AMIT LEARNING

INTRODUCTION TO EMBEDDED SYSTEMS

CONTENTS

1 What is embedded systems

1.1 Embedded systems definition

Embedded system : A combination of multiple systems merged together at one system.

Example_1 : PCB + CODE = TV

Hardware system + Software system = Embedded system

Example_2 : PCB + CODE + Motors = Robotic Arm

Hardware system + Software system + Mechanical system = Embedded systems

1 What is embedded systems

1.2 Embedded systems applications

1	ROBOTS	2	HOME APP.	3	MILITARY	4	BIO MEDICAL	6	AUTOMOTIVE	7	OTHER
											
1	ROV	1	HOME DEVICES	1	JETS	1	X -RAY	1	MOTOR CONTROL	1	ATM MACHINES
											
2	DRONES	2	SMART HOMES	2	WEAPONS	2	E C G	2	DASHBOARD	2	CAMERAS
											
3	ROBOTS	3	SMART CITIES	3	SATELLITES	3	VENTILATOR	3	SELF DRIVING	3	ROUTERS

AMIT

EMBEDDED SYSTEMS CHARACTERISTIC

- Reliability.
- Efficiency.
- Tightly-constrained.
- Single-functionality
- Complex functionality.
- Safety-critical.
- Maintainability
- Interactive System
- Strong association between the H W and SW.

EMBEDDED SYSTEMS CHARACTERISTIC

➢ Reliability:

- Personal computer programs such as word processors and games do not need to achieve the same standard of reliability that a microcontroller application must. Errors in programs such as word processors may result in errors in a document or loss of data. An error in an embedded system application such as a TV remote control or compact disc player will result in a product that does not work and consequently does not sell.

- An error in embedded system application such as an Antilock Braking System (ABS) or Breaking Assistant (BA) or autopilot could be fatal.

EMBEDDED SYSTEMS CHARACTERISTIC

EMBEDDED SYSTEMS CHARACTERISTIC

- efficiency:
- Considered in real time applications.
- A real time application is one in which must be able to act at a speed corresponding with the occurrence of an actual process. Must compute certain results in certain real-time without delay.
- Some Embedded systems may have real-time performance constraints that must be met, for reasons such as safety and usability. others may have low or no timing performance requirements, allowing the system hardware to be simplified to reduce costs.

EMBEDDED SYSTEMS CHARACTERISTIC

- Tightly-constrained:
- Embedded Systems normally come with constraints in hardware resources:
  - Processing (speed)
  - Storage (Code Size)
  - Memory(Data)
  - Most of the time it targets real time objectives, this means,
    - It needs to be fast and efficient (Response Time).
    - It needs to be predictable (execution time known ahead, and almost constant)
  - Power limited (battery operated devices).
  - Cost
  - Size.
  - The developer has to deal with all of these constraints •Development should take into consideration code efficiency and code foot print. •Debugging tools are “closer to the metal” •Special attention to power consumption in some cases.

EMBEDDED SYSTEMS CHARACTERISTIC

- Single-functionality:
- Dedicated to perform a specific function and include processors dedicated to specific function.
- Complex functionality :
- Often have to run sophisticated algorithms or multiple algorithms Cell phone, laser printer, automotive, etc.
- Safety-critical:
- Must not endanger human life and the environment.

EMBEDDED SYSTEMS CHARACTERISTIC

- Maintainability :
- The ability to modify the system after its initial release and enhance its performance like execution time, code and memory size.
- Interactive System :
- An Embedded system should be as interactive as possible so that it is easily understandable, Operated and Handled by a user and provide ease of task.
- Strong association between the H W and SW.

