RV College of Engineering®

IOT AND EMBEDDED COMPUTING (CS344AI)

UNIT-1

Introduction to Embedded Systems and Applications

Text Book: Embedded Systems – An integrated approach, Lyla B. Das, 2013, Pearson Education, ISBN- 978-81-317-8766-3

Course In charge: Dr.Mohana
Department of Computer Science & Engineering (Cyber Security)
RV College of Engineering, Bangalore-560059

Go, change the world®

.

Syllabus

Unit – I	9 Hrs
Introduction to Embedded Systems and Applications Embedded Systems: Definition, Desirable Features & General Characteristics. Embedded Systems Vs General Computing Systems, Model of an Embedded System, Classification of Embedded Systems, Examples of Embedded Systems. ARM Processor/Controllers: History of the ARM Processor, the ARM Core, features of ARM Processors, ARM Processor families - Cortex A, Cortex R and Cortex M. Interfacing and Application Development Using ARM Microcontroller:LPC 2148 ARM Microcontroller-Features of the LPC 214X Family,Internal Block Diagram of LPC 2148. Block Diagram of MCB 2140 compatible board / RV-ARM-Board, Keil IDE features for embedded application development	

Introduction

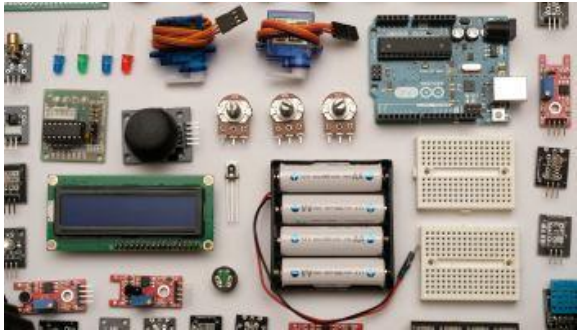
- An electronic system which is designed to perform one or a limited set of functions, using **hardware and software**.

- An Electronic/ Electromechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software)

- Quite difficult to ‘define’, because of the large variety of devices included in this class.

E.g. Electronic Toys, Mobile Handsets, Washing Machines, Air Conditioners, Automotive Control Units, automobile, Set Top Box, DVD Player etc…

- Having hardware and software makes an embedded system a computer, but this computer performs only a limited set of functions.
- Automobile industry, ECU
- intelligence to the operation or working

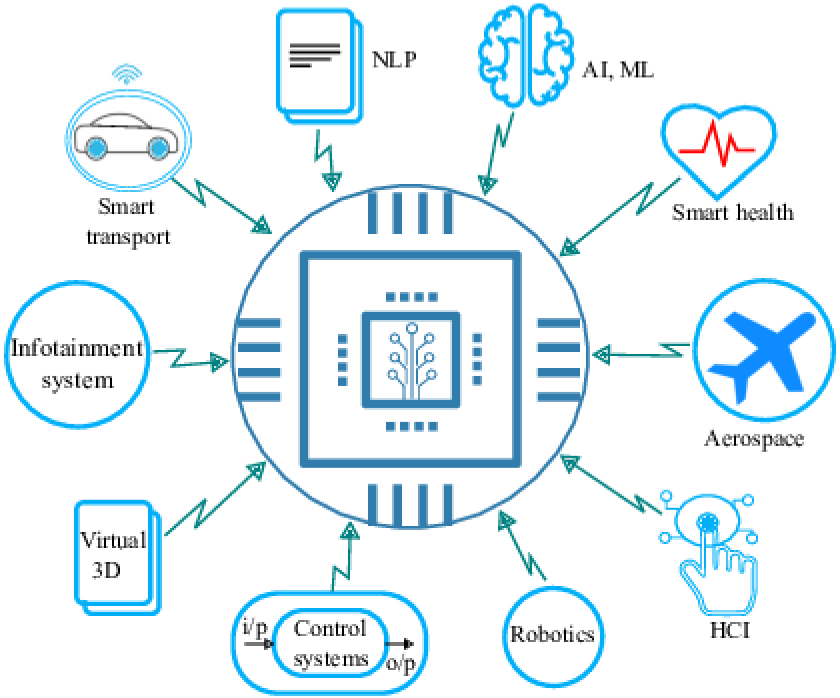


Application Domain of Embedded Systems

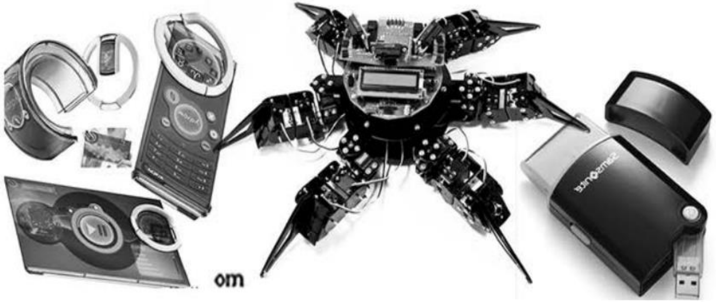
- **Consumer Electronics:** Camcorders, Cameras etc.
- **Household Appliances:** Television, DVD players, washing machine, Fridge, Microwave Oven etc.
- **Home Automation and Security Systems:** Air conditioners, sprinklers, Intruder detection alarms, Closed Circuit Television Cameras, Fire alarms etc.
- **Automotive Industry:** Anti-lock breaking systems (ABS), Engine Control, Ignition Systems, Automatic Navigation Systems etc.
- **Telecom:** Cellular Telephones, Telephone switches, Handset Multimedia Applications etc.
- **Computer Peripherals:** Printers, Scanners, Fax machines etc.

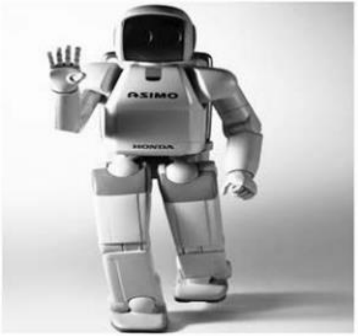
- **Computer Networking Systems:** Network Routers, Switches, Hubs, Firewalls etc.
- **Health Care:** Different Kinds of Scanners, EEG, ECG Machines etc.
- **Measurement & Instrumentation:** Digital multi meters, Digital CROs, Logic Analyzers PLC systems etc.
- **Banking & Retail:** Automatic Teller Machines (ATM) and Currency counters, Point of Sales (POS)
- **Card Readers:** Barcode, Smart Card Readers, Hand held Devices etc.

