```
0x54 0x68 0x69 0x73 0x20 0x69 0x73 0x20 0x65
0x61 0x73 0x69 0x65 0x72 0x20 0x3a 0x29
```

- Does my microcontroller run C language code?
- Where does it store its instructions?
- How does the microcontroller know where to begin?
- What happens during a reset?
- How does a computer *program* a microcontroller
- Are there going to be burgers in today’s class?

MICROCONTROLLER PROGRAMMING

Daniel Martínez

I7266 - Programación de Sistemas Embebidos

CUCEI - UDG

MEMORY

Do you remember?

The 21st night of September?

Love was changing the minds of pretenders

While chasing the clouds away

MEMORY CLASSIFICATIONS

- Volatility
- Access method
- Read/write characteristics
- Physical location
- Functionality
- Technology
- Usage in CPU
- Data organization

VOLATILITY

- Volatile Memory:
  - Loses its stored information when power is turned off. Examples include RAM (Random Access Memory).
- Non-Volatile Memory:
  - Retains data even when power is turned off. Examples include CDs, flash memory, and hard drives.

ACCESS METHOD

- Random Access:
  - Allows data to be accessed in any random order. (RAM)
- Sequential Access:
  - Requires accessing data in a sequential manner. Examples include tape drives and optical drives.

READ/WRITE CHARACTERISTICS

- Read-Only:
  - Typically used for storing permanent or semi-permanent data, and the data cannot be easily modified. Examples include some optical drives.
- Read/Write:
  - Allows both reading and writing of data. Examples include RAM and Flash memory.

PHYSICAL LOCATION

- Internal Memory:
  - Embedded within the microprocessor or closely connected to it. Examples include Cache memory or SRAM.
- External Memory:
  - Separate from the microprocessor and connected via buses. Examples include EEPROM, hard drives, etc.

FUNCTIONALITY

- Primary Memory:
  - Fast and used for temporary storage (e.g., RAM).
- Secondary Memory:
  - Slower but provides larger storage capacity (e.g., hard drives).

TECHNOLOGY

- Dynamic Random Access Memory (DRAM):
  - Requires periodic refresh to maintain data integrity.
- Static Random Access Memory (SRAM):
  - More stable and does not require frequent refreshing.
- Electrically Erasable Programmable ROM (EEPROM):
  - Non-volatile but needs UV light to be erased.
- Flash Memory:
  - Non-volatile memory commonly used in storage devices.

USAGE INSIDE CPU

- Cache Memory:
  - High-speed memory used by the CPU to store frequently accessed instructions and data.
- Registers:
  - Fast, small, and internal memory locations within the CPU.

HELLO WORLD!

Our first program

NOT SO FAST…

THE MINIMAL SYSTEM

What is needed for a microcontroller to work?

A minimal system refers to the basic set of components required to create a functional electronic system. The components in a minimal system may vary depending on the specific application, for embedded systems, some common elements include:

- Microcontroller or Microprocessor: The central processing unit responsible for executing instructions and controlling the overall operation of the system.
- Clock Source: A clock oscillator or crystal to provide the system with a timing reference, ensuring synchronization of operations.

- Memory: This includes both volatile memory (RAM) and non-volatile memory (ROM, Flash), for storing data and program instructions, respectively.
- Input/Output (I/O) Components: These could be sensors, actuators, communication interfaces, or other peripherals that allow the system to interact with its environment.
- Power Supply: A source of electrical power to provide the necessary voltage levels for the components in the system.
- Basic Support Components: Resistors, capacitors, and other passive components that are necessary for circuit stability and functionality.

ATMEGA328P MINIMAL SYSTEM

POWER SUPPLY

- VCC
- GND
- AVCC
- AREF

CLOCK SOURCE

- Low Power Crystal Oscillator
- Full Swing Crystal Oscillator
- Low Frequency Crystal Oscillator
- Internal 128kHz RC Oscillator
- Calibrated Internal RC Oscillator
- External Clock

RESET

HELLO WORLD!

No BS this time!

(HERE I WILL PASTE MY CODE)

EMBEDDED SYSTEMS

- Embedded systems are computing systems that are integrated into larger devices, products, or machinery to perform specific functions. They are designed to operate within the constraints of the system they are embedded in and are often dedicated to a single task or a narrow range of tasks.
- Applications: automotive systems, medical systems, personal health systems, smartphones and mobile devices, manufacturing devices, flight control systems, home appliances, network switches and routers, robots, public transportation, gaming, RFID devices, weather stations, agriculture monitoring systems, surveillance systems, Internet of Things, measuring devices, onboard computers, lighting, energy, musical instruments or devices, etc.

COMPUTER ARCHITECTURE

- This refers to the elements and its organization inside a computer system, encompassing structure and functionality.
- It is comprised, mainly, of the ISA design and the microarchitecture design.
- The ISA is the set of instructions that a system can execute, the data types it can work with, the addressing modes and other stuff related to how we can describe what we can ask a processing unit to do.
- The microarchitecture describes how a particular processing unit implements an ISA.
- Knowing about computer architectures allows us to design safer, faster, simpler, and overall, more efficient embedded systems.

ALU

- It is a combinational digital circuit that performs arithmetic and logic operations on binary integers.
- They usually have two data inputs, or operands.
- They usually have an opcode input to select the desired operation to execute.
- They usually connect to external status registers, at their input and their output, that contain useful information about the last operation performed.

NOTICES!

WhatsApp and classroom groups, missing data, class formality.

THANKS!

Please feel free to ask any related questions at any time.