

```
01010010 01100101 01100001  
01101100 01101100 01111001?
```

- What happens when you program in C?
- How does a processor do what I want?
- What does “instruction” mean?
- What is assembly and machine code?
- What happens when you program a microcontroller?
- How does a microprocessor execute a program?

WHAT HAVE WE LEARNED SO FAR?

WHAT IS AN EMBEDDED
SYSTEM?

HOW MANY EXAMPLES CAN
YOU LIST?

WHAT IS AN EMBEDDED
SYSTEM ARCHITECTURE?

WHAT IS IMPORTANT TO
KNOW ABOUT IT AND WHY?

WHAT DO EMBEDDED
SYSTEMS DO?

HOW DO THEY DO IT?

WHAT IS AN ALU?

MICROPROCESSOR INSTRUCTIONS

Daniel Martínez

I7266 - Programación de Sistemas Embebidos

CUCEI - UDG

2 DOUBLE WESTERN BACON
WITH FRIES AND COKE!

Interpreting instructions

BURGERS

(AGAIN)

(YES. I'M HUNGRY)

- When we ask for a burger, in the outside we just see an order appearing on the screen, a minute after, it's done.
- On the inside, when we order a burger, a series of steps is followed so the burger can be made.

ADDING TWO NUMBERS V3

ALU operation selection and execution example

ABSTRACTION LEVELS

PYTHON

- **Abstraction Level:** High-level language.
- **Readability:** High; Python emphasizes readability and simplicity.
- **Portability:** Generally portable, but Python interpreters are required on target systems.
- **Syntax:** Clear and concise, using indentation for code blocks.
- **Memory Management:** Automatic memory management (garbage collection).
- **Execution:** Interpreted language, but can be compiled to bytecode for improved performance.

C

- **Abstraction Level:** Mid-level language.
- **Readability:** Moderate; C provides a good balance between abstraction and control.
- **Portability:** Highly portable, but code may need adjustments for different architectures and depends on the compiler used.
- **Syntax:** Uses a more explicit and traditional syntax.
- **Memory Management:** Manual memory management (malloc/free).
- **Execution:** Compiled language, resulting in machine-specific executable code.

ASSEMBLY

- **Abstraction Level:** Low-level language.
- **Readability:** Lower than C; involves mnemonics and symbols.
- **Portability:** Not portable; assembly code is architecture-specific.
- **Syntax:** Directly corresponds to machine instructions, involving registers and memory addresses.
- **Memory Management:** Direct control over memory, registers, and hardware resources.
- **Execution:** Assembled into machine code; specific to the target CPU architecture.

MACHINE CODE

- **Abstraction Level:** Lowest-level language.
- **Readability:** Extremely low; consists of binary instructions directly executed by the CPU.
- **Portability:** Not portable at all; machine code is specific to the CPU architecture.
- **Syntax:** Comprises binary instructions understood by the CPU.
- **Memory Management:** Direct manipulation of hardware resources.
- **Execution:** Directly executed by the CPU.



ISA

Instruction Set Architecture

AT THE LOWEST LEVEL...

- We have a set of instructions that our microcontrollers will follow.
- Just as in our language we use syntax, semantics, grammar and other linguistic resources to understand what we say, and just as that makes it difficult to understand other languages, each ISA describes (for one or a group of microarchitectures) how it will process what we want it to do, and at the same time, it makes it different from other instruction sets.
- Understanding machine code is a little hard, but we have different levels of abstraction that we can use to program a microcontroller. However, we will now see why it is important to understand about computer architectures.