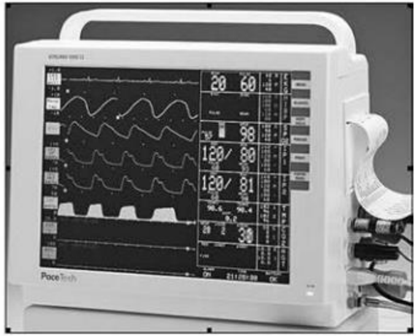
i) **Consumer electronics**: Cameras, music players, TVs, DVD players, microwave ovens, washing machines, refrigerators and remote controls.
ii) **Household appliances/home security systems**: Airconditioners, intruders and fire alarm systems.
ii) **Automobile controls**: Anti-lock braking system, engine and transmission control, door and wiper control, etc.
iv) **Handheld devices**: Mobile phones, PDAs, MP3 players, digicams, etc.
v) **Medical equipments**: Scanners, ECG and EEG units, testing and monitoring equipments.
vi) **Banking:** ATMs, currency counters, etc.
vii) **Computer peripherals**: Printers, scanners, webcams, etc.
viii) **Networking:** Routers, switches, hubs, etc.
ix) **Factories**: Control, automation, instrumentation and alarm systems.
xi) **Aviation**: Airplane controls, guidance and instrumentation systems.
xii) **Military:** Control and monitoring of military equipments.
xiii) **Robotics:** Used in factories, household and hobby-related activities.
xiii) **Toys**.


NLP
AI, ML
Smart transport
Smart health
Infotainment system
Aerospace
Virtual 3D
HCI
i/p
Control systems
o/p
Robotics

Sample Embedded products







Desirable Features and General Characteristics of Embedded Systems

- one or a small set of functions
- Low-power dissipation
- Limited memory and limited number of peripherals
- Highly reliable.
- Operation with time constraints.

PC vs Embedded System

Category	Computer	Embedded device
Description	A computer is a combination of hardware and software resources that integrate and provide **various functionalities** to the user.	An embedded device is part of an integrated system that is formed as a **combination of computer hardware and software for a specific function** and which can operate without human interaction.
Human Interaction	A computer **needs** Human Interaction to perform tasks.	Embedded devices **do not need** Human Interaction to perform tasks.
Types based on architecture	Analog computer, Digital computer, Hybrid computer, Harvard architecture, Von Neumann architecture, Reduced instruction set computer	Small Scale Embedded Systems, Medium Scale Embedded Systems, Sophisticated or Complex Embedded Systems
Parts	It has 2 parts: Hardware and Software.	It has 3 parts: Hardware, **Firmware**, and Software.
Tasks	It can perform many tasks.	It performs **limited** tasks.
Cost to user	The user has to pay more for a computer.	The user incurs a **lesser cost** for an embedded system.
Peripherals	Computers have peripherals such as keyboards and mice, displays, printers, Hard disk drives, floppy disk drives, optical disc drives, etc.	Embedded Devices have peripherals such as Serial Communication Interfaces (SCI), Synchronous Serial Communication Interfaces, Universal Serial Bus (USB), Multi Media Cards (SD cards, Compact Flash), etc.
Purpose	Computers can be **reprogrammed** for a new purpose.	Embedded Devices are **made only for a specific set** of purposes.

Power Consumption	The computer needs **more operational power** than Embedded Devices.	An embedded Device needs **lesser operational power** than a Computer.
Complexity	Computers are **more complex** devices than Embedded Devices.	Embedded Devices are **less complex** devices than Computers.
Need of another device	Computers may be installed in other devices but are self-sufficient to exist.	Embedded Devices only **exist inside other Systems.**
Usage Difficulty	Computers are more Difficult when used, compared to an Embedded System.	Embedded are **easier to use** than Computers.
User Interfaces	It requires more user interface than Embedded Devices.	It requires less to no user interface than Computers.
Time Specificity	Computers are **not time-specific**. They may need to perform tasks that are not time-bound and take days to perform as well.	Embedded Devices are **time-specific**. The tasks assigned to them need to be performed within a specific time frame.

Size	Computers are usually bigger with larger hardware and input-output devices attached to them.	Embedded Devices are **smaller in size** than Computers, with limited hardware.
Developed in	1833 A.D.	1965 A.D.
Developer	Charles Babbage	Charles Stark Draper
Memory Requirement	Computers have **larger memory requirements** due to a lot of storage of data.	Embedded Devices need **less Memory**.

PC

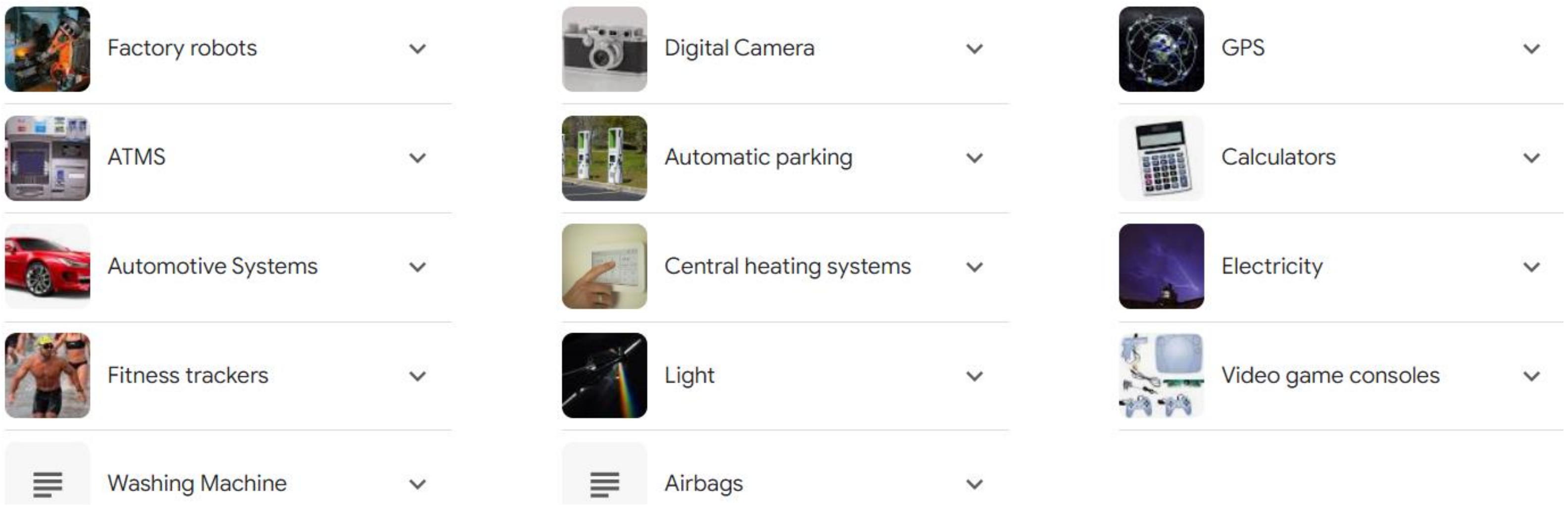
- PC is **not considered to be an embedded system**.

