

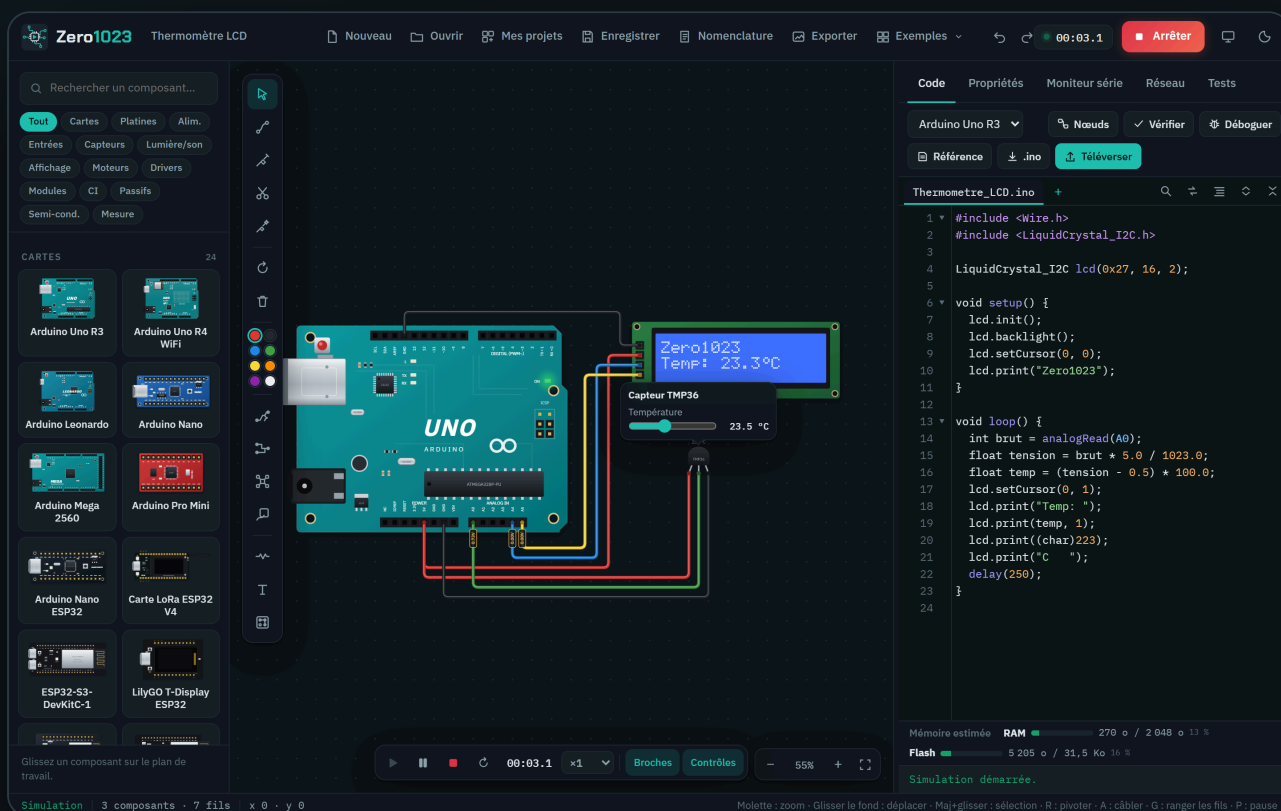


Zero1023

Mode d'emploi
Édition octobre 2026

De 0 à 1023, donnez vie à vos circuits.

Concevoir, programmer et simuler des montages Arduino, ESP32 et ESP8266 — puis les envoyer sur une vraie carte.



Câbler

Arduino, ESP32, ESP8266, Pico, STM32... et leurs composants

Programmer

en langage Arduino ou en reliant des blocs

Simuler

tensions, courants, mémoire, Wi-Fi et radio

Téléverser

sur la vraie carte, par USB

Zero1023 en bref

Zero1023 est un atelier d'électronique complet qui s'ouvre dans le navigateur. On y dessine un montage, on le programme, on le regarde fonctionner comme en vrai... puis on l'envoie sur la vraie carte. Sans installation et sans rien griller.

0 → 1023

ANALOGREAD()

Pourquoi ce nom ? Sur une carte Arduino, une entrée analogique renvoie une valeur entre **0** et **1023**. Entre les deux, il y a toutes les nuances : la lumière qui baisse, la température qui monte, le bouton qu'on tourne. C'est là que vivent les capteurs, les moteurs et les idées. Zero1023 couvre tout ce spectre, du premier clignotement de LED au réseau d'objets connectés.

Imaginer et câbler

Des cartes Arduino, ESP32, ESP8266, Raspberry Pi Pico, STM32 ou micro:bit, et une large bibliothèque de composants : capteurs, écrans (jusqu'à l'écran rond), moteurs, modules radio. Platines d'essai, plaques à souder, câblage automatique et vue schéma normalisée.

Simuler comme en vrai

Tensions, courants, mémoire : le simulateur calcule ce que ferait le circuit et signale les erreurs qui abîmeraient les composants (LED sans résistance, court-circuit, module 3,3 V en 5 V...).

Vérifier et comprendre

Débogueur pas à pas, traceur de courbes, scénarios de test rejouables, plusieurs cartes qui dialoguent en même temps : on voit ce qui se passe, et pourquoi.

Programmer à votre façon

En langage Arduino, avec un éditeur qui complète et vérifie le code ; ou sans écrire une ligne, en reliant des blocs, ou en partant d'un modèle prêt à l'emploi. Le code généré est un vrai programme, lisible et téléversable.

Connecter

Wi-Fi et page web de pilotage, Bluetooth, MQTT, radio directe ESP-NOW, LoRa longue portée, SMS et appels GSM, notifications sur le téléphone. Un bloc suffit à chiffrer les échanges, et un espion radio permet de jouer le pirate.

Passer au réel

Liste des composants à acheter, schéma et images à partager, puis téléversement par USB sur la vraie carte, avec pages web en https sur le téléphone. Le mode miroir fait même piloter la simulation par les vrais boutons, ou par le téléphone en Wi-Fi.

■ Pour qui ?

Pour apprendre et enseigner l'électronique et la programmation, du collège aux études supérieures ; pour les makers qui veulent essayer une idée avant d'acheter les composants ; pour tous ceux qui préfèrent griller une LED virtuelle qu'une vraie. Une bibliothèque d'exemples prêts à ouvrir, classés du plus simple au plus avancé, sert de point de départ. L'interface, les exemples et les aides existent en français et en anglais.

■ Le parcours type

1

Ouvrir

un exemple, ou partir d'un plan vide

2

Câbler

les composants sur la carte

3

Programmer

en code ou en blocs

4

Simuler

tester et déboguer

5

Téléverser

sur la vraie carte

Sommaire

1 Bienvenue

Ce que fait Zero1023, comment l'ouvrir, version découverte et Pro, les cartes

2 L'interface

Les zones de l'écran en un coup d'œil

3 Construire un montage

Composants, fils, platines, plaques à souder, étiquettes

4 Programmer en code

Éditeur, autocomplétion, plusieurs fichiers, vérification

5 Programmer avec des blocs

L'éditeur de nœuds, modèles, variables, texte, blocs réutilisables, écran rond

6 Lancer la simulation

Interagir, vitesse, mémoire, débogueur, tests automatiques

7 Plusieurs cartes ensemble

Série, I²C, Wi-Fi, radio directe ESP-NOW, LoRa, MQTT

8 Sans fil et page web

Onglet Réseau, page web, caméra du téléphone, LoRa, sécurité radio, notifications, veille, ESP32-CAM, GSM

9 Schéma, nomenclature et images

Documenter et partager un montage

10 Projets, versions et exemples

Enregistrer, retrouver, revenir en arrière

11 Téléverser sur une vraie carte

Installer le pont, téléverser, pages web en https sur le téléphone, mode miroir et miroir Wi-Fi

12 Personnaliser l'affichage

Thèmes, styles, plein écran, multi-écran, langue, compagnon

13 Raccourcis clavier

Tous les raccourcis sur une page

14 Questions fréquentes et limites

Dépannage et ce que le simulateur ne fait pas

1 Bienvenue

Ce chapitre fait le tour de ce que propose Zero1023, montre comment l'ouvrir et présente les cartes disponibles.

■ Ce que vous pouvez faire

Câbler

Poser cartes, capteurs, moteurs et écrans, les relier par des fils, sur platine d'essai ou plaque à souder.

Programmer

En langage Arduino (C++) avec un éditeur qui aide à écrire, ou en assemblant des blocs.

Simuler et vérifier

LED, moteurs, écrans et sons réagissent en direct. Débogueur pas à pas et scénarios de test rejouables.

Connecter

Wi-Fi, page web, Bluetooth, MQTT, radio directe ESP-NOW, LoRa longue portée, notifications sur le téléphone, réseau mobile GSM, liaisons série et I²C. Échanges radio chiffrés en un bloc.

Documenter

Vue schéma normalisée, liste de courses (nomenclature), export en image.

Téléverser

Compiler et envoyer le programme sur une carte branchée en USB, sans l'IDE Arduino.