1 What is embedded systems

1.3 Embedded systems companies

1	ROBOTS
1	HONDA
2	BOSTON DYNAM.
3	CANVAS

2	HOME APP.
1	GOOGLE
2	SAMSUNG
3	L G

3	MILITARY
1	ETC USA
2	N.GRUMMAN
3	A O I

4	BIO MEDICAL
1	PHILIPS
2	SIEMENS
3	VENTILATOR

6	AUTOMOTIVE
1	BOSCH
2	MENTOR
3	AVELABS

7	OTHER
1	HONEYWELL
2	SONY
3	CANON

2 AMIT Diploma

2.1 Outlines

1	INTRODUCTION
1	INTR. TO E.S

2	C - PROGRAMMING
1	HELLO C
2	CONTROL FLOW
3	FUNCTIONS
4	POINTERS
5	ARRAY
6	UDDT
7	ALGORITHMS
8	DATA STRUCTURE

3	TOOLING
1	TOOLING_1
2	GITHUB

4	ISTQB
1	SESSION_1
2	SESSION_2
3	SESSION_3
4	SESSION_4

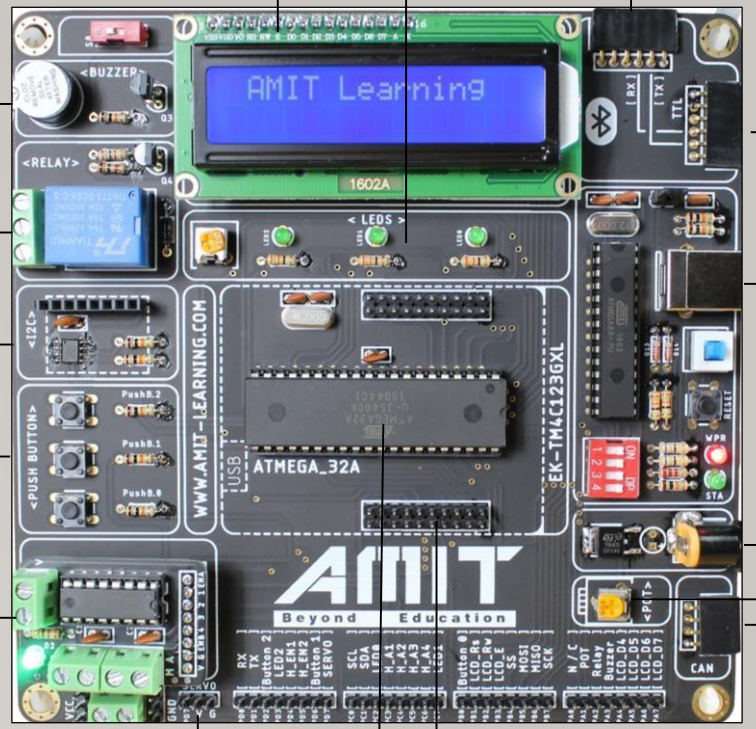
5	INTERFACING
1	EMBEDDED C
2	COMPUTER ARC.
3	DIO
4	LCD
5	KEYPAD
6	INTERRUPT
7	ADC
8	TIMER_1
9	TIMER_2
10	UART
11	SPI
12	I2C
13	ARM

6	RTOS
1	SESSION_1
2	SESSION_2
3	SESSION_3
4	SESSION_4

7	AUTO.BUS TECH.
1	CAN
2	LIN

2 AMIT Diploma

2.2 Amit KIT



P LCD / 7-SEGMENTS
2*16 LCD / (2) 7 SEG.