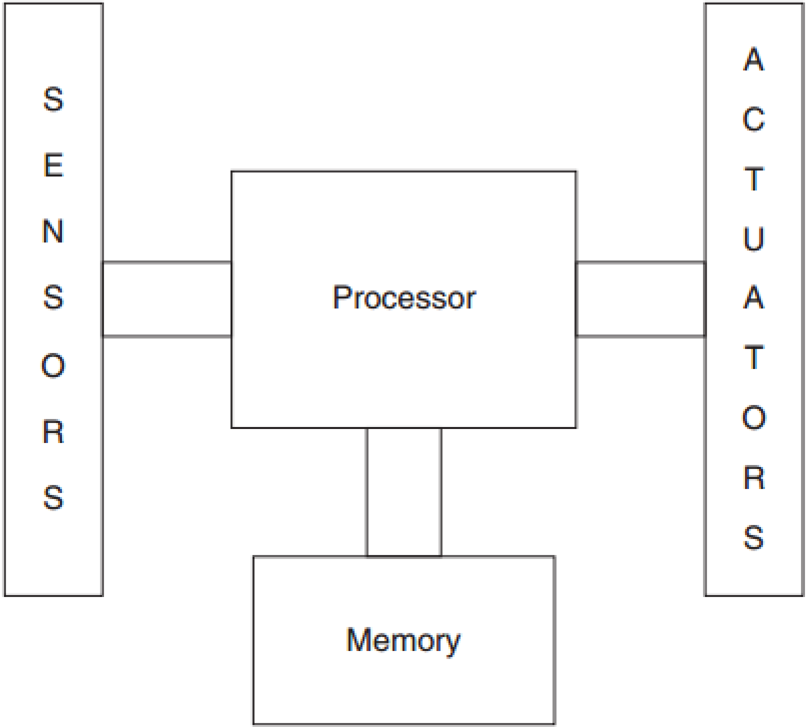
i) The PC has a large application set, from word processing and computation to communications, printing, scanning and many more.
ii) Low power consideration is a good idea, but that is not the guiding principle in its design.
iii) Memory is available in various forms: RAM, ROM and secondary memory devices like the hard disk, CDROMs and the like. More memory can be added if the user desires.
iv) Since the PC is used for various applications, more applications can be added as and when needed.
v) The PC can be accessed by input devices like the keyboard, mouse, modem, etc.
vi) Like any other system, the PC also needs to be reliable, but since it is unlikely to be the part of a very critical system, it can afford to fail once in a while (not a very good idea, though because PCs are used in critical monitoring applications sometimes).
vii) The applications on the PC need to be fast for better performance, but usually there is no time criticality involved.

Model of an Embedded System

- Any application should be able to provide solution to a real-world problem

Ex.





General model of an embedded system

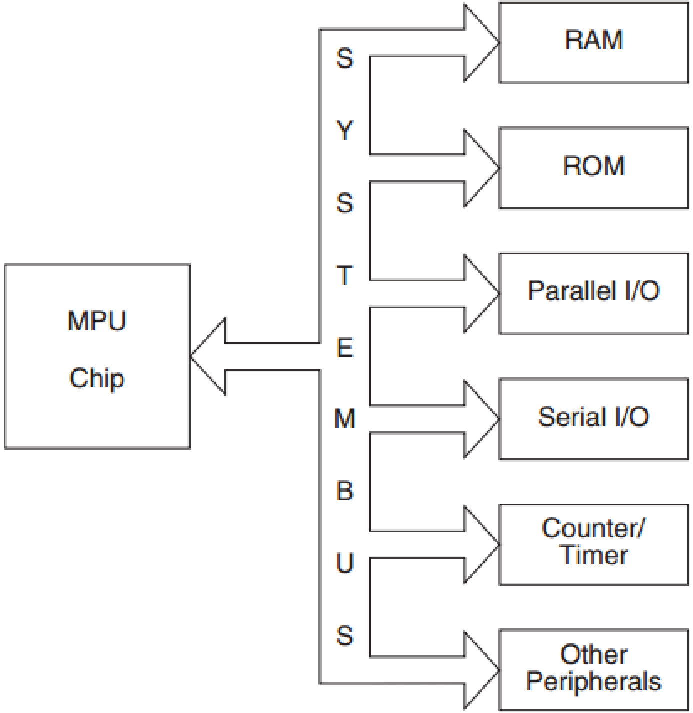
Microcontroller (µC) and Microprocessor (µP)

- Microcontroller (µC) contains not only a processing unit but a small amount of memory (ROM, RAM, etc.), a few IO ports for peripherals, timer, etc.
- Microprocessor (µP) contains only a processing unit which is quite powerful in terms of computing.

Feature	Microcontroller (µC)	Microprocessor (µP)
Purpose	Designed for specific embedded system applications	Designed for general-purpose computing applications
Architecture	Single-chip computer system with on-board memory, peripherals, and I/O interfaces	CPU with minimal on-board memory, peripherals, and I/O interfaces
Integration level	Highly integrated	Less integrated
System architecture	Single-chip system	CPU + support chips
Processing power	Lower power	Higher power
Instruction set	Fixed instruction set	More flexible
On-board memory	On-chip memory	No on-board memory
Input/Output (I/O)	More I/O ports	Fewer I/O ports