■ Ouvrir Zero1023

- 1 **En ligne** : ouvrez le site de Zero1023 et cliquez sur [Ouvrir Zero1023](#) . Aucune inscription ni installation. **Chrome** ou **Edge** sont conseillés (ils savent aussi lire une carte branchée en USB).
- 2 **En fichier** : si vous disposez du fichier **Zero1023.html**, double-cliquez dessus. Il fonctionne **sans Internet** et peut être copié sur une clé USB.
- 3 Pour démarrer vite, ouvrez le menu [Exemples](#) et choisissez un montage, puis cliquez sur [Démarrer la simulation](#) .

Vos données restent chez vous

Les projets sont enregistrés dans des fichiers `.json` sur votre ordinateur (**Ctrl** + **S**). La liste « Mes projets » et l'historique des versions sont gardés par le navigateur, sur ce poste uniquement.

■ Version découverte et Zero1023 Pro

La **version découverte**, en ligne, est gratuite et sans compte. Elle comprend les cartes les plus répandues (Arduino Uno et Nano, ESP32, ESP8266, Raspberry Pi Pico...) et tous les composants d'un kit débutant : LED, boutons, capteurs, écrans LCD et OLED, matrice LED, NeoPixel, servo, moteur pas à pas, télécommande IR, horloge RTC... Programmation en code et en blocs, simulation, moniteur série, traceur, débogueur, Wi-Fi, page web et Bluetooth sont inclus.

Zero1023 Pro ajoute le matériel spécialisé (Mega, ESP32-CAM, STM32, GPS, RFID, écrans TFT, rond et e-paper, moteurs NEMA, oscilloscope, plaques à souder...), les projets à **plusieurs cartes**, la **radio avancée** (MQTT, ESP-NOW, LoRa, GSM, notifications, canal sécurisé), les **scénarios de test**, le **téléversement** sur la vraie carte avec le

mode miroir, et les **exports haute définition**. Dans la version découverte, ces éléments portent la mention **Pro** ; ils restent visibles pour que vous puissiez les découvrir.

La liste évolue

Le contenu de chaque version s’enrichit au fil des mises à jour : la bibliothèque de l’application fait foi.

 Les cartes disponibles

FAMILLE	CARTES	PARTICULARITÉS
Arduino AVR	Uno R3, Leonardo, Nano, Mega 2560, Pro Mini, ATtiny85	5 V, entrées analogiques 0–1023
Arduino Uno R4 WiFi	Uno R4 WiFi	Format Uno, Wi-Fi, matrice de LED 12 × 8 intégrée
ESP32	ESP32 DevKit V1, ESP32 DevKitC, ESP32 Mini, Keystudio ESP32 PLUS, Arduino Nano ESP32, XIAO ESP32-C3, ESP32-C3 SuperMini, ESP32-S3-DevKitC-1, LilyGO T-Display, ESP32-CAM, carte LoRa ESP32	3,3 V, Wi-Fi, Bluetooth, analogique 0–4095, veille profonde ; écran couleur sur la T-Display, caméra sur l’ESP32-CAM, radio LoRa et écran OLED sur la carte LoRa
ESP8266	ESP-01, Wemos D1 mini, NodeMCU	3,3 V, Wi-Fi
Autres	Raspberry Pi Pico (RP2040), STM32 Blue Pill, BBC micro:bit V2	3,3 V ; matrice 5 × 5 et boutons A / B sur la micro:bit

Changer de carte

Pour passer d’une carte à une autre sans refaire le montage : propriétés de la carte, section **Changer de carte**. Les broches et le code sont adaptés automatiquement. Le bloc **LED de la carte** pilote la petite LED soudée sur la carte, quelle qu’elle soit (y compris celles qui s’allument à l’état LOW).

2 L'interface

Tout se passe sur un seul écran : la bibliothèque à gauche, le plan de travail au centre, le code et les outils de simulation à droite.

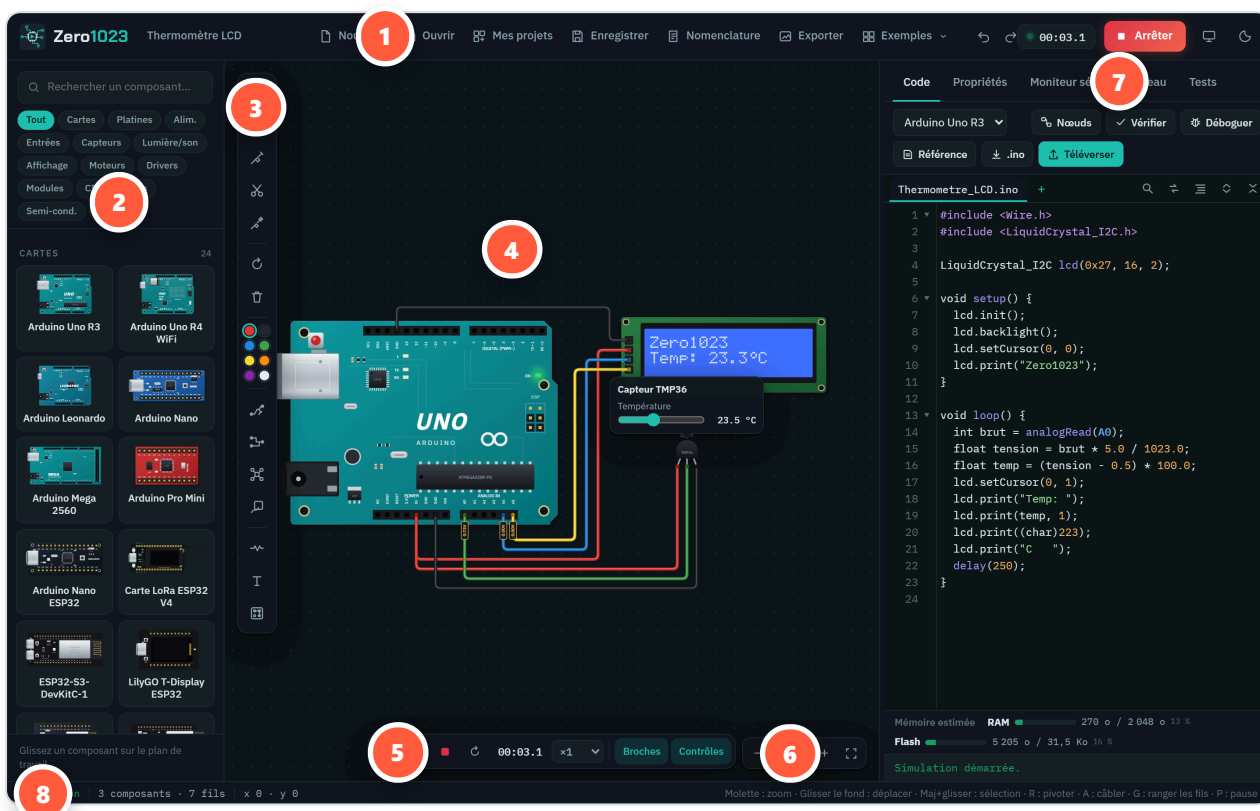


Figure 2.1 — L'exemple « Thermomètre LCD » pendant la simulation.

- 1 **Barre du haut** : nouveau, ouvrir, Mes projets, enregistrer, nomenclature, exporter, exemples, annuler, démarrer la simulation, affichage, thème clair / sombre.
- 2 **Bibliothèque** : cartes et composants à glisser sur le plan, avec recherche et filtres par catégorie.
- 3 **Outils** : sélection, fil, fer à souder, coupe de piste, pivoter, supprimer, couleur des fils, câblage automatique, rangement, schéma, étiquette.
- 4 **Plan de travail** : votre montage. Molette pour zoomer, glisser le fond pour se déplacer.
- 5 **Barre de simulation** : démarrer, pause, arrêter, redémarrer, chronomètre, vitesse ($\times 0,1$ à $\times 10$), étiquettes **Broches** et curseurs **Contrôles**.
- 6 **Zoom** : dézoomer, niveau, zoomer, tout afficher (F).
- 7 **Panneau de droite** : onglets **Code**, **Propriétés**, **Moniteur série**, **Réseau**, **Tests** ; jauge **Mémoire estimée** sous le code. Largeur réglable.
- 8 **Barre d'état** : mode (édition ou simulation), nombre de composants et de fils, position du pointeur, rappel des raccourcis.

Se déplacer sur le plan

Molette : zoom. Glisser le fond (ou maintenir **Espace**) : déplacer la vue. **Maj** + glisser : sélection rectangulaire. **F** : tout afficher.

3 Construire un montage

Un montage se compose de cartes, de composants et de fils. Tout se manipule à la souris, avec quelques raccourcis pour aller plus vite.



Figure 3.1 — La bibliothèque.

Ajouter des composants

- 1 Cherchez un composant par son nom (« servo », « DHT », « relais »...) ou filtrez par catégorie : Cartes, Platines, Alimentation, Entrées, Capteurs, Lumière et son, Affichage, Moteurs, Drivers, Modules, Circuits intégrés, Passifs, Semi-conducteurs, Mesure.
- 2 Glissez la vignette sur le plan de travail.
- 3 Déplacez le composant en le faisant glisser. Il s'enfiche tout seul dans une platine d'essai quand ses pattes tombent sur des trous.

