GUIDE GRATUIT • ÉLECTRONIQUE

Introduction aux microcontrôleurs

Arduino, ATtiny, ESP32 :
lequel choisir pour débuter ?

• • •

📐 FONDAMENTAUX

🧰 MATÉRIEL

⚡ PRATIQUE

BLOG DAUTEK

Introduction aux microcontrôleurs : Arduino, ATtiny ou ESP32, lequel choisir ?

Comprendre ce qu'est un microcontrôleur avant de programmer votre premier projet

📖 Chapitre 12/14 | 🗂️ Passer à la pratique | ⏱️ ~5 min de lecture

🎯 Vous avez vu au premier chapitre qu'un microcontrôleur n'est rien d'autre qu'un circuit intégré capable d'exécuter un programme. Il est temps de rentrer dans le détail : Arduino, ATtiny, ESP32, ces noms reviennent partout dès qu'on parle d'électronique programmée, et pourtant ils désignent des choses très différentes.

⚡ Qu'est-ce qu'un microcontrôleur, concrètement ?

Un microcontrôleur est une puce qui regroupe, dans un seul boîtier, un processeur, de la mémoire, et des broches d'entrée/sortie capables de lire un capteur ou de piloter une LED, un relais, un moteur. On y installe un programme (le "firmware") qui définit son comportement : c'est ce qui transforme un simple montage électronique passif en objet réellement intelligent, capable de réagir à son environnement.

🔌 Arduino, ATtiny, ESP32 : trois familles, trois usages

⚡ Arduino (Uno et compatibles)

La référence pour apprendre et prototyper

Une carte de développement complète, avec microcontrôleur, port USB, et broches facilement accessibles. Pensée dès le départ pour l'apprentissage, avec un écosystème de tutoriels immense.

✅ CE QU'IL FAUT RETENIR

- Écosystème et documentation les plus riches du marché
- Programmation simple via l'IDE Arduino, entièrement gratuit
- Idéal pour prototyper avant de miniaturiser

⚠️ POINTS DE VIGILANCE

- Plus volumineuse qu'une puce nue, moins adaptée à un objet fini compact
- Prix plus élevé qu'un microcontrôleur seul pour un usage en série
- Consommation électrique plus importante qu'un ATtiny en veille

📍 **À privilégier pour : l'apprentissage, le prototypage, et les projets où la taille importe peu.**

🔌 ATtiny (ATtiny13, 85, 1614…)

Le microcontrôleur minimaliste et économe

Une puce nue, sans carte de développement autour, pensée pour être intégrée directement dans un circuit imprimé final. Moins de broches et de mémoire qu'un Arduino Uno, mais un encombrement et une consommation bien plus faibles.

✅ CE QU'IL FAUT RETENIR

- Très compact, parfait pour un objet fini miniaturisé
- Consommation électrique très faible, idéal sur batterie
- Coût unitaire très bas pour une production en petite série

⚠️ POINTS DE VIGILANCE

- Moins de broches et de mémoire disponibles qu'un Arduino Uno
- Nécessite un programmateur externe (pas de port USB intégré)
- Communauté et documentation plus restreintes qu'Arduino

📍 **À privilégier pour : un projet fini, compact et autonome sur batterie, une fois le prototype validé sur Arduino.**

ESP32 (et ESP8266)

Le microcontrôleur connecté, taillé pour l'IoT

Une puce intégrant en plus un module Wi-Fi (et souvent Bluetooth), pensée pour les objets connectés. C'est le choix naturel dès qu'un projet doit communiquer avec un réseau domestique ou une box domotique.

✅ CE QU'IL FAUT RETENIR

- Wi-Fi et Bluetooth intégrés, sans module supplémentaire
- Puissance de calcul largement supérieure à un Arduino Uno classique
- Compatible avec Home Assistant et la plupart des box domotique via des firmwares dédiés

⚠️ POINTS DE VIGILANCE

- Consommation électrique plus élevée qu'un ATtiny, moins adapté au tout petit budget batterie
- Configuration réseau (Wi-Fi) à gérer en plus du reste du programme
- Légèrement plus complexe à appréhender pour un tout premier projet

📍 **À privilégier pour : tout projet domotique ou connecté ayant besoin de communiquer en Wi-Fi.**