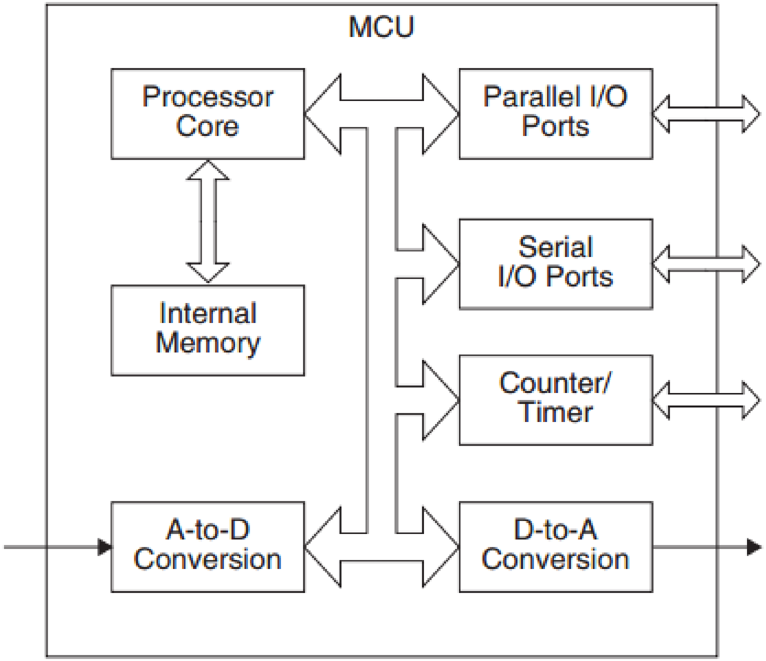
Peripheral devices	On-board peripherals	External peripherals
Cost	Lower cost	Higher cost
Power consumption	Lower power	Higher power
Applications	Embedded systems	General-purpose
Development	Integrated development environment (IDE) provided by manufacturers, with specialized programming languages and tools	Standard development tools and languages such as C, C++, and assembly
Clock speed	Lower clock speed, typically less than 100 MHz	Higher clock speed, typically greater than 1 GHz

Microprocessor Unit (MPU)



An MPU with peripherals and memory external to the chip

Microcontroller Unit (MCU)



An MCU with peripherals and memory inside the chip

Example of a Simple Embedded System

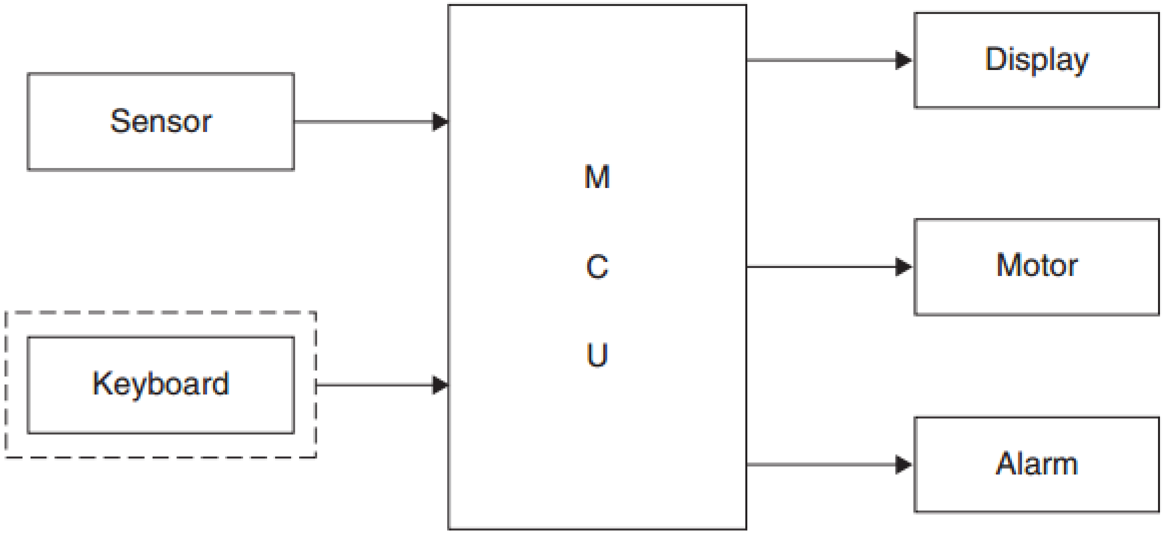


Figure 1.6 | A simple temperature monitor

Example of a Simple Embedded System

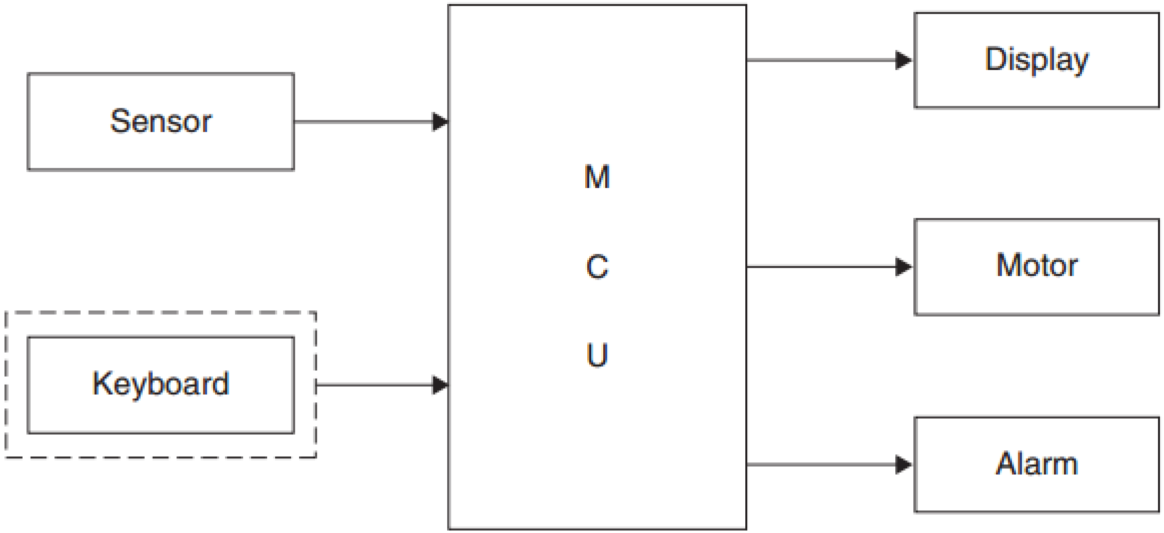


Figure 1.6 | A simple temperature monitor

Figures of Merit for an Embedded System

- Low-power dissipation
- Small physical size
- Small code size
- High speed of response

Classification of MCUs: 4/8/16/32 Bits

- 4-bit MCUs
- 8-bit MCUs
- 16-bit MCUs
- 32-bit MCUs

- **Small Scale:** The embedded systems built around low performance and low cost 8 or 16 bit microprocessors / microcontrollers.
- It is suitable for simple applications and where performance is not time critical.
- It may or may not contain OS.
- **Medium Scale:** Embedded Systems built around medium performance, low cost 16 or 32 bit microprocessors / microcontrollers or DSPs.
- These are slightly complex in hardware and firmware.
- It may contain GPOS/RTOS.

