



ARDUINO

AVR Architecture (1)

Farouk MEDDAH



PLAN

➤ **AVR ARCHITECTURE**

- Au cœur de L'Arduino
- Présentation de l'AVR
- Architecture vue générale

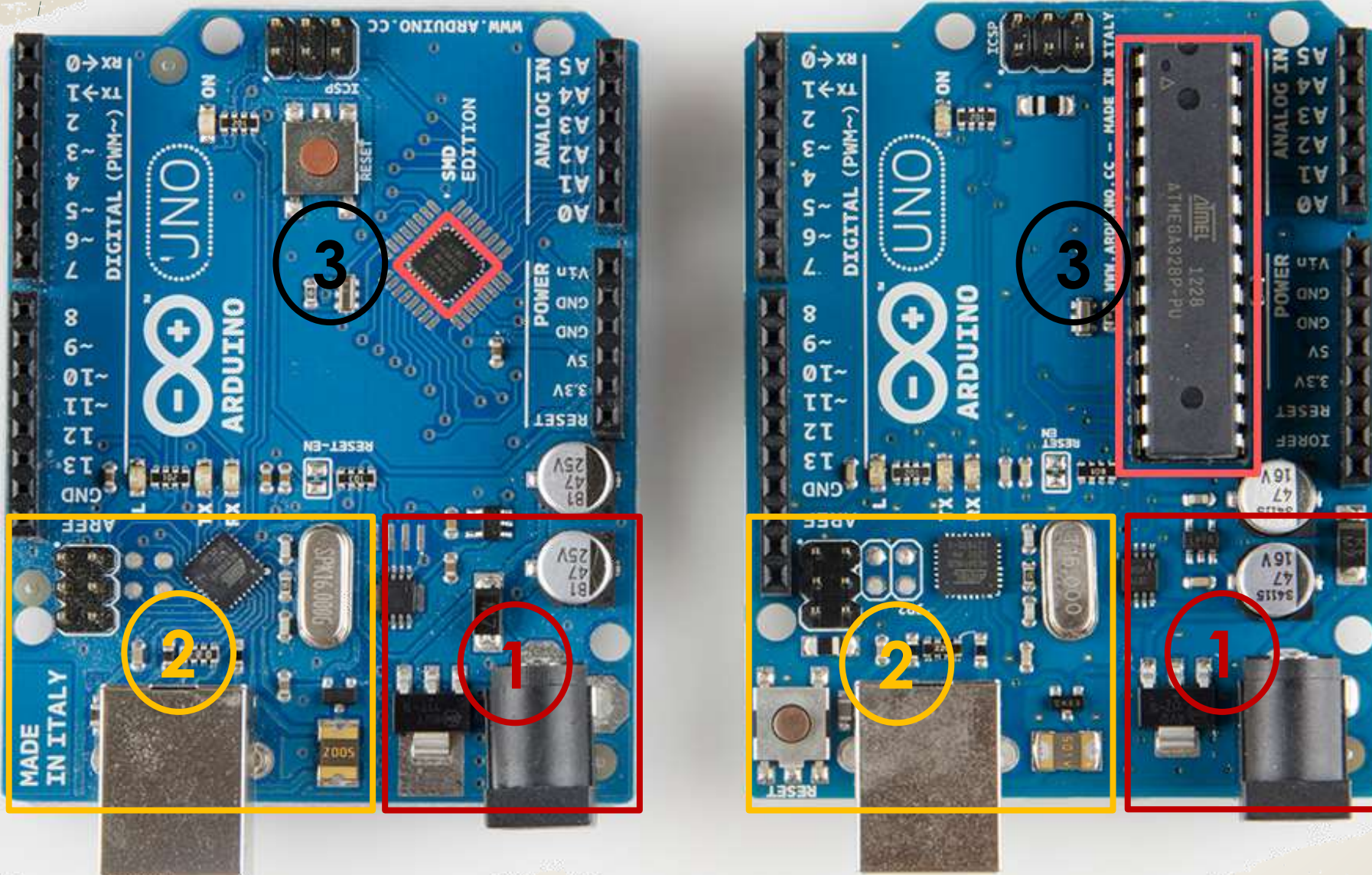
➤ **ARCHITECTURE DE L'AVR**

- Registres E/S
- Registres (généraux, spéciaux)
- Organisation mémoire
- Instructions AVR