Critère	Arduino Uno	ATtiny85	ESP32
Idéal pour	Apprentissage, prototypage	Objet fini compact	Objet connecté Wi-Fi
Connectivité	USB uniquement	Aucune (programmateur externe)	Wi-Fi + Bluetooth
Mémoire programme (flash)	32 Ko	8 Ko	4 Mo ou plus
Broches d'entrée/sortie	14 numériques + 6 analogiques	5 (dont 1 souvent réservée)	Plus de 30
Consommation en veille	≈ 15-40 mA	Quelques µA à quelques mA	≈ 20-100 mA (Wi-Fi actif)
Prix indicatif	≈ 4 à 10€ (compatible)	≈ 1 à 2€	≈ 3 à 8€
Difficulté pour débuter	Facile	Intermédiaire	Intermédiaire

📌 **À propos de la mémoire :** 8 Ko peuvent sembler dérisoires face aux 32 Ko d'un Arduino Uno, mais suffisent largement pour un programme simple (lire un capteur, piloter une LED ou un relais selon une condition). Les projets plus élaborés, avec plusieurs capteurs et une logique complexe, s'orientent naturellement vers un Arduino ou un ESP32.

💡 **Le saviez-vous ?** L'ATtiny et l'Arduino Uno partagent souvent la même famille de microcontrôleur à la base (Atmel/Microchip AVR) : on peut d'ailleurs programmer un ATtiny directement depuis l'IDE Arduino, une fois quelques réglages effectués.

🧩 Comment programmer un ATtiny depuis l'IDE Arduino

Contrairement à un Arduino, un ATtiny n'a pas de port USB intégré : il faut utiliser un programmateur externe pour y transférer un programme. La méthode la plus accessible pour un débutant consiste à utiliser une carte Arduino Uno elle-même comme programmateur ("Arduino as ISP"), en installant simplement un petit croquis dédié fourni avec l'IDE, puis en ajoutant la définition de carte "ATtiny" via le gestionnaire de cartes. Une fois ces deux étapes réalisées, on programme un ATtiny presque exactement comme un Arduino classique, avec la même interface.

📌 **À retenir :** l'Arduino Uno n'est jamais "perdu" une fois que vous passez à l'ATtiny : il redevient votre outil de prototypage et de programmation pour tous vos futurs projets, quelle que soit la puce finale choisie.

📐 Mon avis : commencez toujours par Arduino, même pour un projet final en ATtiny

Le réflexe le plus efficace, même pour un projet finalement destiné à un ATtiny, est de d'abord tout valider sur une carte Arduino Uno classique : la programmation y est plus simple, le débogage plus rapide, et l'écosystème de documentation bien plus large en cas de blocage. Une fois le projet stable, la migration vers un ATtiny (pour la miniaturisation) ou un ESP32 (pour la connectivité) devient un choix éclairé, plutôt qu'un pari.

✏️ **Petit défi :** pour le projet que vous avez en tête, listez ses contraintes réelles (taille, autonomie, besoin de Wi-Fi) et déterminez, à l'aide du tableau ci-dessus, quelle famille de microcontrôleur conviendrait le mieux.

📥 GUIDE GRATUIT

Gardez ce comparatif sous la main

La version PDF reprend les trois fiches et le tableau comparatif Arduino / ATtiny / ESP32.

🛠 Questions fréquentes

1 Quel microcontrôleur choisir pour un tout premier projet ?

Une carte Arduino Uno (ou compatible) reste le choix le plus sûr pour apprendre sans se heurter à des difficultés annexes.

2 Peut-on utiliser un ESP32 pour un projet non connecté ?

Oui, rien n'empêche d'utiliser un ESP32 sans jamais activer son Wi-Fi, mais cela reste un choix peu économique si la connectivité n'est pas nécessaire.

3 Faut-il apprendre un langage de programmation spécifique ?

Le langage utilisé (proche du C/C++) est commun à la quasi-totalité de ces cartes via l'IDE Arduino, ce qui facilite grandement le passage de l'une à l'autre.

4 Un Arduino Uno peut-il vraiment servir à programmer un ATtiny ?

Oui, c'est la méthode "Arduino as ISP", très répandue chez les amateurs : l'Arduino agit comme simple intermédiaire de programmation, sans que son propre programme ne soit affecté.

💡 Et maintenant ?

Théorie terminée : place au tout dernier chapitre de cette formation, où vous allez programmer votre premier projet avec un microcontrôleur, de bout en bout.

© Blog Dautek — blog.dautek.fr — Guide gratuit à usage personnel, non destiné à la revente.