- **Large Scale/Complex:** Embedded Systems built around high performance 32 or 64 bit RISC processors/controllers, RSoC or multi-core processors and PLD. It requires complex hardware and software.
- These system may contain multiple processors/controllers and co-units/hardware accelerators for offloading the processing requirements from the main processor.
- It contains RTOS for scheduling, prioritization and management.

ASIC: Application Specific Integrated Circuit

- ASIP stands for ‘Application Specific Instruction Set Processor’.
- It is a processor whose instructions set is tailor-made for a specific application, like graphics, for example.
- it will be a sort of tradeoff design between the programmability features of a CPU and the performance of an ASIC.

Feature	FPGA	ASIC
Flexibility	High (reprogrammable)	Low (not reprogrammable)
Performance	Lower than ASIC	Higher Performance for specific tasks
Power Consumption	Higher compared to ASIC	Lower (optimized for efficiency)
Development Cost	Low (no NRE cost)	High (high NRE cost)
Production Cost per Unit	Higher compared to ASIC	Lower (optimized for efficiency)
Time to Market	Shorter (reprogrammable, adaptable)	Longer (due to design and fabrication)
Reprogrammability	Yes (can change algorithms post-production)	No (fixed design)
Suitable Production Cycle	Small to medium scale	High volume (to offset NRE costs)
Design Cycle	Shorter	Longer

History of Embedded Systems

First Recognized Modern Embedded System: Apollo Guidance Computer (AGC) developed by Charles Stark Draper at the MIT Instrumentation Laboratory.

- It has two modules
- 1.Command module(CM) 2.Lunar Excursion module(LEM)
- RAM size 256 , 1K ,2K words
- ROM size 4K,10K,36K words
- Clock frequency is 1.024MHz
- 5000 ,3-input RTL NOR gates are used
- User interface is DSKY(display/Keyboard)

First Mass Produced Embedded System: *Autonetics **D-17** Guidance computer for Minuteman-I missile*

Apollo Guidance Computer

Apollo Guidance Computer and DSKY

Invented by	Charles Stark Draper Laboratory
Manufacturer	Raytheon
Introduced	August 1966; 57 years ago
Discontinued	July 1975; 48 years ago
Type	Avionics Guidance computer
Processor	Discrete silicon integrated circuit (IC) chips (RTL based)
Frequency	2.048 MHz
Memory	15-bit wordlength + 1-bit parity 2048 words RAM (magnetic-core memory) 36,864 words ROM (core rope memory)[1]

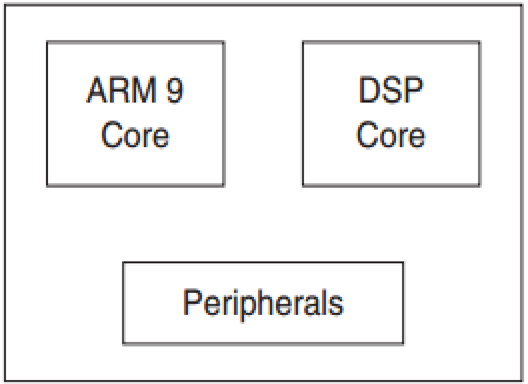


Figure 1.7 | Typical embedded dual core setup

Current Trends

i) **Multi-core processors:** It has become very clear that trying to improve processor performance by increasing clock frequencies is fraught with difficulties, because the direct result of higher clock frequency is high power dissipation. Thus, the option of using more than one processor core (at lower clock frequencies) is being tried out. Thus, the current smart phones and gaming consoles use multi-core processors. It may be understood that if there are two cores, one may be a DSP core while the other is a general purpose core. The design of multi core systems requires new design environments which are being developed at a rapid rate.

ii) **Embedded and real-time operating systems:** With the emergence of complex applications, many new embedded and real-time operating systems have become popular. Linux has emerged as a popular embedded OS, and others like Android and newer versions of Symbion have came up for mobile applications and handheld devices.

iii) **Newer areas of deployment of embedded devices:** Embedded devices have applications in the entertainment, healthcare and automotive segments. Besides that, there are applications in the communication and military fields as well. Research and development in these fields is going ahead.

ARM Processors and Controllers

Reduced Instruction Set Architecture (RISC)

- RISC is the way to make hardware simpler.
- simplify hardware by using an instruction set composed of a few basic steps for loading, evaluating, and storing operations just like a load command will load data, a store command will store the data.

Advantages of RISC

- **Simpler instructions:** RISC processors use a smaller set of simple instructions, which makes them easier to decode and execute quickly.
- **Faster execution:** Because RISC processors have a simpler instruction set, they can execute instructions faster than CISC processors.
- **Lower power consumption:** RISC processors consume less power than CISC processors, making them ideal for portable devices.

Characteristics of RISC

- Simpler instruction, hence simple instruction decoding.
- Instruction comes undersize of one word.
- Instruction takes a single clock cycle to get executed.
- More general-purpose registers.
- Simple Addressing Modes.
- Fewer Data types.
- A pipeline can be achieved.

Introduction to ARM

- ARM is a family of RISC instruction set architectures for computer processors.
- very popular 32-bit processor.
- ARM stands for '**Advanced RISC Machine'.**
- 'Acorn RISC Machine' manufactured by **Acorn Computers** Ltd., Cambridge, England, in 1985.
- Bits: 32-bit, 64-bit Design: RISC Open: Proprietary

Key Characteristics of ARM Architecture

1. **RISC Architecture**: This design philosophy focuses on a simplified set of instructions that can be executed very quickly, leading to high efficiency and performance.
2. **Low Power Consumption**: ARM processors are renowned for their energy efficiency, making them ideal for battery-powered devices like smartphones and tablets.
3. **Scalability**: ARM designs range from high-performance processors for servers and desktop computers to highly efficient microcontrollers for embedded systems.

History of the ARM Processor

- ARM has revolutionized the semiconductor industry with its innovative processor designs and licensing model.
- Its processors are found in billions of devices worldwide, from smartphones to embedded systems.
- ARM is well-positioned to play a pivotal role in the future of computing, driving advancements in various industries through its efficient and scalable processor architectures.