INTRODUCTION

AU CŒUR DE L'ARDUINO



- 1-Alimentation
- 2-Communication avec le PC
- 3-Microcontrôleur AVR

AVR

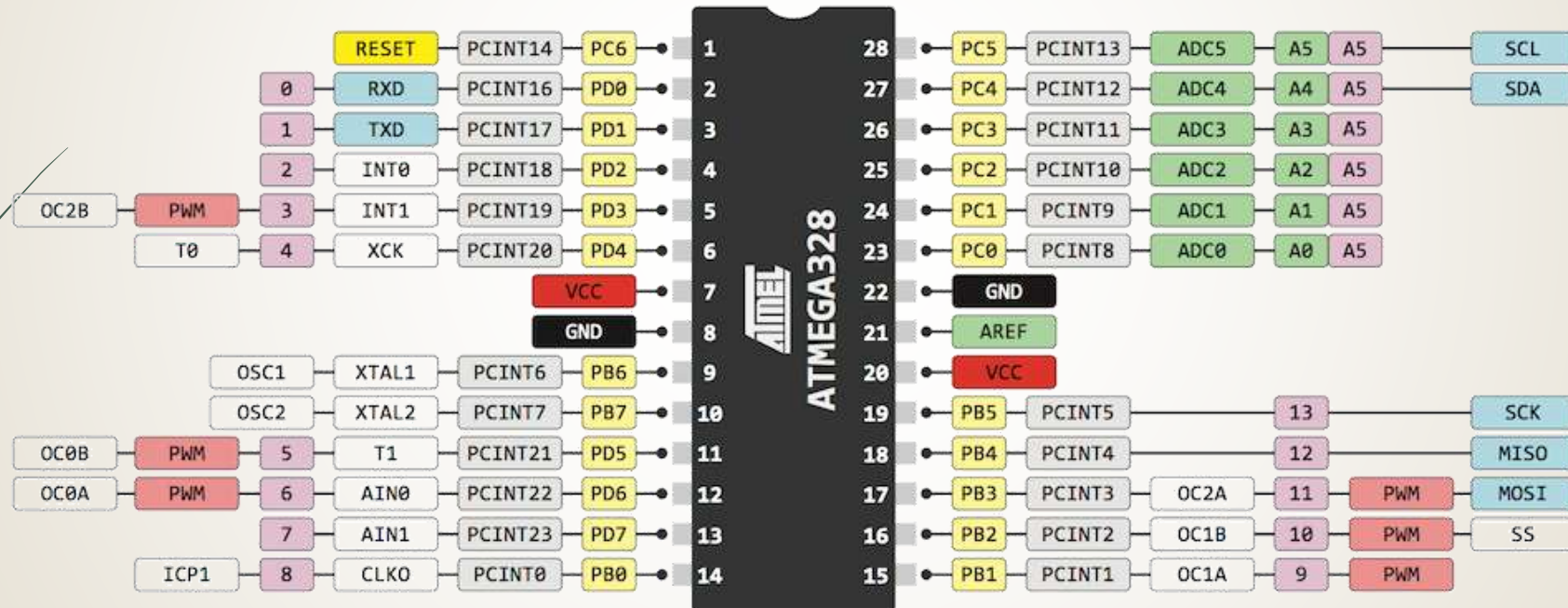
Un peu d'histoire

- Conçu par deux étudiants de NIT (**N**orwegian **I**nstitute of **T**echnology), Alf-Egil Bogen et Vegar Wollan. [**A**lf & **V**egar **R**isc]
- Acheté par la suite par Atmel en 1996.
- 8 bits CMOS, RISC, technologie Harvard.
- ISP Flash memory, read-while-write.
- Jusqu'à 20 MHz

THE
DEFINITIVE
ATMEGA328
&Arduino
PINOUT DIAGRAM

AVR ARCHITECTURE (2)

Version **Dual Inline Package**



AVR ARCHITECTURE (2)

