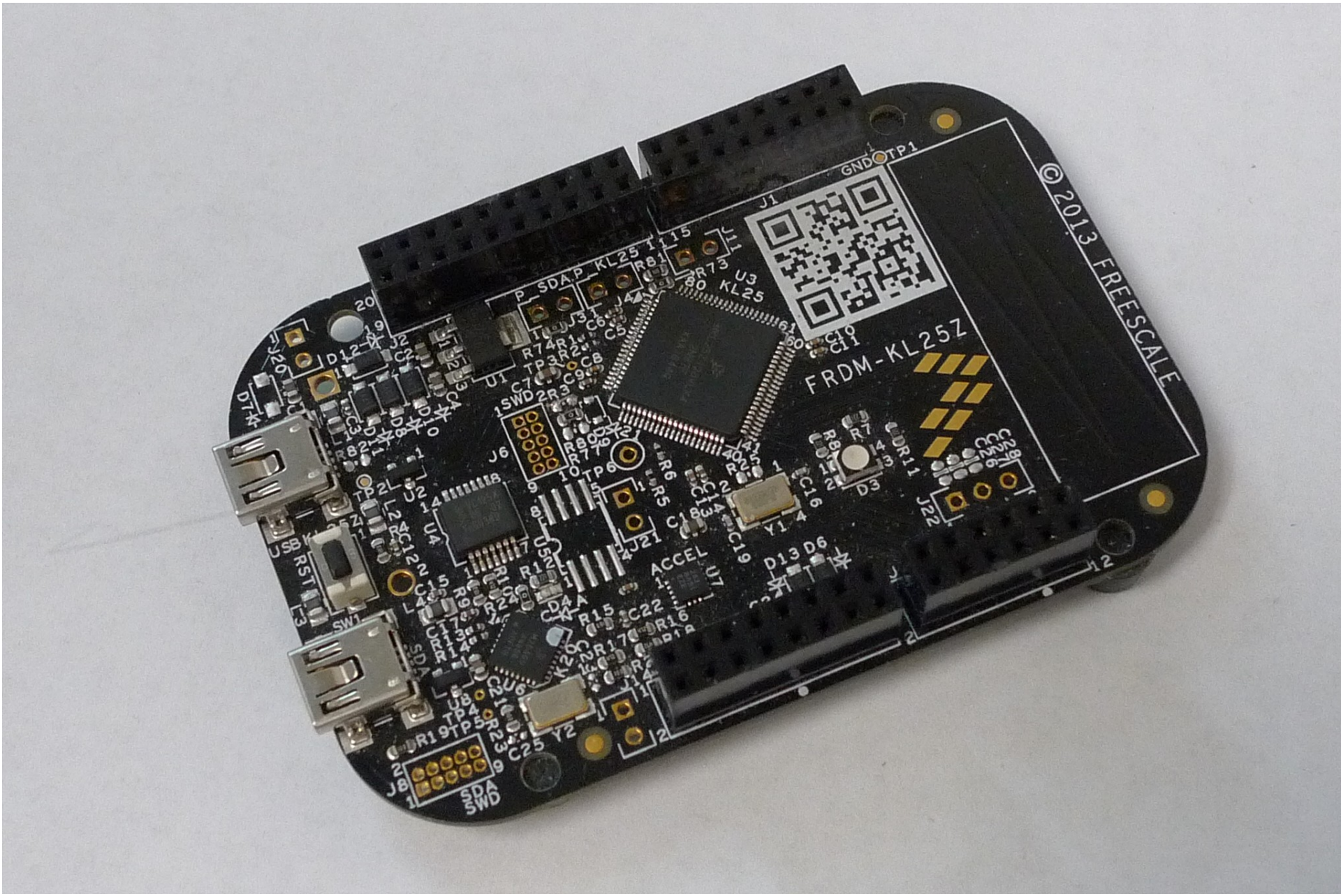
Key players in the embedded market are

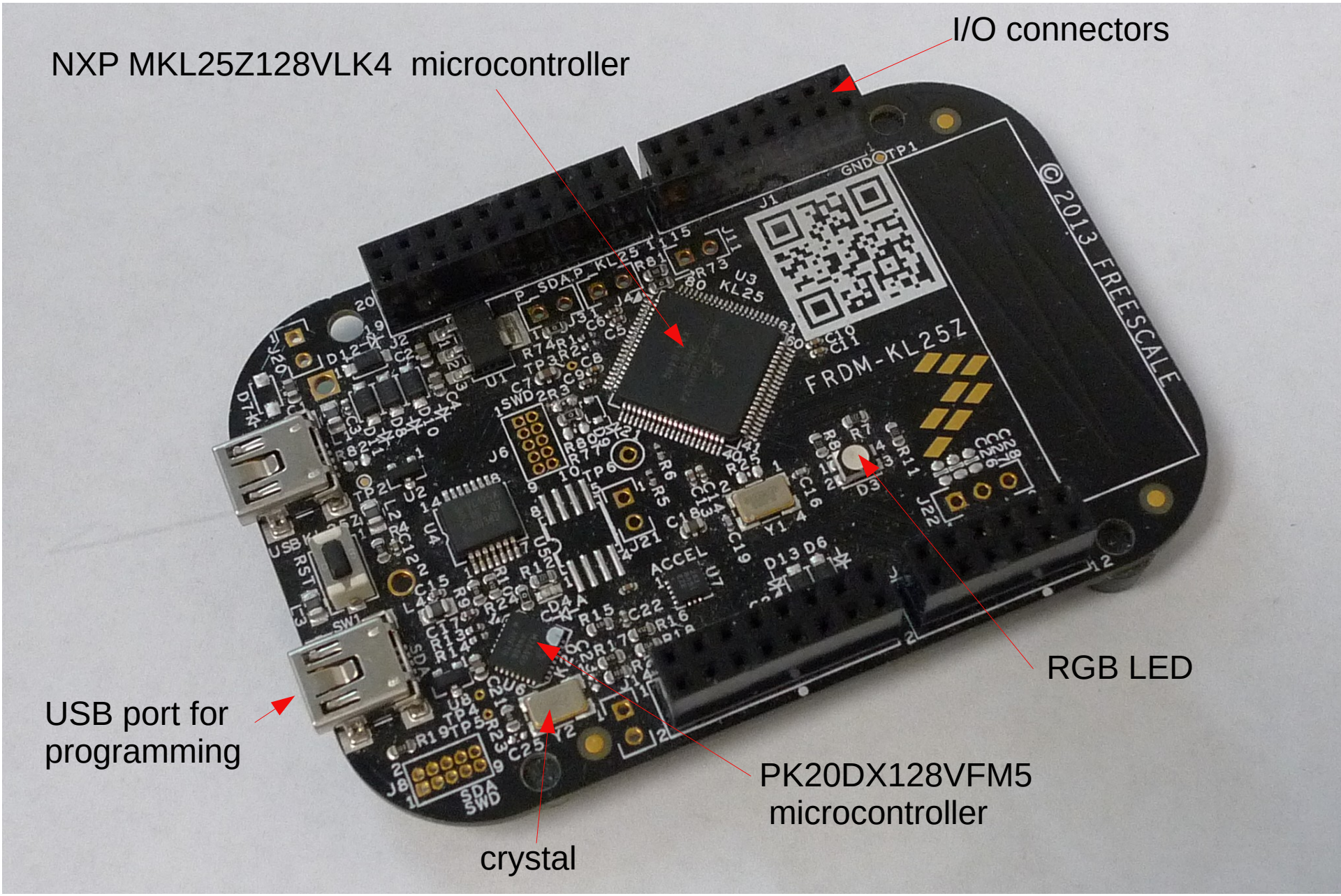
- Intel Corporation (US)
- Infineon Technologies AG (Germany)
- Renesas Electronics Corporation (Japan)
- Texas Instruments Incorporated (US)
- Microchip Technology Incorporated (US)
- STMicroelectronics N.V. (Switzerland)
- Qualcomm Technologies Incorporated (US)
- Cypress Semiconductor Corporation (US)
- Analog Devices Inc. (US)
- NXP Semiconductors N.V. (Netherlands)