3 GREEN LED

N BLUETOOTH MOD.
OUTLET

A. BUZZER
5V

B. RELAY MODULE
10A /220 VAC

C. I2C
COMM. PROTOCOL

D. BUTTONS
3 PUSH BUTTONS

E. H-BRIDGE
FOR MOTOR CONTROL

M TTL
OUTLET

L PROGRAMMER
WITH USB CABLE

K POWER INPUT
EXTERNAL

J POT
10 K

I CAN
OUTLET

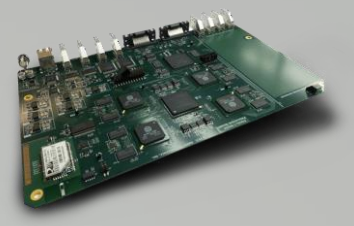
F. SERVO
OUTLET

G MICROCONTROLLER
ATMEGA32

H TIVA-C
OUTLET

3 Embedded systems vs General purpose systems

3.1 Embedded systems vs General purpose systems

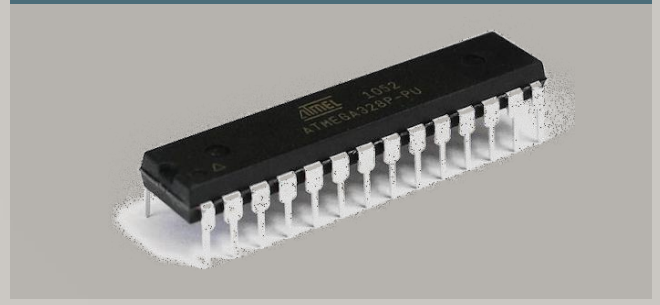


EMBEDDED SYSTEMS		GENERAL PURPOSE SYSTEMS
A system which is combination of a specific purpose hardware and a software for exciting a specific application	FUNCTIONALITY	A system which is combination of a generic hardware and a general purpose operating system for exciting a variety of applications
Known at designing time	APPLICATIONS	Unknown at designing time
Not programmable by end user	END USER	Programmable by end user
Low	SPEED	High
Low	POWER CONSUMPTION	High
Critical	RESPONSE TIME	Not Critical
Low	COST	High

4 Microcontroller components

4.1 Microcontroller components

MICROCONTROLLER



- 1 CPU
  - 1.1 ALU
  - 1.2 CU
  - 1.3 ACCUMELATOR REGISTER
  - 1.4 GENERAL PURPOSE REGESTERS
  - 1.5 SFR REGISTER
    - 1.5.1 IR
    - 1.5.2 PC
    - 1.5.3 SP
    - 1.5.4 PSW
- 2 MEMORY UNIT
  - 2.1 RAM
  - 2.2 ROM
- 3 GPIO S
- 4 PERIPHERALS
  - 4.1 TIMERS
  - 4.2 COUNTERS
  - 4.3 ADC
  - 4.4 DAC

4 Microcontroller components

4.2 CPU

1	CPU		Central Processing Unit Responsible of all the computing process Fetch, decode and execute all program instructions Direct the flow of data to / from memory unit Most CPUs are synchronous, this means it depends on external clock
	1.1	ALU	Arithmetic Logic Unit Responsible of all the arithmetic and logic and shifting instructions
	1.2	CU	Control Unit Convert the instructions stored in program memory (ROM) into codes which ALU can understands. Generate the control signal of this instruction.
	1.3	Accumulator Register	Is an SFR closely related to the operation of ALU
	1.4	General Purpose Registers	Also called Register File A group of registers the processor use them to store the data fetched from memory
	1.5	SFR Registers	
		1.5.1 IR	Instruction Register : Contain the instruction after getting it from program memory.
		1.5.2 PC	Program counter : Contain the address of the next instruction in program memory which the CPU will process
		1.5.3 SP	Stack Pointer : Hold the memory address of the next available location on the stack to store data
		1.5.4 PSW	Status Register : contains information about the result of the most recently executed arithmetic instruction

4 Microcontroller components

4.3 Memory Unit -> RAM Vs ROM

RAM		ROM
Random Access Memory	STANDS FOR	Read Only Memory
Read and Write while run time	READ / WRITE	Read only while run time
Used to read and write data while run time	APPLICATIONS	Used to store the
code Volatile “Lost all the data when power is off “	VOLATILITY	Non Volatile “Can keep all data when power is off”
Fast	SPEED	Slow
SRAM - DRAM - NVRAM	TYPES	ROM - PROM - EPROM - EEPROM - FLASH