Version Surface Mount Device

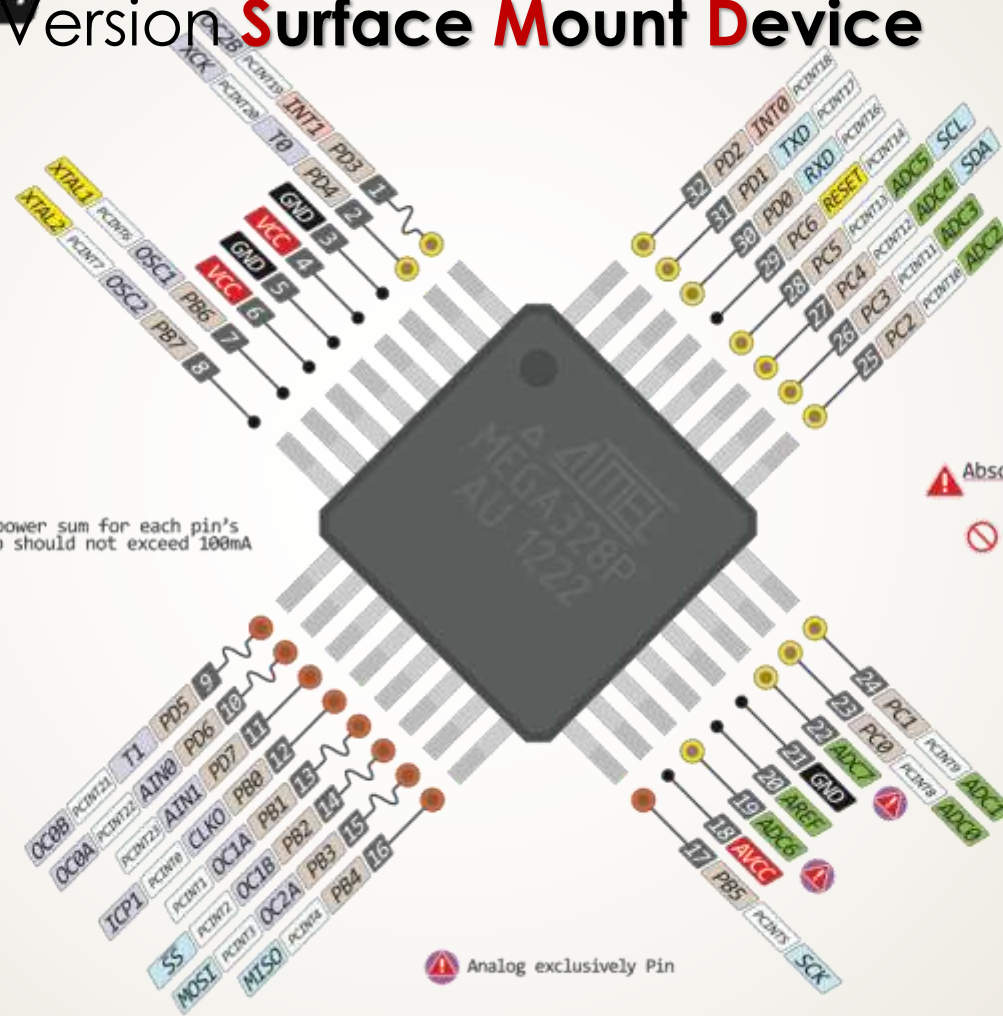
ATMEGA328 PINOUT



