

# **Module 2**

## **Architecture Matérielle & Énergie**

---

Le bon outil pour le bon objet connecté.

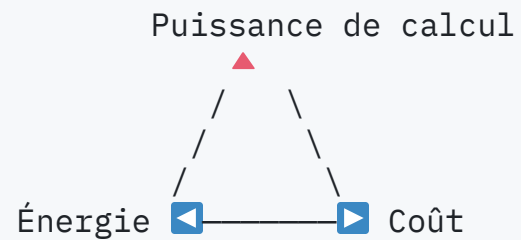
## Changelog – V0.1.0

---

- Deck 2 : MCU vs MPU, comparatif architectures, Deep Sleep, protocoles basse conso.

## Le Triangle de Contraintes IoT

Tout projet IoT embarqué est un **compromis** entre trois facteurs :



- Plus on veut de puissance → plus on consomme → plus c'est cher
- Un capteur autonome 10 ans sur pile = **sacrifice de puissance**
- Une passerelle traitant de la vidéo = **sacrifice d'autonomie**

## ● Le Microcontrôleur (MCU)

---

**Exemples :** ESP32, ESP8266, Arduino (ATmega328), STM32, SAMD21

### **Architecture :**

- CPU + RAM + Flash + Périphériques **intégrés dans une seule puce**
- Pas de système d'exploitation
- Code exécuté **en boucle** (bare metal ou RTOS léger)

### **Forces :**

- Démarrage en millisecondes
- Consommation de quelques microampères en veille
- Coût unitaire : 1 € à 10 €

## ● Le Microprocesseur (MPU)

---

**Exemples :** Raspberry Pi (BCM2835), BeagleBone, i.MX6

**Architecture :**

- CPU puissant + cache L2 + contrôleur mémoire externe
- Nécessite RAM externe, stockage (SD card), OS (Linux)
- Code exécuté dans un **environnement complet**

**Forces :**

- Traitement vidéo, IA embarquée, serveur web local
- Support de frameworks haute productivité (Python, Node.js)
- Matrice de connectivité riche

## Comparatif MCU vs MPU

---

## Tableau de décision

---

Cas d'usage	Architecture recommandée
Capteur temp/hygro toutes les 5 min	MCU (ESP32 + Deep Sleep)
Relais domotique Wi-Fi	MCU (ESP8266/ESP32)
Passerelle MQTT avec SSL	MPU (Raspberry Pi)
Caméra avec détection de mouvement IA	MPU (Raspberry Pi 4)
Tag de suivi sur batterie	MCU ultra-basse conso (SAMD21)
Dashboard local avec écran	MPU ou MCU + display

## La Problématique Énergétique

---

**Question :** Comment faire durer un capteur **3 ans sur 3 piles AA** ?

Une pile AA contient environ **2500 mAh**. Si le capteur consomme :

- En fonctionnement Wi-Fi :  $\sim 80 \text{ mA}$  → **31 heures** ❌
- En Deep Sleep :  $\sim 10 \mu\text{A}$  → **28 ans** ✅

**La clé :** minimiser le temps de fonctionnement actif.

## zzZ Le cycle Deep Sleep

---

## Protocoles réseau basse consommation

---

Protocole	Portée	Conso	Débit	Usage typique
Wi-Fi	~50m	⚡⚡⚡ Élevée	100 Mbps	Passerelles, caméras
BLE 5.0	~100m	⚡ Très basse	2 Mbps	Capteurs portables
LoRa	~10 km	⚡ Très basse	50 kbps	Capteurs ruraux, Smart City
Zigbee	~100m	⚡ Basse	250 kbps	Domotique mesh
NB-IoT	Cellulaire	⚡ Basse	100 kbps	Compteurs, tracking

## L'ESP32 : le champion du rapport qualité/prix

---

### Spécifications clés :


- CPU : Xtensa dual-core 240 MHz
- RAM : 520 KB SRAM
- Flash : 4 MB (externe)
- Wi-Fi 802.11 b/g/n **intégré**
- Bluetooth 4.2/BLE **intégré**
- 34 GPIO (ADC, DAC, I2C, SPI, UART, PWM...)
- **Deep Sleep : 10  $\mu$ A**
- Prix : ~4-8 €

## Calcul d'autonomie : l'ESP32 en capteur

---

**Scénario** : mesure toutes les 10 minutes + envoi MQTT + Deep Sleep

Phase	Durée	Courant	Énergie
Deep Sleep	598 s	10 $\mu$ A	1.66 $\mu$ Ah
Réveil + Wi-Fi	1 s	160 mA	44.4 $\mu$ Ah
Mesure capteur	0.1 s	40 mA	1.1 $\mu$ Ah
<b>Cycle total</b>	<b>600 s</b>	<b>Moy. ~0.27 mA</b>	<b>~47 <math>\mu</math>Ah</b>

Batterie 18650 (3000 mAh) → **~250 jours**  $\approx$  8 mois 

## 🧩 Périphériques typiques d'un MCU

```

+----- ESP32 -----+
|
GPIO --> | GPIO (34 broches) | --> LED, Relais, Moteur
ADC --> | ADC 12-bit (18 canaux) | --> Capteurs analogiques
        | DAC 8-bit (2 canaux) | --> Signal audio, tension
I2C <-> | I2C (SDA/SCL) | <-> OLED, BMP280, MPU6050
SPI <-> | SPI (MOSI/MISO/CLK/CS) | <-> SD Card, NRF24
UART <-> | UART (3 bus) | <-> GPS, Modem GSM
PWM --> | PWM (tous GPIO) | --> Servo, LED dimmable
+-----+

```

## Synthèse du Module 2

---

Concept	Retenir
MCU	Pas d'OS, ultra-basse conso, tâches spécifiques
MPU	Linux, haute puissance, consommation élevée
Deep Sleep	Clé de l'autonomie sur batterie
ESP32	MCU polyvalent Wi-Fi+BLE, 10 $\mu$ A en veille
Protocoles	Wi-Fi = gourmand, BLE/LoRa = économes

**Prochain module** → TP 1 — Premiers programmes avec PlatformIO

## Questions ?

---

Module 2 – Architecture Matérielle & Énergie  
*Réda BOUREBABA & Sébastien Antonico*