4 Microcontroller components

4.3 Memory Unit -> RAM Types

1	SRAM	Stands for Static RAM
		Consists of multiple transistors for each memory cell
		SRAM does not required a periodic refreshing
		Faster than DRAM
		Low power consumption
		More expensive than DRAM
		Most used in microcontrollers because limited power source
2	DRAM	Stands for Dynamic RAM
		Consists of Transistors and capacitors
		DRAM need a periodic refreshing " a charge every few milliseconds to make capacitors retain its charge "
		Slower than SRAM
		High power consumption
		Cheaper than SRAM
		Most used in PCs because open power source

4 Microcontroller components

4.3 Memory Unit -> ROM Types

1	MROM	MASKED PROGRAMABLE ROM The microcontroller chip burned by the manufacture.
2	PROM OR (OTP)	ONE TIME PROGRAMMABLE Can be burned one time by the developer
3	EPROM	ERASABLE PROGRAMMABLE ROM Can be erased and programmed by the developer. The erasing process done using ultraviolet light.
4	EEPROM	ELECTRICALLY ERASABLE PROGRAMMABLE ROM Can be erased and programmed by the developer. The erasing process done using electric charge. For example in (ATMEGA32) the finite number of writing / erasing is 100,000 times. EEPROM write the data by byte.
5	FLASH	FLASH ELECTRICALLY ERASABLE PROGRAMMABLE ROM Can be erased and programmed by the developer. The erasing process done using electric charge. For example in (ATMEGA32) the finite number of writing / erasing is 10,000 times. Developed from EEPROM EEPROM write the data by sector.

4 Microcontroller components

4.4 GPIOS

GPIOS stands for General Purpose Input Output

This microcontroller (ATMEGA32) s

Consist of 40 pin

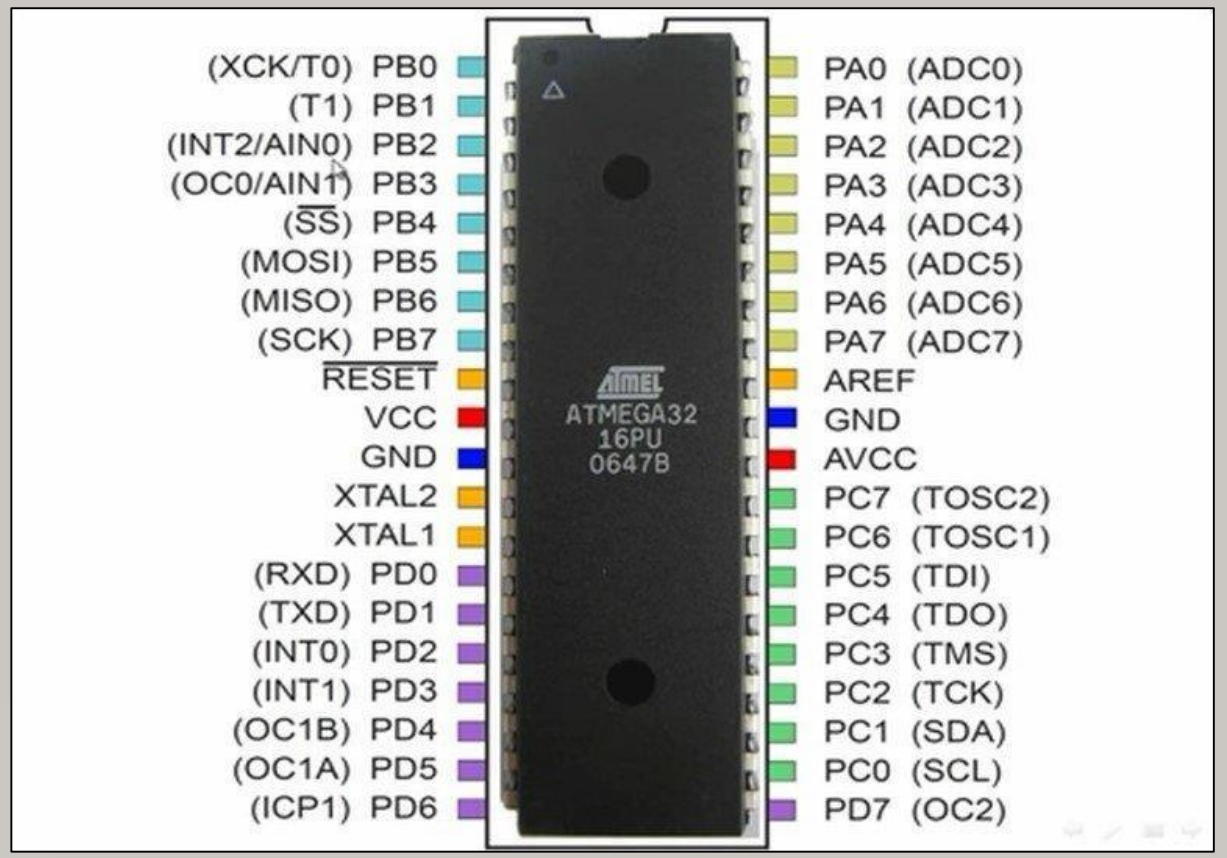
32 pins from the mare general purpose I/o