History and Milestones

- **1983**: The ARM project began within Acorn Computers, aiming to create a new microprocessor for the BBC Micro computer.
- **1990**: ARM Ltd was founded as a joint venture between Acorn Computers, Apple, and VLSI Technology.
- **1991**: The ARM6 processor was released, marking ARM’s entry into the microprocessor market.
- **1998**: ARM Holdings went public, listing on the London Stock Exchange and NASDAQ.
- **2000s**: ARM processors gained dominance in the mobile phone market, particularly with the ARM7 and ARM11 families.
- **2016**: ARM was acquired by the Japanese conglomerate SoftBank Group.
- **2023**: ARM returned to the public markets through an initial public offering (IPO) on NASDAQ.

- ARM1(1985) which had less than 25,000 transistors, and operated at 6 MHz
- ARM2 (in 1987) with 30,000 transistors.
- Intel/Motorola’s processor
- ARM3, ARM4 and ARM5 (1990)
- **ARM6, ARM7:** mobile phones, PDAs, IPods, computer hard disks,
- ARM made rapid strides in the 32-bit embedded market, accounting for a very high percentage of applications in the high-end embedded systems market.
- ARM family (ARM9, ARM10, ARM11, Cortex) have been built on the success of the ARM7 processor.
- advanced features have been added to the ARM processor, but the core has remained more or less the same

Technological Contributions

- **Mobile Devices**: ARM processors are the backbone of most smartphones and tablets, known for their balance of performance and power efficiency.
- **Embedded Systems**: ARM's Cortex-M series is widely used in microcontrollers for IoT (Internet of Things) devices.
- **Computing and Servers**: The ARM Cortex-A series targets high-performance applications, and ARM-based servers are becoming more common due to their energy efficiency.

Future Prospects

ARM continues to innovate, focusing on expanding its presence in areas like:

- **Automotive**: With advanced driver-assistance systems (ADAS) and autonomous vehicles.
- **Artificial Intelligence**: Developing processors optimized for AI and machine learning applications.
- **Edge Computing and IoT**: Enhancing capabilities for connected devices and edge computing solutions.

The ARM Core

- The core is the **‘processing unit’** or the **‘computing engine’**
- Design core and licenses this IP (Intellectual Property) to others.
- does not ‘fabricate’ the chip, but sells only the design.
- buyer can also modify the basic design to a minor extent.
- ARM sells its IP.
- **Soft IP:** design is sold as RTL (VHDL/Verilog code)
- **Hard IP:** it means the buyer gets only the layout or the net list (connection of nets or electronic wires).
- ARM the company does not ‘fabricate’ ARM chips.
- ARM chips and boards of various companies—Samsung, Philips, Atmel,

Texas Instruments, ST Microelectronics and so on—the list is very long.

Business Model

ARM’s business model revolves around licensing:

- **IP Licensing**: ARM licenses its processor designs to semiconductor companies. These licensees can then develop and manufacture their own chips based on ARM’s designs.
- **Royalties**: ARM receives royalties on every chip produced using its designs, providing a steady revenue stream.

The ARM Microcontroller

- on-chip peripherals.
- To the ARM core, peripherals are added and thus it becomes a ‘microcontroller’ or an MCU (microcontroller unit)
- peripherals added, depends on the requirements of the buyer of the IP.
- varying number of peripherals for ARM processors supplied by different companies.
- more peripherals, more powerful.
- When a chip has the core and the necessary peripherals to perform as a system, it is called a **System on Chip (SoC)**
- **‘ARM SoC’**

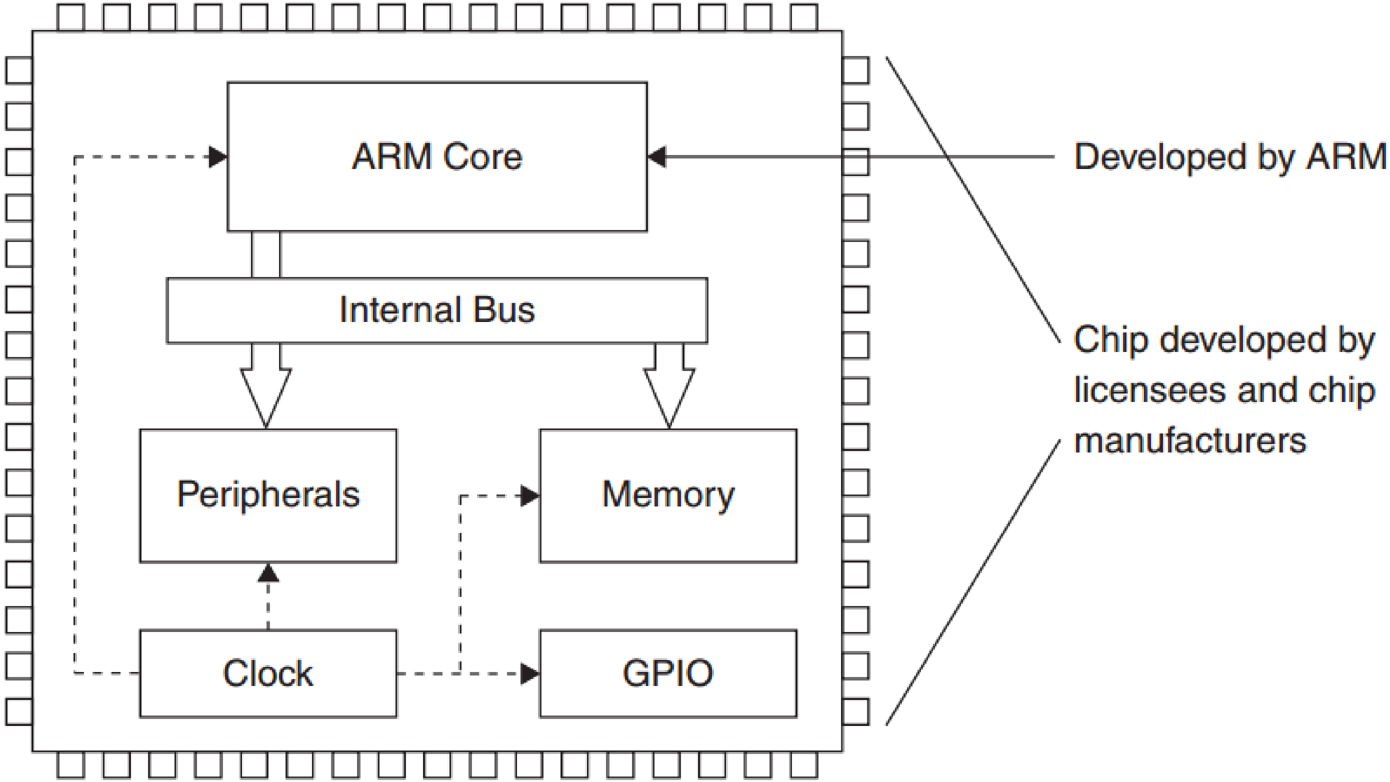


Figure 10.1 | ARM SoC—core with peripherals

Advanced Features of ARM Processor

- ARM core design.
- more and more features added.
- Features ‘standard’ and some are still optional.
- Naming conventions were adopted