Agir sur la sélection

- **R** pivoter d'un quart de tour
- **Ctrl** + **D** dupliquer
- **Suppr** supprimer
- **Ctrl** + **A** tout sélectionner
- **Ctrl** + **Z** / **Ctrl** + **Y** annuler / rétablir

■ Régler un composant : l'onglet Propriétés

Cliquez sur un composant : l'onglet **Propriétés** affiche ses réglages (couleur d'une LED, valeur d'une résistance, tension d'une alimentation, température d'un capteur...), ses broches et des boutons d'action :

Câbler automatiquement, **Ranger ses fils**, **Pivoter**, **Dupliquer**, **Supprimer**. Double-cliquer sur une carte ouvre directement son code.

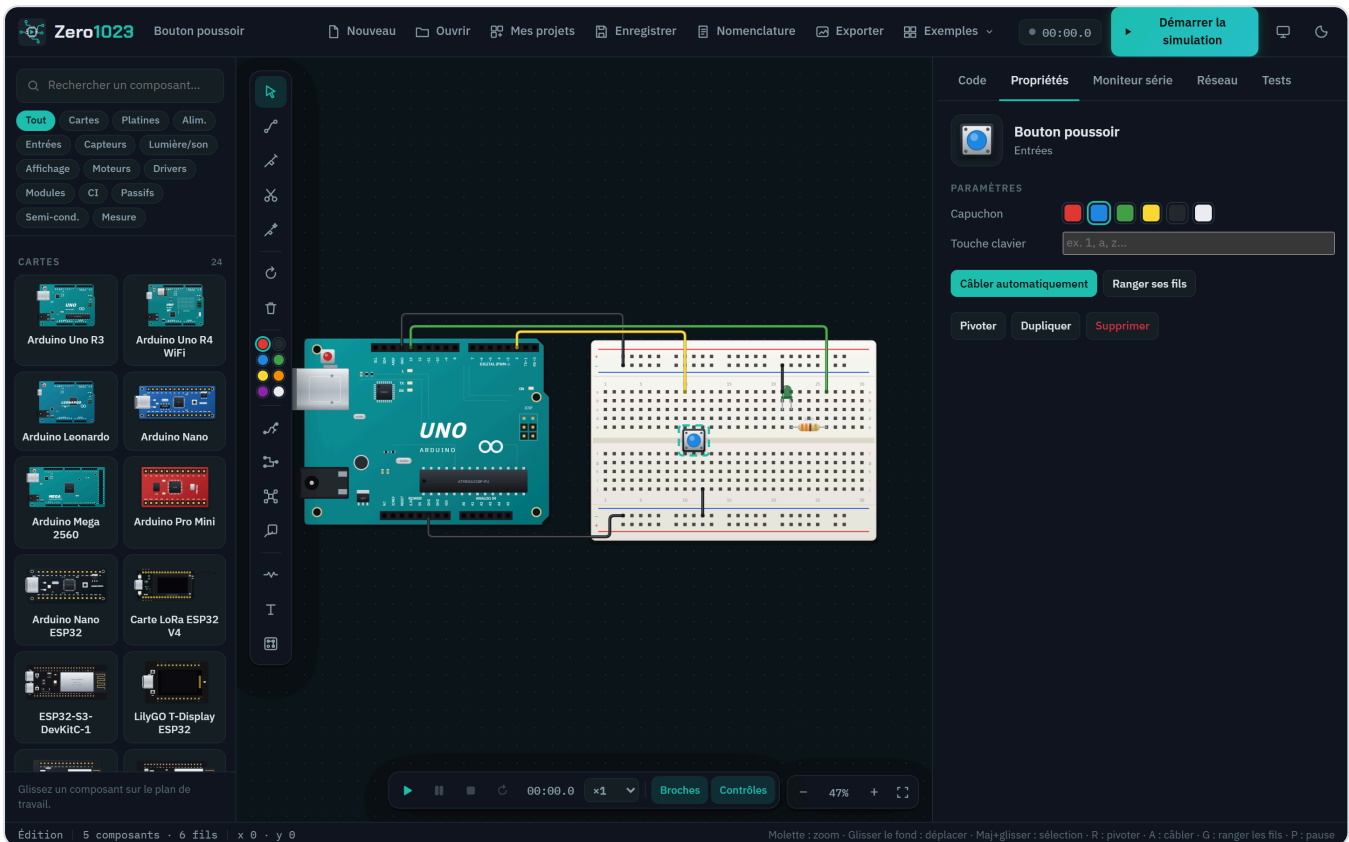


Figure 3.2 — Un bouton sélectionné et ses propriétés : couleur du capuchon et touche clavier associée.

■ Tirer des fils

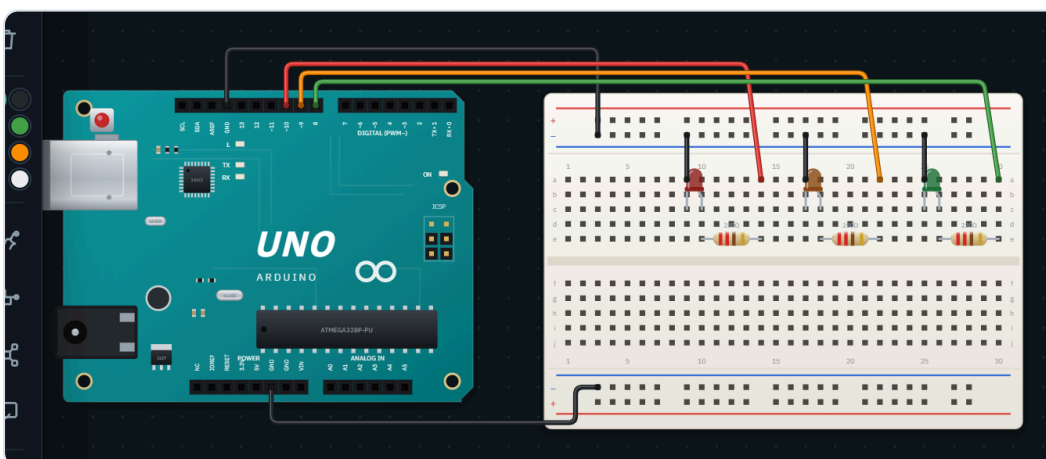


Figure 3.3 — Fils de couleur entre la carte et la platine d'essai.

- 1 Choisissez l'outil **Fil** (**W**) et une couleur dans la palette d'outils.
- 2 Cliquez sur une broche, puis sur la broche d'arrivée. Des clics intermédiaires sur le plan créent des coudes.
- 3 **Échap** abandonne le fil en cours.

Un **double-clic sur un fil** ajoute un point de passage ; un double-clic sur un point le retire.

Laisser Zero1023 câbler

- **Câblage automatique** (A) : relie le composant sélectionné, ou tous les composants non câblés, aux bonnes broches de la carte (alimentation, masse, I²C, SPI...).
- **Ranger les fils** (G) : trace des chemins propres à angles droits.
- **Organiser le montage** (O) : range les composants autour de la carte et retrace les fils.
- **Fils en arrière-plan** (B) : fait passer les fils sous les composants.

Capteurs de mesure et modules récents

COMPOSANT	CE QU'IL MESURE OU FAIT	PENDANT LA SIMULATION
Balance (capteur de force + HX711)	Masse posée, capteur de 1, 5, 20 ou 50 kg	Réglez la masse posée
Wattmètre INA219	Tension, courant et puissance d'un circuit, en I ² C : insérez-le en série dans l'alimentation à mesurer (VIN+ → VIN-)	La mesure suit le vrai courant du montage
Distance laser VL53L0X	Distance d'un obstacle, de 3 cm à 2 m, au millimètre (I ² C)	Réglez la distance de l'obstacle
Lecteur d'empreintes digitales	Reconnaît une empreinte enregistrée (AS608 / R307, liaison série)	Choisissez le doigt posé sur la vitre
Module radio LoRa	Radio longue portée SX1276 / SX1278 (433, 868 ou 915 MHz), en SPI	Réglez la distance et l'environnement (chapitre 8)
Écran rond GC9A01	Écran couleur rond de 1,28" (240 × 240 points), en SPI : jauges, cadrans, montres	L'écran affiche le dessin du programme (chapitre 5)

Platines d'essai

Trois tailles sont proposées : 830, 400 et 170 points. Comme en vrai, les cinq trous d'une colonne (a–e, puis f–j) sont reliés entre eux, et les rails + et – courent sur toute la longueur. Survoler une broche ou un trou affiche son nom et, pendant la simulation, sa tension.

Plaques à souder

Pour passer au montage définitif, posez les composants sur une **plaque pastillée** ou une **plaque à bandes** :

- **Fer à souder** (S) : trace des pistes d'étain entre pastilles.
- **Coupe de piste** (X) : coupe une bande ou retire une soudure.
- **Face de la plaque** (L) : bascule entre les deux faces, le dessus (composants, fils de liaison) et le dessous (cuivre, étain).
- **Soudure automatique** : transforme les fils posés sur la plaque en pistes et fils de liaison, et coupe les bandes si nécessaire.