That mean you can make each pin from these eit

32 pins her input pin or output pin and

that's why we call it general purpose input pi

output ns



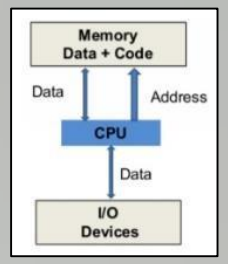
4 Microcontroller components

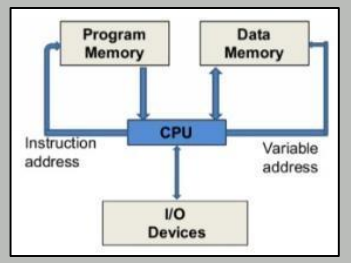
4.5 Peripherals

1	ADC	ANALOG DIGITAL CONVERTER
		To convert any analog signal like sound waves to a digital signal (0,1)
2	DAC	DIGITAL ANALOG CONVERTER
		To convert the digital signal into analog
3	UART	UNIVERSAL ASYNCHRONOUS RECEIVER/ TRANSMITTER
		A communication protocol to send or receive data from microcontrollers
4	TIMERS	Very important peripheral in any embedded system application

5 CPU

5.1 CPU Architecture





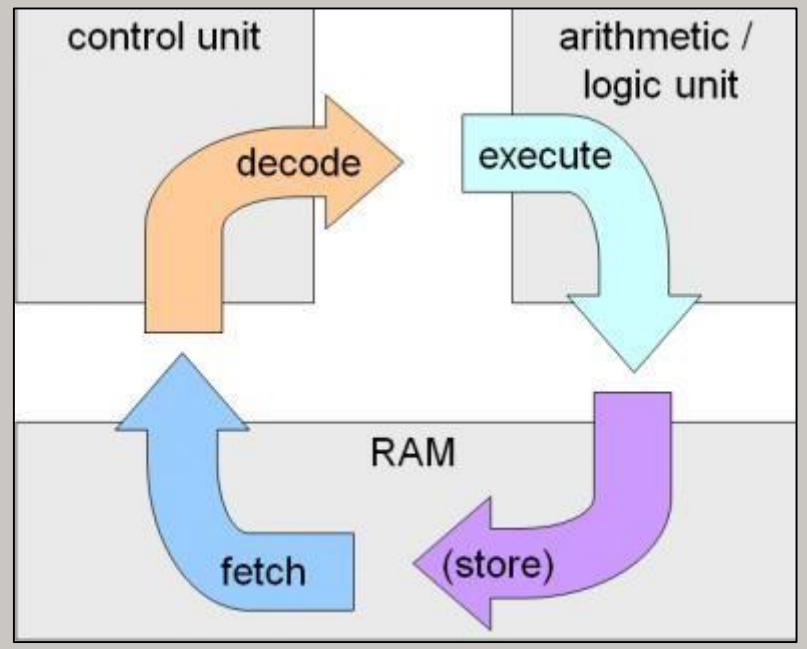
VON - NEUMANN ARCHITECTURE		HARVARD ARCHITECTURE
RAM + ROM in one block connected to CPU	DESCRIPTION	RAM and ROM separated an CPU between them
One bus Connect CPU to RAM and ROM	BUSSES	Two Busses #1 connect CPU to ROM ,#2 connect CPU to RAM
Slow	COMMUNICATIONS	Fast
CPU can read instruction from ROM or write data in RAM	READ / WRITE	CPU can read instruction from ROM and write data in RAM at the same time
Simple	DESIGN	Comple
Not available	PIPELINING	x
		Availabl