Naming Conventions for ARM:

ARM {x}{y}{z}{T}{D}{M}{I}{E}{J}{F}{H}{S}

1. Thumb:

- 16-bit instruction set called ‘Thumb’
- all applications do not need the full power of 32-bit ARM instructions.
- The higher the amount of code that can be contained in unit area of memory, the
- higher is the code density.
- facility for mixing ARM and THUMB instructions, this is called ‘ARM THUMB
- interworking’.

ii) **MMU and MPU:** These are two aspects related to memory. One is the 'memory management unit' and the other is the 'memory protection unit'. Such units are mandatorily available in all advanced desktop processors (like Pentium), but for embedded systems, the necessity of such units is dictated by the product for which the processor is to be used. Thus, we have some ARM processors with both MPU and MMU, and others with one or neither of them.

iii) **Cache:** The first ARM processor with a cache was ARM3. It had an on-chip cache of 4 KB. ARM 7 had a cache of 8 KB which was improved in ways other than just the size. Current ARM processors have cache as a standard component.

iv) **Debug interface:** There is an on chip unit for testing called the JTAG interface. JTAG stands for 'Joint Test Action Group' and defines a set of standards for testing the functionality of hardware. For any chip/system there is a set of scan cells located at the boundaries and there are specific signals designed to enable ctesting' of the device. Such a unit is called the JTAG debug interface, and some ARM chips have this facility.

Embedded ICE macrocell:

while other units (peripheral units as well) may also be added as 'macrocells'. Some processors have an embedded ICE (In Circuit Emulator) macrocell to enable testing. This unit is powered by breakpoint and watch point registers and control and status registers. All this together can work to halt the ARM core to read status and thus do active debugging.

vi) **Fast muliplier:** Even though ARM is a RISC processor, there are many features in it which do not conform exactly to the RISC philosophy. Having dedicated hardware for complex operations is one such deviation. Multiplication is a complex operation, and for fast multiplication, there may be a fast multiplier unit.

vii) **Enhanced instructions:** Most advanced embedded systems require DSP operations, and for that a DSP unit with complex arithmetic operations, may be made available on the chip.

vii) **Jazelle DBX (Direct Bytecode eXecution):** allows some ARM processors to execute Java bytecode in hardware as a third execution state along with the existing ARM and Thumb mode. This is useful to increase the execution speed of Java ME games and applications. ARM claims that such Java applications get run in hardware (rather than software) so that 'more speed' is achieved.

viii) **Vector floating point unit:** This implies hardware support for floating point computation.

ix) **Synthesizable:** If an ARM processor is synthesizable, it means that its RTL code is available with the licensee, using which extensions and modifications are possible to the basic core.

Table 10.1 | Early Naming Conventions for ARM

ARM {x}{y}{z}{T}{D}{M}{I}{E}{J}{F}{H}{S}	
x	Family (7, 8, 9, 10, 11, . . .)
y	Memory management/protection unit
z	Cache
T	Thumb 16-bit decoder
D	JTAG debug
M	Fast Multiplier
I	EmbeddedICE macrocell
E	DSP Enhanced instructions (assumes TDMI)
J	Jazelle
F	Vector Floating-point Unit
S	Synthesizable Version

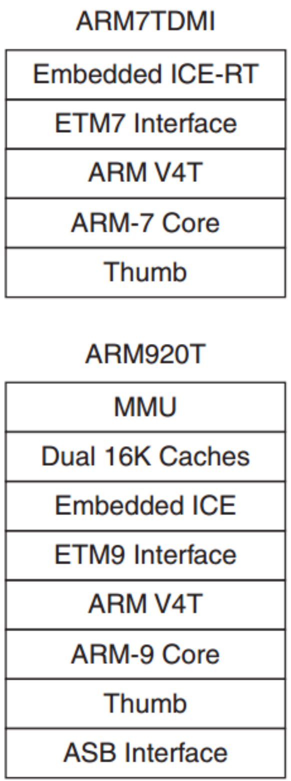


Figure 10.2 | Two ARM cores

Table 10.2 | Variants of the ARM Processor

Processor Name	Architecture Version	Memory Management Features	Other Features
ARM7TDMI	ARMv4T		
ARM7TDMI-S	ARMv4T		
ARM7EJ-S	ARMv5E		DSP, Jazelle
ARM920T	ARMV4T	MMU	
ARM922T	ARMv4T	MMU	
ARM926EJ-S	ARMv5E	MMU	DSP, Jazelle
ARM946E-S	ARMv5E	MPU	DSP
ARM966E-S	ARMv5E	DSP	
ARM968E-S	ARMv5E		DMA, DSP
ARM966HS	ARMv5E	MPU (optional)	DSP
ARM1020E	ARMv5E	MMU	DSP
ARM1022E	ARMv5E	MMU	DSP
ARM1026EJ-S	ARMv5E	MMU or MPU	DSP, Jazelle
ARM1136J(F)-S	ARMv6	MMU	DSP, Jazelle
ARM1176JZ(F)-S	ARMv6	MMU+TrustZone	DSP, Jazelle
ARM11MPCore	ARMv6	MMU+Multiprocessor Cache Support	DSP, Jazelle
ARM1156T2(F)-S	ARMv6	MPU	DSP
Cortex-M0	ARMv6-M		NVIC
Cortex-M1	ARMv6-M	FPGA TCM interface	NVIC
Cortex-M3	ARMv7-M	MPU (optional)	NVIC

(Courtesy: The Definitive Guide to ARM Cortex-M3 by Joseph Liu, Newnes Publications)

Architecture Versions

- Architectural features have also been enhanced
- More powerful Versions v4 and v4T are the early versions
- Later versions are v5, v5E, v6 and v7.

Table 10.3 | Features of the Architecture Variants of ARM

Architecture Versions	Features
v4	ARM instructions only
v4T	THUMB instructions also added
v5	More advanced ARM and THUMB instructions
v5E	Advanced ARM instructions and enhanced DSP instructions
v6	Advanced ARM and THUMB. SIMD and memory support instructions added
v7	THUMB-2 technology, in which both 16-bit and 32-bit instructions are supported, and there is no need to switching between ARM and THUMB instruction sets

ARM CORTEX (A , R, M)

- ARM2, which was the first one to be commercially produced.
- ARM7: dominant player in the 32-bit embedded processor market.
- **ARM9, ARM10 and ARM11:** boasted of more and more computing powers.
- CORTEX series which has the architecture v7 version.