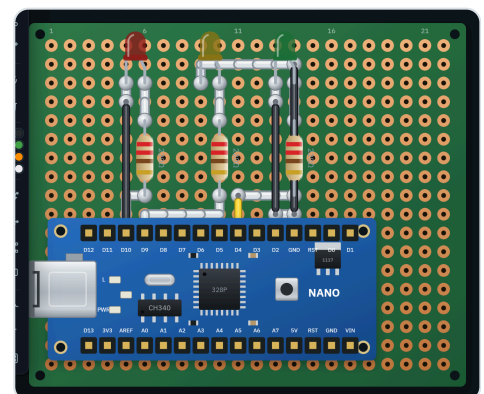


Figure 3.4 — Feu tricolore soudé automatiquement sur plaque pastillée.

■ Étiquettes de texte

L'outil **Étiquette** (T) pose un texte sur le plan : nom d'un sous-ensemble, consigne, remarque. Double-cliquez dessus pour le modifier ; couleur, taille et cadre se règlent dans ses propriétés.

Le simulateur surveille votre montage

Il signale les erreurs qui abîmeraient les vrais composants : LED sans résistance, court-circuit, courant trop fort sur une broche, module 3,3 V branché en 5 V, moteur alimenté directement par la carte... Lisez ces avertissements : ce sont les mêmes pièges qu'en vrai.

4 Programmer en code

Chaque carte a son propre programme, écrit en langage Arduino comme dans l'IDE officiel. L'éditeur aide à l'écriture et le simulateur l'exécute.

■ L'onglet Code

Le menu en haut de l'onglet choisit la carte dont on modifie le code (utile avec plusieurs cartes). Les boutons :

- **Nœuds** : programmer avec des blocs (chapitre 5).
- **Vérifier** : contrôle le programme sans lancer la simulation et place le curseur sur la première erreur.
- **Référence** : liste des fonctions et bibliothèques prises en charge.
- **.ino** : exporte le croquis pour l'IDE Arduino.
- **Téléverser** : envoie sur la vraie carte (chapitre 11).

■ Écrire plus vite

Ctrl + Espace	Autocomplétion. Elle s'ouvre aussi seule après un point : <code>lcd.</code> propose <code>print</code> , <code>setCursor</code> ... avec leurs paramètres.
Survol	Laisser la souris sur une fonction affiche son aide en français.
Maj + Alt + F	Remet en forme tout le code (indentation, espaces).
Ctrl + F	Rechercher. Ctrl + H : rechercher et remplacer, avec expressions régulières.
Ctrl + Q	Plier / déplier la fonction sous le curseur. Les boutons de la barre plient ou déplient tout.
Ctrl + S	Enregistrer le projet.

■ Plusieurs fichiers

Le bouton + à côté du nom du croquis ajoute un onglet `.h` ou `.cpp`, comme dans l'IDE Arduino. Un nouveau `.h` est inclus automatiquement dans le croquis principal. Double-cliquez sur un onglet pour le renommer. Les erreurs s'affichent dans le bon fichier, à la bonne ligne. Au téléversement, les fichiers sont réunis en un seul croquis.



Figure 4.1 — L'autocomplétion connaît les fonctions de chaque bibliothèque.

■ Ce que le simulateur comprend

Le langage Arduino courant : variables et tableaux, `String`, fonctions, structures, `switch`, interruptions (`attachInterrupt`, `ISR`), `millis()`, `delay()`, division entière comme sur la vraie carte. Parmi les bibliothèques simulées :

Affichage

LiquidCrystal I²C, Adafruit SSD1306 / GFX, TFT ST7735, ST7789, ILI9341, TFT_eSPI, Adafruit GC9A01A (écran rond), GxEPD2 (e-paper), LiquidCrystal, TM1637, LedControl (MAX7219), Adafruit NeoPixel

Capteurs

DHT, DallasTemperature, MPU6050, BMP280 / BME280, TinyGPS++, MFRC522 (RFID), IRremote, RTCLib, Keypad

Moteurs

Servo, ESP32Servo, Stepper, AccelStepper

Communication

Wire (I²C), SPI, Serial1/2/3, SoftwareSerial, RF24 (nRF24L01)

Réseau

WiFi, WebServer, HTTPClient, PubSubClient (MQTT), BluetoothSerial, BLE série

Mémoire et énergie

EEPROM, SD, watchdog, LowPower, veille profonde ESP32

Astuce

En cas de doute, cliquez sur [Référence](#) : la liste exacte des fonctions reconnues s'affiche, avec leur description.

5 Programmer avec des blocs

L'éditeur de nœuds permet de programmer sans écrire de code : on relie des blocs, et Zero1023 écrit le programme Arduino correspondant.

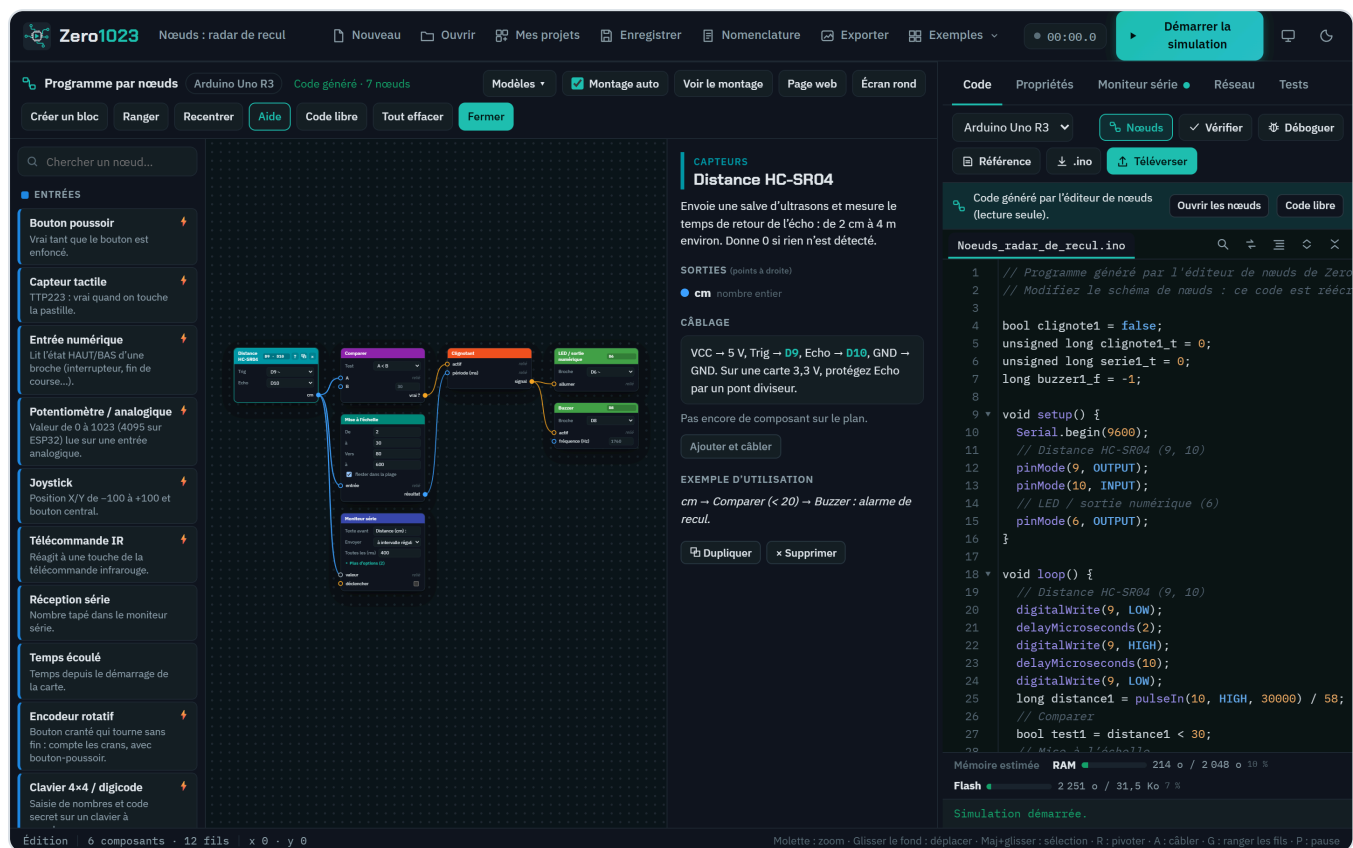


Figure 5.1 — Le radar de recul en nœuds : palette à gauche, schéma au centre, aide du bloc sélectionné à droite, code généré dans le panneau.

■ Premiers pas

- 1 Ouvrez l'éditeur avec **Nœuds** (onglet Code) ou la touche **N**.
- 2 **Ajoutez** des blocs depuis la palette, en les glissant ou d'un clic. Les blocs marqués ⚡ ajoutent aussi leur composant au montage et le câblent sur la broche choisie (case **Montage auto**).
- 3 **Reliez** la sortie d'un bloc (point à droite) à l'entrée d'un autre (point à gauche). L'information circule de gauche à droite.
- 4 Lancez la simulation : les valeurs s'affichent en direct à côté des sorties et les liens « vrai » s'illuminent.