5 Microcontroller components

5.2 Instruction Cycle

All the time the CPU does these 4 steps while runtime We call it the instruction cycle

1 – fetch data from RAM
2 – Decode this data in CU
3 – Execute the instruction in ALU
4 – Write back or store the result of this operation in RAM



5 What is embedded systems

5.3 Pipelining

To understand the pipelining concept let us show this example :
In a car factory there are 30 workers, 10 workers in car body sector, 10 workers in motor sector, and 10 workers in paint sector
To produce one car it takes 30 minutes, 10 minutes in every sector

 CAR BODY SECTOR [10 workers]	 CAR MOTOR SECTOR [10 workers]	 CAR PAINT SECTOR [10 workers]
✓	✗	✗
✗	✓	✗
✗	✗	✓
✓	✗	✗

The first 10 workers working on car body sector to produce the car body
After finishing it, they stop working and the second 10 workers start working on car motor to install it in the car body
After finishing it, they stop working and the third 10 workers start working on car painting.
In this case there is no pipelining so in 30 minutes we produce one car and one body

AMIT

5 What is embedded systems

5.3 Pipelining

CAR BODY SECTOR [10 workers]

CAR MOTOR SECTOR [10 workers]

CAR PAINT SECTOR [10 workers]

✓	✗	✗
✓	✓	✗
✓	✓	✓
✓	✓	✓

The first 10 workers working on car body sector to produce the car body
After finishing it, they do not stop but starting in another body, at the same time the second 10 workers start working on car motor to install it in the first body
After finishing it, they do not stop but working on another motor, and the third 10 workers start working on car painting.
In this case after applying pipelining in 30 minutes we produce two cars and one body and one body with motor
By applying the pipelining concept in software we execute more instruction cycles at the same time