ARM CORTEX A

i) **The A profile:** This profile which has the ARMv7-A architecture is meant for high end applications. It is meant to handle complex applications with high-end embedded operating systems, and typical applications requiring such a profile are mobile phones and video systems.

ARM CORTEX R

ii) **The R profile:** This profile which has the ARMv7-R architecture has been designed for high-end applications which require real-time capabilities. Typical applications are automatic braking systems and other safety critical applications.

iii) **The M profile:** This profile which has the ARMv7-M architecture has been designed for deeply embedded microcontroller type systems. This is to be used in industrial control applications where a large number of peripherals may have to be handled and controlled.

Introduction to Embedded C

- Extension of C
- Designed to support the diverse and specific needs of embedded systems.
- writing software that directly interacts with the hardware of microcontrollers,

Ex.LPC2148.

Key Concepts in Embedded C

1. **Direct Hardware Access:**
   - Embedded C provides mechanisms to access hardware registers directly, which is essential for controlling peripherals and other hardware-specific features.
2. **Memory Management:**
   - Unlike general-purpose programming, embedded systems often have limited memory, so efficient memory usage is critical. This includes careful use of RAM and Flash memory.
3. **Real-time Constraints:**
   - Embedded systems often have real-time constraints, requiring the software to perform tasks within strict timing bounds.
4. **Concurrency:**
   - Embedded systems may need to handle multiple tasks concurrently, which can be managed through interrupts, timers, and real-time operating systems (RTOS).

Common Embedded C Constructs

- **Registers and Memory-mapped I/O:**
- Accessing hardware registers usually involves defining pointers to specific memory

```
#define GPIO0DIR  (*(volatile unsigned long *)0xE0028008)
#define GPIO0SET  (*(volatile unsigned long *)0xE0028004)
```

- **Interrupt Service Routines (ISRs):**

  - ISRs handle hardware interrupts and must be written with care to ensure they execute quickly and safely.

```
void __attribute__ ((interrupt)) IRQ_Handler(void) {
   // ISR code
}
```

- **Peripheral Initialization:**

Initializing peripherals (e.g., UART, ADC, timers) is a common task

```
void UART_Init(void) {
   // Set baud rate, configure UART pins, enable UART, etc.
}
```

- **Main Loop:**

```
int main(void) {
    UART_Init();
    while (1) {
        // Main loop code
    }
}
```

Blinking an LED on LPC2148

Embedded C program to blink an LED connected to a GPIO pin on the LPC2148

```c
#include <lpc214x.h> // Device-specific header file

#define LED_PIN 10

void delay(unsigned int count) {
    unsigned int i;
    for (i = 0; i < count; i++);
}

int main(void) {
    // Configure GPIO pin as output
    IO0DIR |= (1 << LED_PIN);

    while (1) {
        // Turn on the LED
        IO0SET = (1 << LED_PIN);
        delay(100000);

        // Turn off the LED
        IO0CLR = (1 << LED_PIN);
        delay(100000);
    }

    return 0;
}
```

Development Environment

1. **Compiler and IDE:**
    - **Keil MDK (Microcontroller Development Kit)**: A popular IDE with support for ARM microcontrollers.
    - **IAR Embedded Workbench**: Another widely used development environment.
    - **NXP's LPCXpresso**: Specifically for LPC microcontrollers.
    - **GNU ARM Toolchain**: Open-source tools for ARM development.
2. **Debugging Tools:**
    - In-circuit emulators (ICE), JTAG, and SWD (Serial Wire Debug) interfaces are commonly used for debugging embedded systems.
3. **RTOS:**
    - Real-time operating systems like FreeRTOS can be used to manage tasks and resources in more complex embedded systems.

Introduction to LPC2148 microcontroller

- The LPC2148 is a widely used microcontroller from NXP (formerly Philips).
- It is based on the ARM7TDMI-S core and is known for its versatility, making it popular for a variety of embedded systems applications.

Key Features:

1. **ARM7TDMI-S Core:**
    - 32-bit ARM7TDMI-S processor with real-time emulation and embedded trace support.
2. **Memory:**
    - 32 to 512 kB of on-chip Flash program memory with In-System Programming (ISP) and In-Application Programming (IAP).
    - 8 to 40 kB of on-chip static RAM.

1. **Clock and Power:**
    - 60 MHz maximum CPU clock available from programmable on-chip PLL with settling time of 100 μs.
    - On-chip integrated oscillator operates with an external crystal in range from 1 MHz to 25 MHz.
    - Power-saving modes (Idle and Power-down).
2. **Digital and Analog I/O:**
    - Two 32-bit general-purpose I/O (GPIO) ports.
    - Up to 45 fast GPIO lines with up to nine edge or level sensitive external interrupt pins.
    - 10-bit ADC with 14 channels and conversion rates up to 400 kHz.
    - Two 10-bit DACs.

1. **Timers and PWM:**
   - Two 32-bit timers with a total of four capture inputs and four compare outputs.
   - PWM unit with six output pins.
   - Real-time clock (RTC) with optional battery backup.
2. **Communication Interfaces:**
   - Two UARTs with fractional baud rate generation and internal FIFO.
   - Two I2C-bus interfaces (one with open-drain and one with standard I/O pins).
   - Two SPI interfaces.
   - SSP interface (SPI, 4-wire SSI, and Microwire) for synchronous serial communication.

USB:

- Full-speed USB 2.0 compliant device controller with 2 kB of endpoint RAM.

Other Features:

- Watchdog timer.
- Low power real-time clock with independent power and dedicated 32 kHz clock input.
- Brown-out detection with separate thresholds for interrupt and forced reset.
- On-chip PLL allows CPU operation up to the maximum CPU rate without the need for a high-frequency crystal.

Applications:

- Embedded control systems
- Data acquisition systems
- Industrial automation
- Consumer electronics
- Medical devices
- Automotive applications

Development Tools:

- **Keil MDK (Microcontroller Development Kit)**: Provides comprehensive development tools for ARM-based microcontrollers.
- **IAR Embedded Workbench**: Another popular development environment for ARM microcontrollers.
- **NXP's LPCXpresso IDE**: An integrated development environment specifically for LPC microcontrollers.
- **GNU ARM Toolchain**: A free and open-source set of tools for ARM development.

Thank you