■ Partir d'un modèle

Plutôt que de partir d'un schéma vide, insérez un **modèle** : un petit programme complet et déjà relié, que vous adaptez ensuite.

- 1 Cliquez sur **Modèles** dans la barre de l'éditeur. Quand le schéma est vide, la liste s'affiche aussi au centre, sous « Ou partez d'un modèle ».
- 2 Les modèles sont rangés par thème : **Pour débiter**, **Automatismes et capteurs**, **Lumière et son**, **Affichage, Écran rond**, **Sans fil**. Chaque fiche indique ce que fait le modèle et le matériel utilisé ; le champ de recherche filtre par mot (« thermostat », « servo », « écran »...).
- 3 Cliquez sur une fiche : les blocs sont ajoutés au schéma. Avec **Montage auto** coché, les composants sont aussi posés sur le plan et câblés.

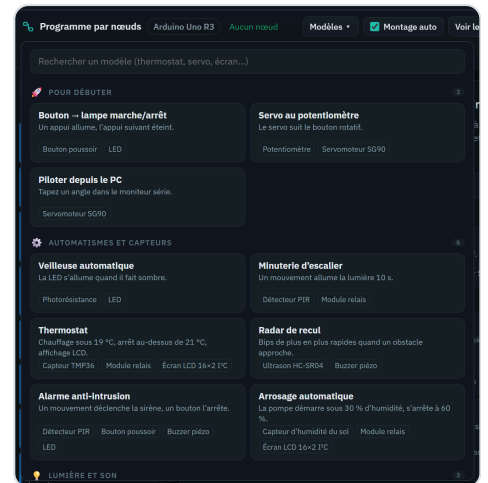


Figure 5.2 — La liste des modèles, classés par thème, avec le matériel de chacun.

■ La couleur des points

● **vrai / faux** · ● **nombre entier** · ● **nombre décimal** · ● **nombre ou vrai / faux** · ● **texte**. Une entrée non reliée utilise la valeur de sa case.

■ Les familles de blocs

FAMILLE	EXEMPLES DE BLOCS
Entrées, capteurs	Bouton, capteur tactile, potentiomètre, joystick, télécommande IR, distance HC-SR04, luminosité, DHT, DS18B20, mouvement, humidité du sol, pression BMP280, niveau de batterie, balance HX711, wattmètre INA219, distance laser VL53L0X, empreinte digitale
Logique, calcul	Comparer, ET / OU / NON, seuil avec hystérésis, si... alors... sinon, choisir parmi 4 , opérations, mise à l'échelle, lissage, aléatoire, onde, minimum / maximum, vitesse de variation, couleur TSV → RVB, dégradé de couleur
Temps et mémoire	Clignotant, métronome, bascule, marche / arrêt, front, compteur, minuterie, chronomètre, mémoire permanente (EEPROM), plage horaire, veille profonde (ESP32)
Sorties, moteurs	LED, LED de la carte , relais, PWM, LED RVB, buzzer, mélodie, NeoPixel, servo, pas à pas, moteur CC, vérins
Affichage	Écran OLED (texte, formes, jauge, graphique, image, QR code), LCD, TM1637, moniteur série
Écran rond	Réglages écran rond, jauge ronde, cadran à aiguille, valeur, texte, horloge à aiguilles, voyant, icône, barre, courbe (voir « L'écran rond » plus loin)
Sans fil	Connexion Wi-Fi, Bluetooth série, MQTT publier / recevoir, radio directe ESP-NOW envoyer / recevoir, LoRa envoyer / recevoir, entre cartes (Wi-Fi) envoyer / recevoir, notification sur le téléphone, canal sécurisé (chapitres 7 et 8)
GSM	Module SIM800L, SMS envoyer / recevoir un ordre, téléphoner, appel entrant, GPRS vers Internet (chapitre 8)
Page web	Bouton, curseur, valeur, voyant, message, graphique, saisie, choix, couleur, pavé directionnel, joystick, roue chromatique, bouton rotatif, compteur + / -, heure, journal, alerte, texte web ; caméra du téléphone : photo pour l'écran, pipette à couleur, détecteur de mouvement, lecteur de QR code (chapitre 8)
Variables	Variable, lire une variable, régler une variable, changer une variable
Texte	Texte, composer un texte, texte selon une condition, comparer des textes, découper un texte, infos sur un texte, texte reçu (moniteur série)
Notes	Note, cadre

■ Variables et texte

Une **variable** est une case mémoire avec un nom. On la règle à un endroit du schéma et on la lit ailleurs, **sans tirer de fil** : les grands schémas restent lisibles.

- 1 Posez un bloc **Variable** : choisissez son nom (« consigne »), son type (nombre entier, nombre décimal, vrai / faux ou **texte**) et sa valeur de départ.
- 2 **Régler une variable** lui donne une nouvelle valeur : une fois, au moment où « déclencher » devient vrai (un appui, un message reçu...), ou en continu tant que « autoriser » est vrai.
- 3 **Changer une variable** ajoute ou retire une quantité, avec un minimum et un maximum si besoin. Sur une variable texte, la quantité est ajoutée à la fin ; sur vrai / faux, chaque déclenchement inverse la valeur.

- 4 **Lire une variable** donne sa valeur n'importe où. Les blocs Lire, Régler et Changer proposent la liste des variables existantes ; si vous renommez une variable, ils suivent automatiquement.

Les blocs de la famille **Texte** (points verts) manipulent des messages : **Composer un texte** remplit un modèle comme « T {a} °C · H {b} % » avec des valeurs ; **Découper un texte** extrait « 90 » de « servo=90 » (et le donne aussi en nombre) ; **Comparer des textes** teste « égal », « contient », « commence par »... ; **Texte reçu** lit une ligne tapée dans le moniteur série. Le moniteur série, les écrans LCD et OLED, les SMS et la page web (bloc **Texte web**) affichent directement du texte. Un texte relié à une entrée qui attend un nombre est converti automatiquement.

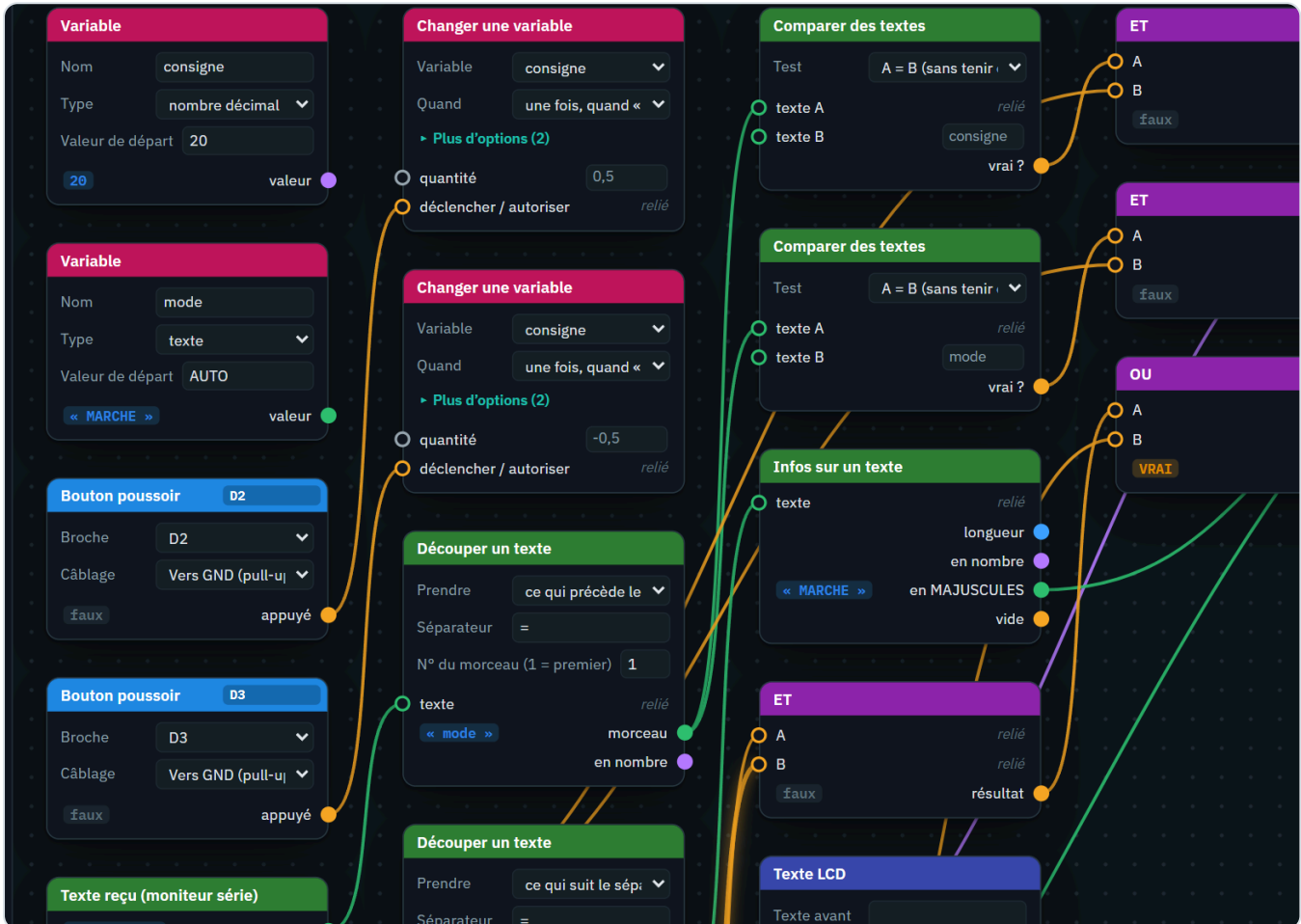


Figure 5.3 — Le thermostat à commandes : variables « consigne » (décimal) et « mode » (texte, qui vaut « MARCHE » après la commande `mode=marche`), boutons + / –, découpage de la commande reçue.