FETCH	DECODE	EXECUTE

6 Vendors / IDE

6.1 Microcontroller vendors

INTEGRATED DEVELOPMENT ENVIRONMENT

MICROCHIP

TEXAS INSTRUMENTS

INFINEON

SILICON LABS

RENESAS

CYPRESS

AMIT

6 Vendors / IDE

6.2 IDE software

INTEGRATED DEVELOPMENT ENVIRONMENT

CODEBLOCKS

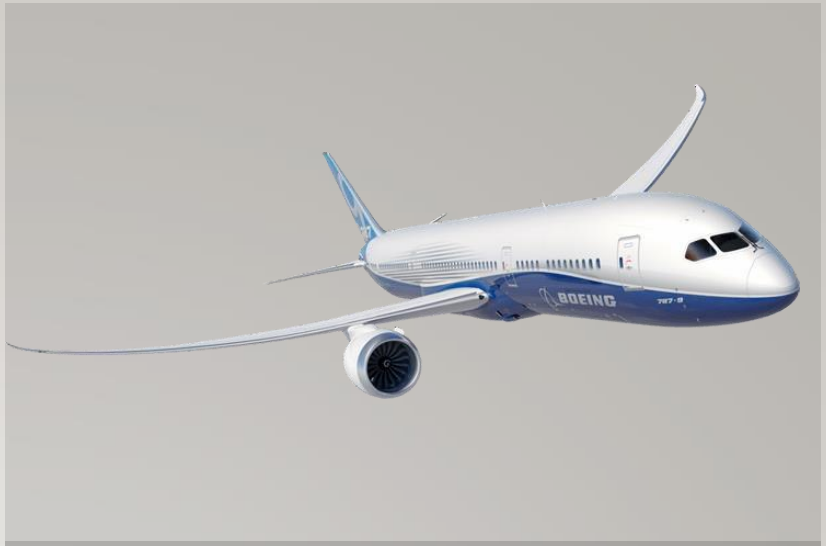
ECLIPSE

ATMEL STUDIO

2 What is embedded systems

2.1 Embedded systems definition

WHICH PRODUCT HAS MORE LINES OF CODE?



BOEING 787

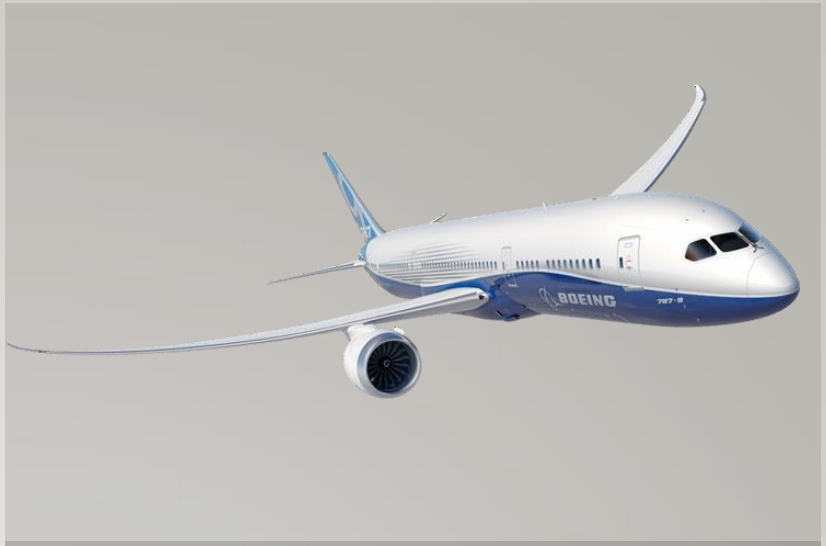
MERCEDES - BENZ S - CLASS

F - 22 RAPTOR

2 What is embedded systems

2.1 Embedded systems definition

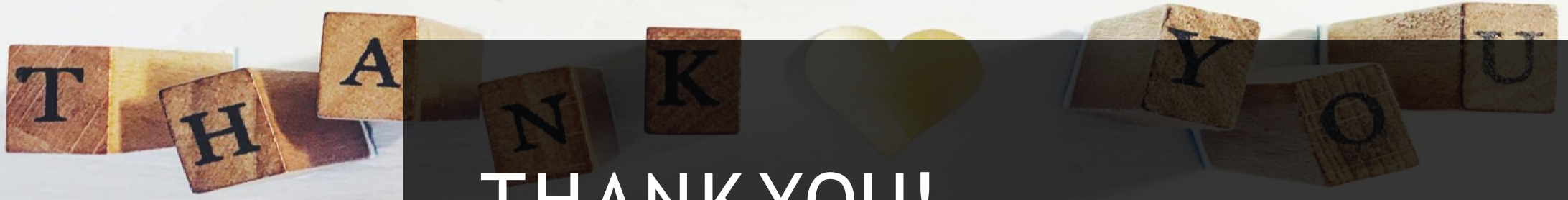
WHICH PRODUCT HAS MORE LINES OF CODE?



6.5 MILLION

100 MILLION

1.7 MILLION



THANK YOU!

AMIT LEARNING

[NEXT SESSION IS -> C PROGRAMMING : SESSION_1 : HELLO C]