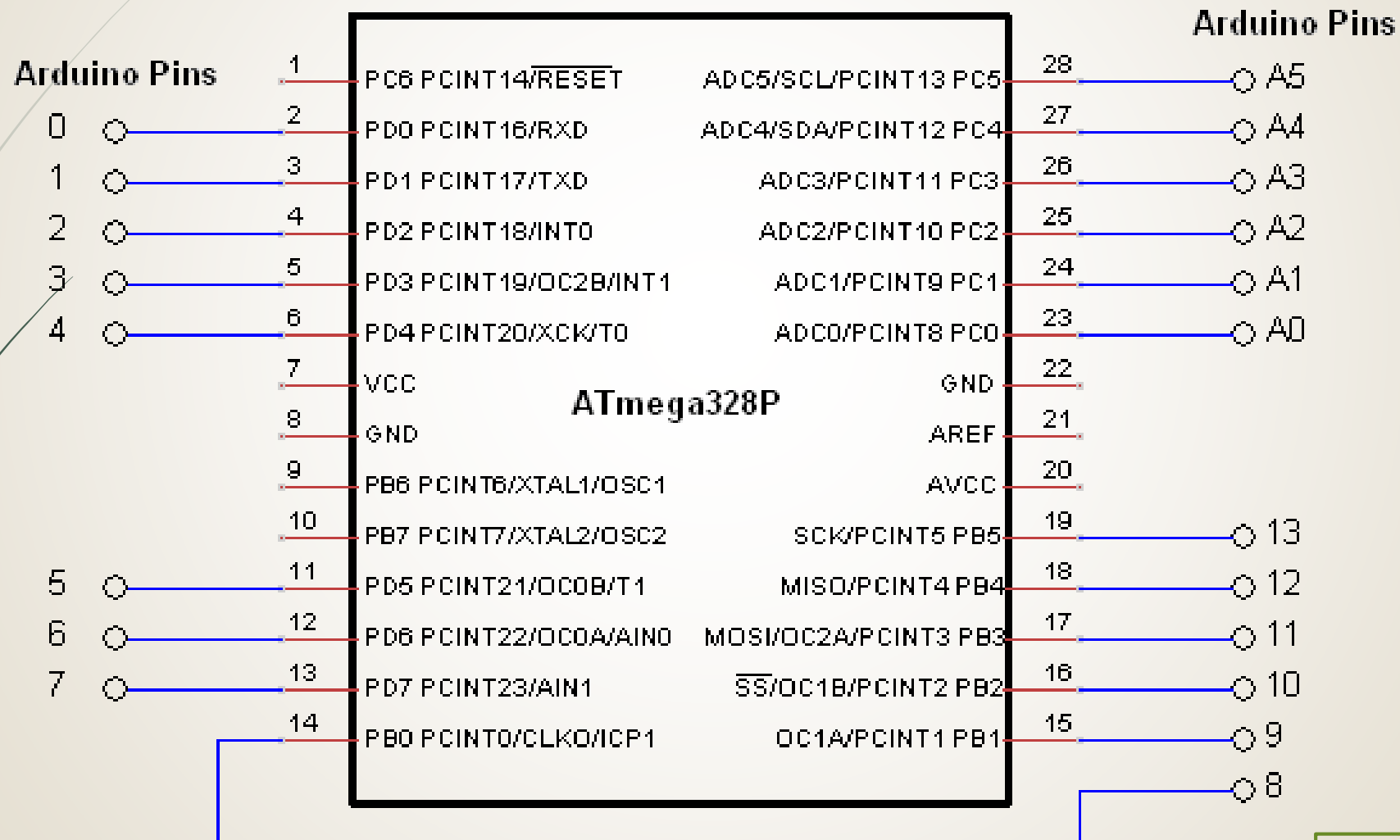
The power sum for each pin's group should not exceed 100mA