Dans le code généré

Chaque variable devient une vraie variable Arduino du même nom (`float consigne = 20;` ; `String mode = "AUTO";`); les accents et espaces sont remplacés (« Température voulue » → `Temperature_voulue`). Le débogueur les affiche sous ce nom.

À essayer

L'exemple « Variables et texte : thermostat à commandes » : tapez `consigne=22.5` ou `mode=marche` dans le moniteur série.

■ Blocs réutilisables et cadres

Quand un groupe de blocs revient souvent (un bouton qui commande une lampe, une régulation...), faites-en **votre propre bloc** :

- 1 Sélectionnez les blocs (**Maj** + clic ou **Maj** + glisser), puis **Créer un bloc** dans la barre (ou clic droit → « Créer un bloc réutilisable... ») et donnez-lui un nom.
- 2 Les blocs sélectionnés sont remplacés par un seul bloc, rangé dans la catégorie **Mes blocs** de la palette. Ses entrées et sorties sont celles des liens qui traversaient la sélection.
- 3 Posez-en autant de copies que nécessaire. Chaque copie a **ses propres broches**, réglables dans le bloc, et un numéro ; écrivez `{n}` dans un texte pour y afficher ce numéro.
- 4 **Double-cliquez** sur un bloc pour l'ouvrir : une modification s'applique à toutes ses copies. **Terminer** referme le bloc, **Renommer** change son nom.

Le bloc **Cadre** (catégorie Notes) entoure une partie du schéma d'un fond de couleur avec un titre. Glissez son titre pour déplacer le cadre **avec tous les blocs qu'il contient** ; tirez son coin en bas à droite pour l'agrandir. Il n'a aucun effet sur la carte, mais son titre apparaît en commentaire dans le code généré.

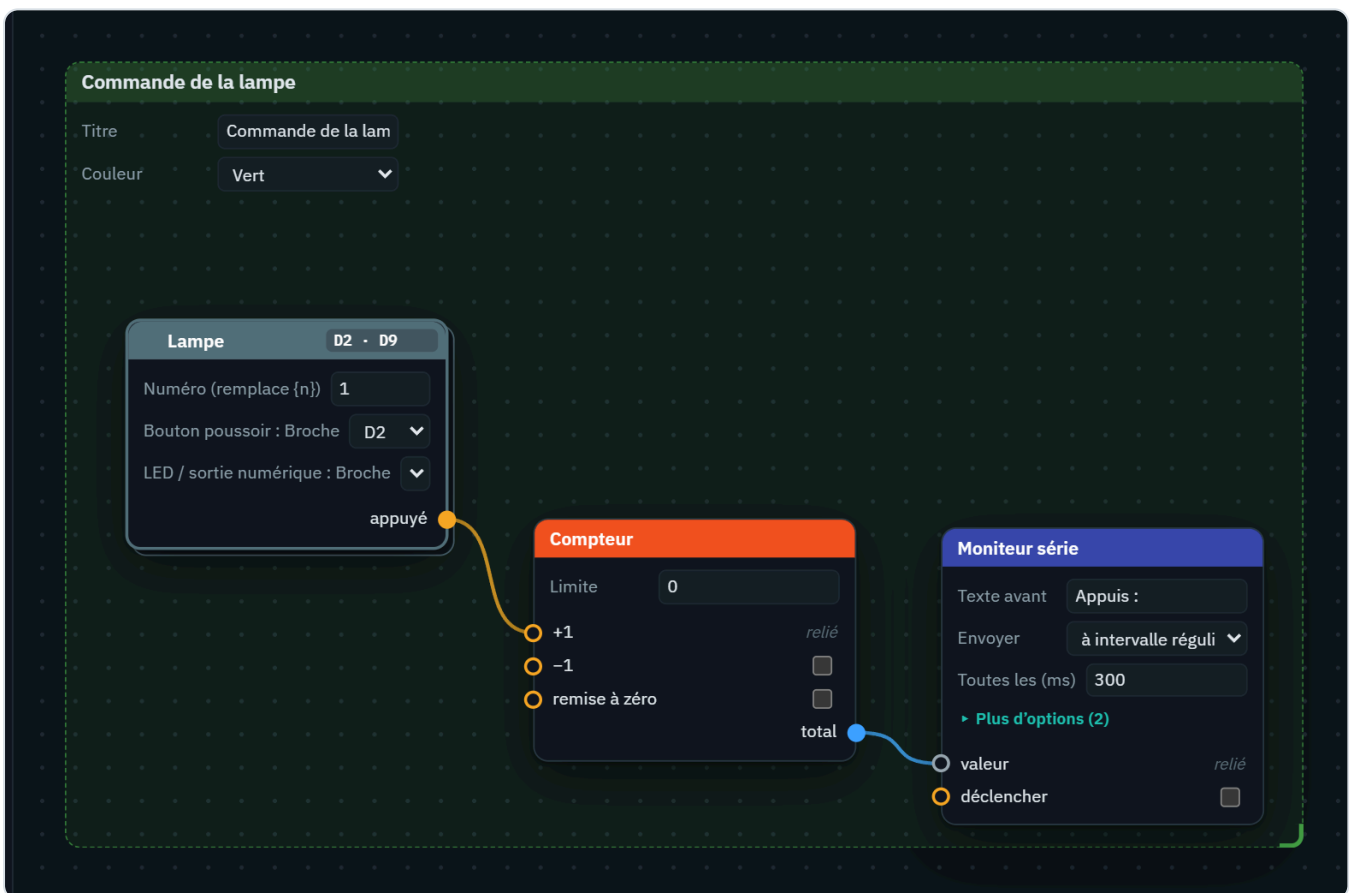


Figure 5.4 — Un bloc réutilisable « Lampe » (bouton + bascule + LED) dans un cadre « Commande de la lampe ».

À savoir

Le montage automatique ne câble pas les composants des blocs réutilisables : câblez-les à la main, sur les broches indiquées dans chaque copie.

■ Outils de l'éditeur

- **Modèles** insère un petit programme prêt à l'emploi (voir « Partir d'un modèle »). **Ranger** aligne les blocs en colonnes ; **Recentrer** recadre la vue.
- **Voir le montage** réduit l'éditeur pour garder le montage et la simulation visibles.
- **Créer un bloc** transforme la sélection en bloc réutilisable (voir ci-dessus).
- **Aide** affiche ou masque le panneau d'aide : fonctionnement du bloc, câblage conseillé et exemple d'utilisation.
- **Page web** ouvre le configurateur de la page servie par la carte (chapitre 8) ; **Écran rond** ouvre celui de l'écran rond (voir ci-dessous).
- **Code libre** arrête la génération : le code redevient modifiable à la main.
- Clic droit : menu contextuel. **Ctrl** + **C** / **V** copier-coller, **Ctrl** + **D** dupliquer, **Suppr** effacer ; double-clic sur un lien pour le supprimer.

Le code généré est un vrai programme Arduino

Il s'affiche en lecture seule dans l'onglet Code, se téléverse tel quel et peut servir de point de départ pour apprendre le langage.

■ L'écran rond GC9A01

L'écran rond de 1,28 pouce (240 × 240 points, puce GC9A01) donne des montres, compteurs et tableaux de bord du plus bel effet. Les blocs de la famille **Écran rond** le pilotent sans écrire de code graphique, et une fenêtre de conception règle son apparence à la souris.

Mettre en place l'écran

- 1 Le plus simple : **Modèles** → famille **Écran rond** (thermomètre, compteur de vitesse, montre, cardio, tableau de bord, deux pages au bouton). L'écran et les capteurs sont posés et câblés.
- 2 Sinon, ajoutez le bloc **Réglages écran rond** : avec Montage auto, l'écran est posé et câblé sur le bus SPI de la carte (SCL → horloge SCK, SDA → MOSI ; DC, CS et RST sur les broches indiquées dans le bloc ; VCC sur **3,3 V**). Le bloc règle aussi l'orientation, la fréquence de rafraîchissement et la page affichée.
- 3 Ajoutez les éléments à afficher : **jauge ronde**, **cadran à aiguille**, **valeur**, **texte**, **horloge à aiguilles**, **voyant**, **icône**, **barre**, **courbe**. Reliez leur entrée à une mesure (potentiomètre, capteur, calcul...).