ATMEGA328P

MICROARCHITECTURE

CLOCK SPEED

MEMORY

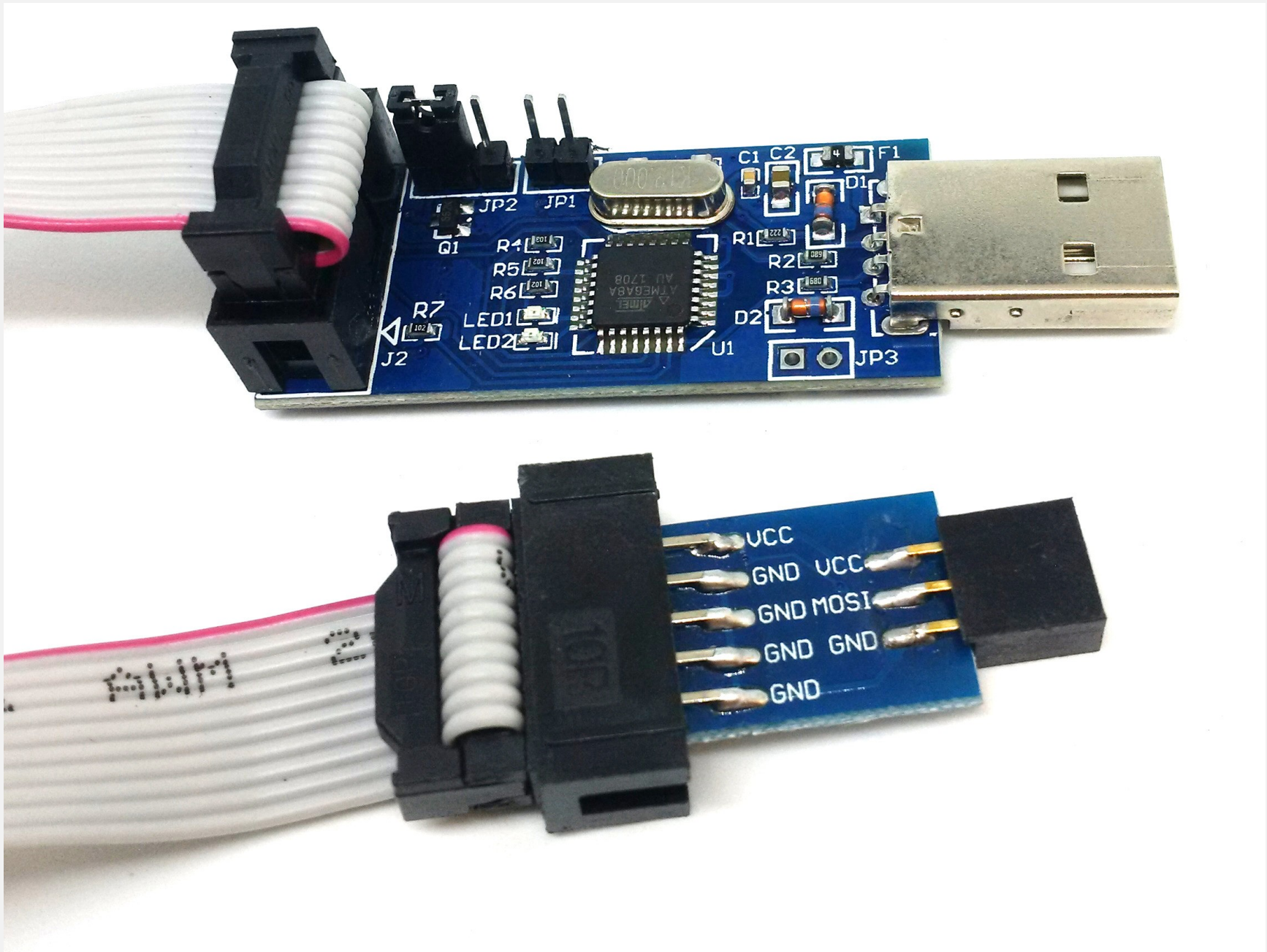
PERIPHERALS

PROGRAMMING THE ATMEGA328P

WHAT WE NEED:

LIST OF REQUIREMENTS TO PROGRAM AN ATMEGA328P

- Microchip Studio:
 - <https://www.microchip.com/en-us/tools-resources/develop/microchip-studio>
- USBASP



LIST OF REQUIREMENTS TO PROGRAM AN ATMEGA328P

- ATmega328p
- Microchip Studio or avr-gcc and avrdude:
 - <https://www.microchip.com/en-us/tools-resources/develop/microchip-studio>
- USBASP
- 16MHz Crystal Oscillator
- 2x 12-22pF Ceramic Capacitors
- 10k Resistor
- Breadboard/Protoboard, wire and jumpers
- (OPTIONAL) Buttons, resistors, LEDs and a 5V power supply

WHAT WILL WE DO WITH THAT?

KNOWLEDGE CHECK!

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!

Homeworks, whatsapp and classroom groups, missing data and stuff.

THANKS!

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