Class Curriculum

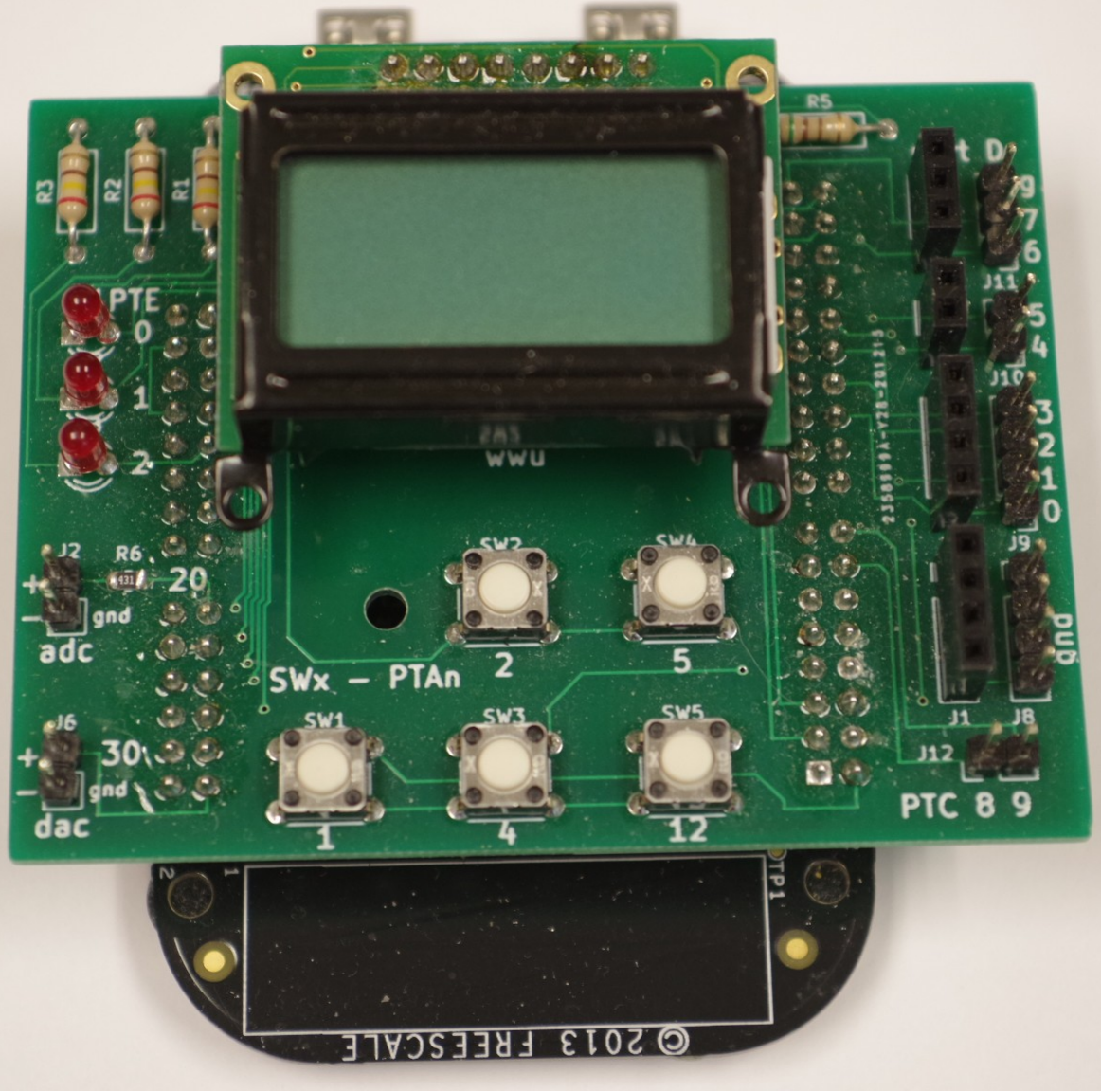
- Introductory Course: Building an Embedded System with an MCU
  - Microcontroller concepts
  - Software design basics
  - ARM Cortex M0+ architecture and interrupt system
  - C language and ARM assembly language
  - Peripherals and interfacing
  -
  - Electrical details of a circuit design
  - Construction methods: Prototyping & Circuit board creation

NXP Freedom Board FRDM-KL25Z




NXP MKL25Z128VLK4 microcontroller
I/O connectors
RGB LED
USB port for programming
PK20DX128VFM5 microcontroller
crystal
FRDM-KL25Z
© 2013 FREESCALE

WWU Input/output board that plugs onto the Freedom board



References

- Embedded Systems: Architecture, Programming, and Design by Raj Kamal
- Embedded Systems Handbook by Richard Zurawski
- Wikipedia.org

Summary

- Provided a general overview of Embedded Systems;
- Showed examples of Embedded Systems;
- Showed current statistics in the field of Embedded Systems.