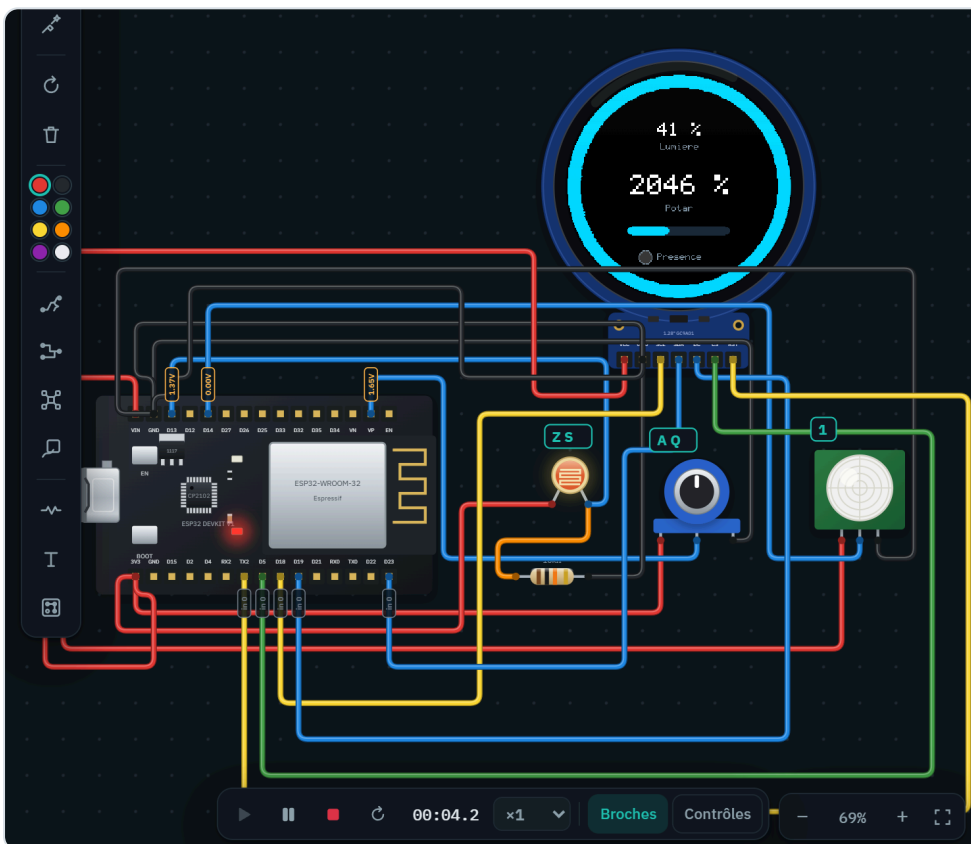


Figure 5.5 — Le modèle « tableau de bord » en simulation : luminosité, potentiomètre et présence sur l'écran rond.

Concevoir l'affichage

Ouvrez la fenêtre **Écran rond** avec le bouton du même nom dans la barre de l'éditeur, ou le bouton  d'un bloc de l'écran. L'aperçu au centre utilise la même police et les mêmes tracés que la vraie carte.

- **Thème** : thèmes prêts (Nuit, Montre, Néon, Voiture, Océan, Forêt...), puis couleurs du fond, du texte, de l'accent, de la piste des jauges et de la couronne ; fond en dégradé ; forme et épaisseur de la couronne.
- **Éléments** : pour chacun, position, taille, couleurs propres, titre, minimum et maximum, seuils de couleur, page où il apparaît.
- **Pages** : plusieurs écrans qui se succèdent (par exemple mesures, puis horloge). Le bloc Réglages écran rond choisit la page affichée pendant le programme, par exemple à chaque appui sur un bouton.

- **Placer** : glissez un élément dans l'aperçu ; flèches du clavier pour avancer d'un point, **Maj** + flèches pour dix. **Aimanter au centre** aide à centrer, et le curseur **Valeurs d'exemple** fait varier toutes les mesures pour vérifier le rendu.

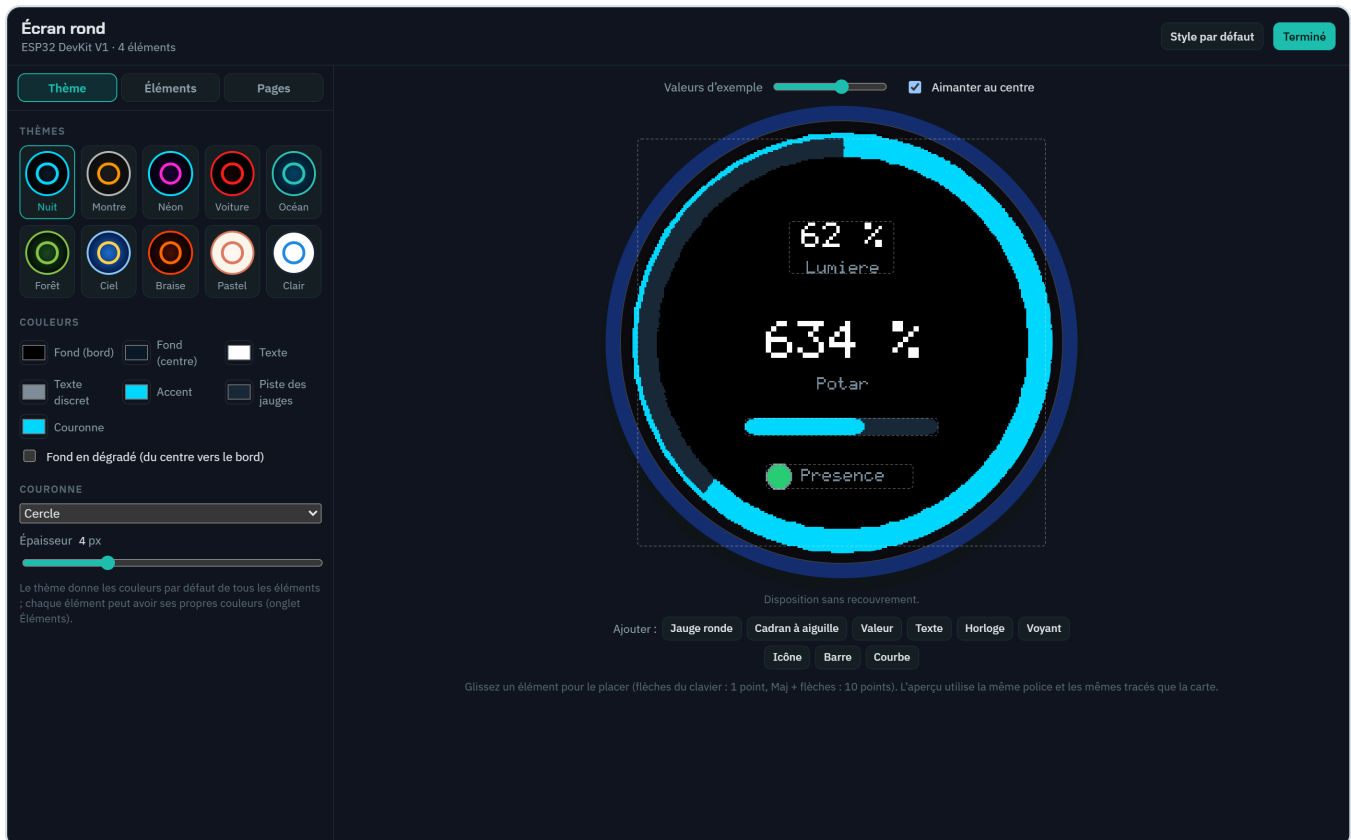


Figure 5.6 — La fenêtre de conception de l'écran rond : thème à gauche, aperçu au centre, éléments à ajouter dessous.

Sur la vraie carte

Le code généré utilise la bibliothèque `Adafruit_GC9A01A` (installée automatiquement au téléversement). Les valeurs sont redessinées sans clignotement, et les lettres accentuées s'affichent.

6 Lancer la simulation

Une fois le montage câblé et programmé, la simulation fait tourner chaque carte en temps réel : on voit, on entend, et on agit sur le montage.

■ Démarrer, mettre en pause, arrêter

Démarrer la simulation (ou **Maj** + **Entrée**) lance toutes les cartes. La barre de simulation propose ensuite pause (**P**), arrêt et redémarrage ; le chronomètre affiche le temps simulé.