Absolute MAX per pin 40mA recommended 20mA
Absolute MAX 200mA for entire package

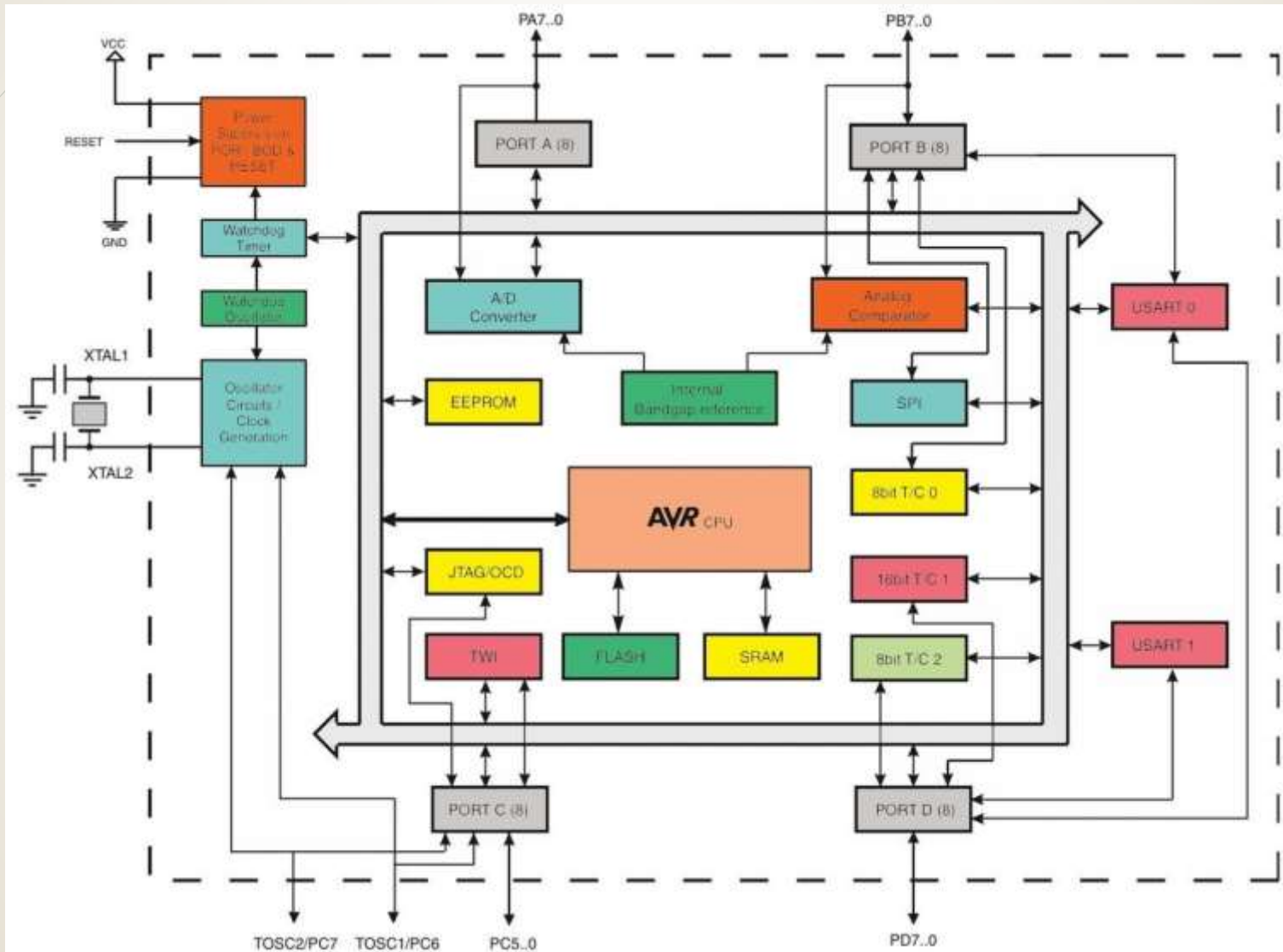
Analog exclusively Pin



AVR ARCHITECTURE (2)



AVR ARCHITECTURE (3)





REGISTRES E/S

REGISTRES E/S (1)

- Chaque port possède 8 pins :
 - Le **port B** occupe les pins 14 à 19 (Arduino: 8 à 13) [**PB6, PB7?!]**.
 - Le **port C** occupe les pins 23 à 28 (Arduino: A0 à A5) [**PC6, PC7?!]**.
 - Le **port D** occupe les pins 2 à 6 et 11 à 13 (Arduino: 0 à 7).
- Pour chaque port d'E/S il existe trois registres associés: **DDRx**, **PORTx**, et **PINx** [**x** désigne le port, exemple pour le port B: **DDRB**, **PORTB**, **PINB**].
 - **DDRx**: (**D**ata **D**irection **R**egister) direction de chaque bit du port (in/out)
 - **PORTx**: (Pin output) l'état de sortie (HIGH ou LOW)
 - **PINx**: (**P**ort **I**Nput) Lecture de l'état d'entrée

REGISTRES E/S (2)

- Pour programmer les pins du port D en entrée, il existe deux solutions:

1. Solution à base de pinMode/digitalWrite

```
for ( int i = 0 ; i <= 7 ; i++ ) {  
    pinMode(i, OUTPUT);  
    digitalWrite(i, HIGH);  
}
```

2. Solution à base de DDR/PORT :

```
DDRD = 0x00; // INPUT sur tout le port D  
PORTD = 0x00; // LOW sur tout le port D
```


REGISTRES E/S (3)

- Attention Pour l'AVR La valeur 1 veut dire SORTIE.
 - PB6, PB7 => XTAL1, XTAL2
 - PC6 => RESET
 - INT0 et INT1 s'appliquent au PD2 et PD3 successivement.
-
- Chaque port est limité à un courant total de 200 mA,
 - Le microcontrôleur peut supporter au maximum 400 mA.



REGISTRES

REGISTRES (1)

➡ 2 Types de registres

1. **General Purpose**: 32 registres de 8 bits chacun nommés R0, R1 à R31
2. **Special Purpose**: Trois registres spéciaux
 - ➡ Program Counter
 - ➡ Stack Pointer
 - ➡ Status Register.

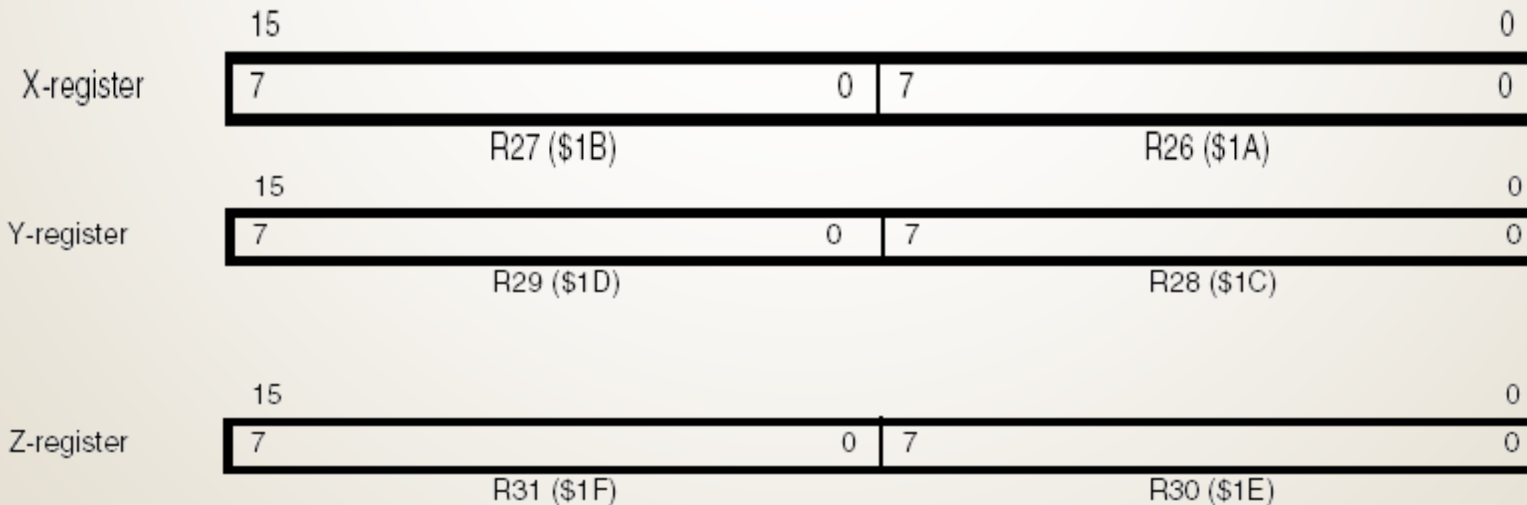
REGISTRES (2)

➤ General Purpose

- Les registres R0 à R15 ne travaillent pas avec les valeurs immédiates

➤ Registres pointeurs

- Les registres R26 à R31 ont une fonctionnalité supplémentaire en AVR, ils peuvent fonctionner en paire X(R27,R26), Y(R29,R28), Z(R31,R30).



REGISTRES (3)

➡ Registre d'état

➡ Chaque bit de ce registre a un sens.

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

I: Global Interrupt Enable/Disable Flag, SREG7

T: Transfer bit used by BLD and BST instructions, SREG6

H: Half Carry Flag, SREG5

S: For signed tests Instruction Set, SREG4

V: Two's complement overflow indicator, SREG3

N: Negative Flag, SREG2

Z: Zero Flag, SREG1

C: Carry Flag, SREG0

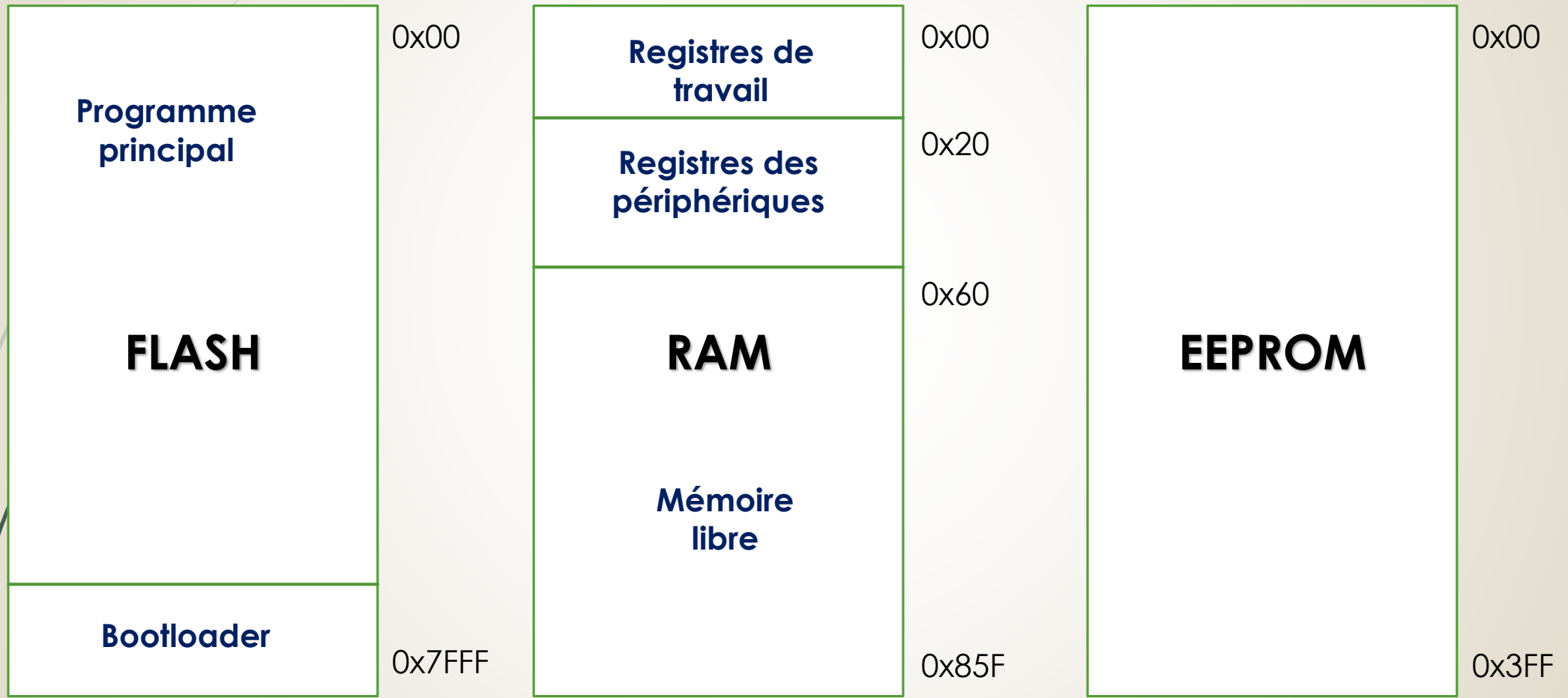


ORGANISATION MÉMOIRE

ORGANISATION MÉMOIRE (1)

- Dans l'ATMega328p Il existe trois type de mémoires:
 1. La mémoire FLASH: sert à stocker les programmes à exécuter. [32 Ko (d'où la référence 32..8)].
 2. La mémoire SRAM: (Static Read Access Memory) sert à stocker des données temporaires. [2 Ko]
 3. La mémoire EEPROM: (Electrically Erasable Programmable Read-Only Memory) permettra à vos programmes de stocker des données et de les conserver après l'arrêt de l'alimentation. [1 Ko]

ORGANISATION MÉMOIRE (2)





INSTRUCTIONS

INSTRUCTION (1)

- Jeu d'instructions de 90 à 135 instructions selon le microcontrôleur (131 pour le ATmega328p).
- Chaque instruction prend 1 cycle (sauf les instructions de saut ou d'accès mémoire).
- **Remarques:** Les codes (**1111 1xxx xxxx 1xxx**) 0xFXYZ où X=1xxb Y=xxxxb Z=1xxb sont réservés.

INSTRUCTION (2)

- L'instruction d'addition {add R14, R27} est codée 0x0EEB.

0000 1110 1110 1011

0000 11rd dddd rrrr

- Le code de l'instruction est composé en trois parties:
 - a. 6 bits **000011** (rouge) code de l'instruction add.
 - b. 5 bits **01110** (bleu) premier opérande registre 14.
 - c. 5 bits **11011** (vert) deuxième opérande registre 27 ici.



MERCI