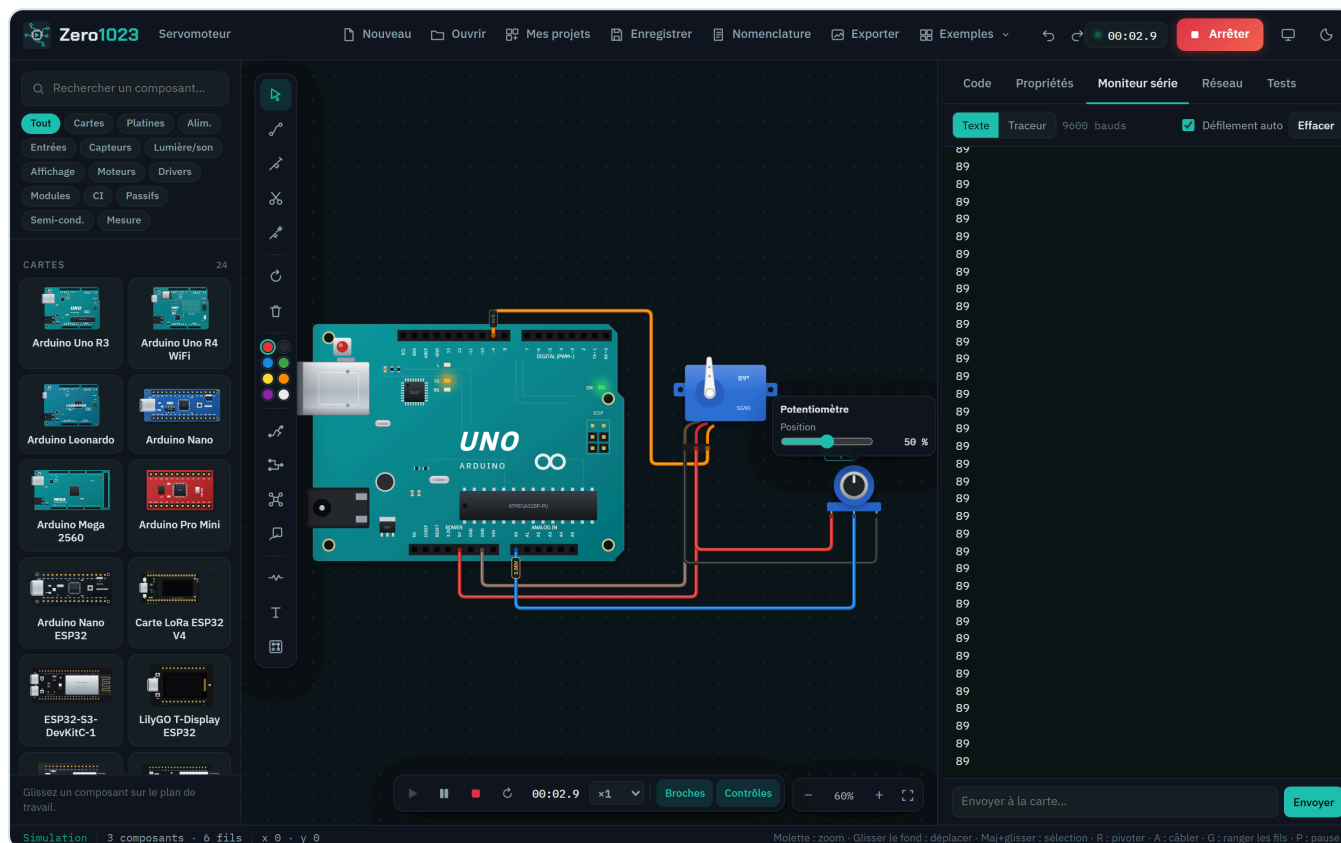


Figure 6.1 — Le servo suit le potentiomètre ; le moniteur série affiche l'angle envoyé par le programme.

■ Agir sur le montage

- **Cliquer** sur un bouton, un interrupteur, une touche de clavier matriciel ou une carte RFID ; **faire glisser** un potentiomètre, un joystick, l'inclinaison d'un MPU6050.
- **Contrôles** pose des curseurs directement sur les capteurs : distance d'un HC-SR04, température d'un DHT22, lumière, humidité du sol...
- **Broches** affiche l'état de chaque broche utilisée : **HIGH**, **LOW**, tension, PWM.

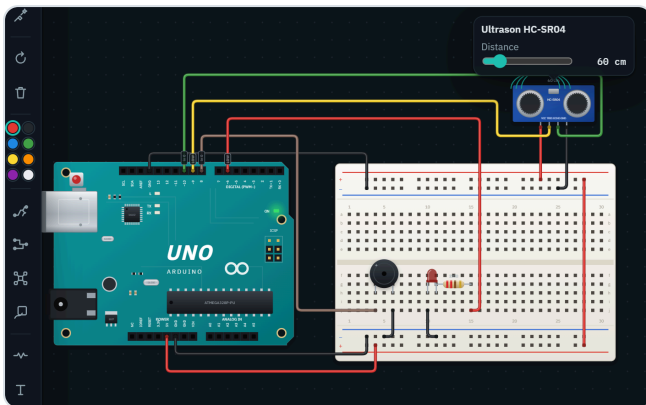


Figure 6.2 — Curseur de distance posé sur le capteur à ultrasons, états des broches sur la carte.

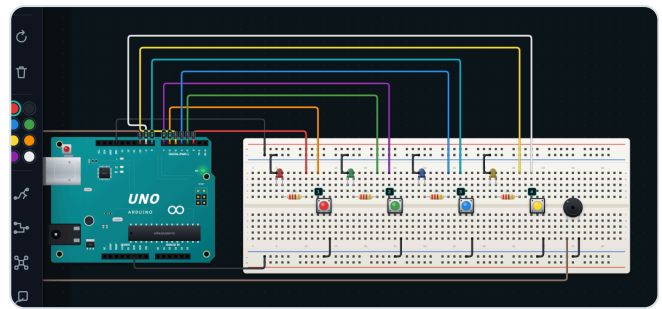


Figure 6.3 — Touches 1 à 4 attribuées aux boutons du jeu Simon.

Piloter au clavier

Au démarrage, Zero1023 attribue des touches aux composants que l'on actionne ; un badge l'indique au-dessus de chacun :

- boutons, interrupteurs, capteurs tout-ou-rien : touches **1** à **9**, sans **Maj** même sur un clavier AZERTY ;
- potentiomètres, glissières, photorésistances, encodeurs : **deux touches**, une par sens (**A**/**Q**, **Z**/**S**...).
- joystick et MPU6050 : les **flèches**.

Pour choisir une autre touche, sélectionnez le composant, cliquez dans le champ **Touche clavier** de ses propriétés et appuyez simplement sur la touche voulue (**Suppr** pour l'effacer).

Voir un écran en grand

Double-cliquez sur un écran (OLED, LCD, TFT, e-paper, afficheur 7 segments, matrice, NeoPixel...) : il s'ouvre dans une fenêtre agrandie, que l'on peut déplacer, redimensionner ou passer en plein écran. Pratique pour lire un petit écran ou le montrer à une classe.

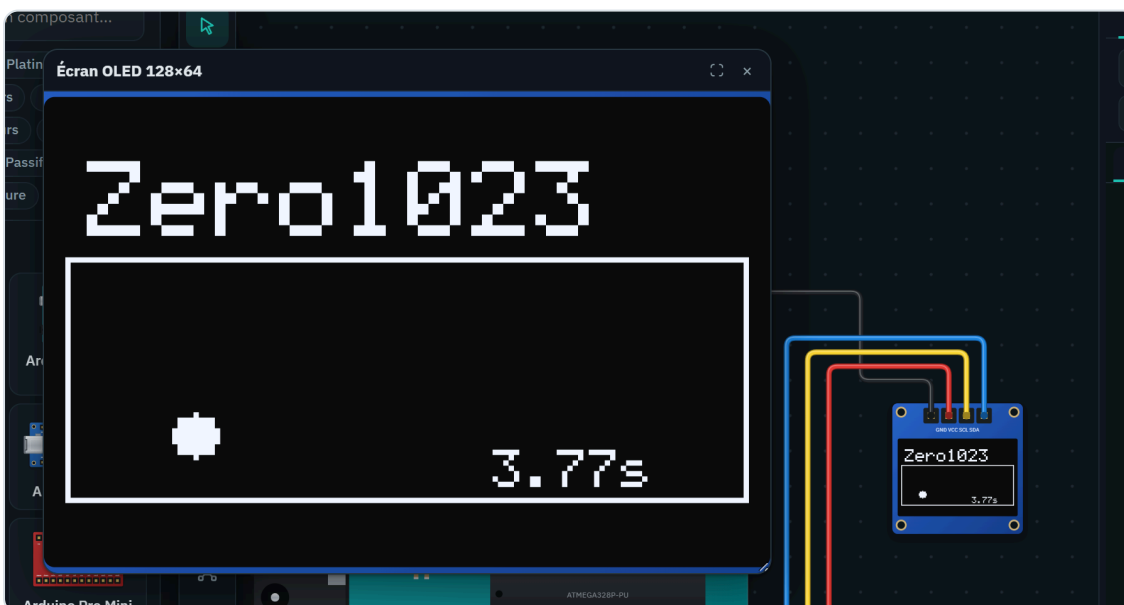


Figure 6.4 — L'écran OLED agrandi, à côté du vrai module.

■ Le moniteur série

L'onglet **Moniteur série** affiche ce que la carte envoie avec `Serial.print()`. La vitesse (bauds) est lue dans le programme. La zone du bas envoie du texte à la carte, comme dans l'IDE Arduino.

- **Texte / Traceur** : le traceur dessine les nombres reçus sous forme de courbes. Envoyez plusieurs valeurs séparées par des virgules ou des espaces, éventuellement nommées (`pot:512` `pwm:127`), pour obtenir plusieurs courbes.
- **Défilement auto** suit les dernières lignes ; **Effacer** vide le journal.
- Avec plusieurs cartes, des boutons permettent de n'afficher qu'une carte, ou toutes **côte à côte** (chapitre 7).

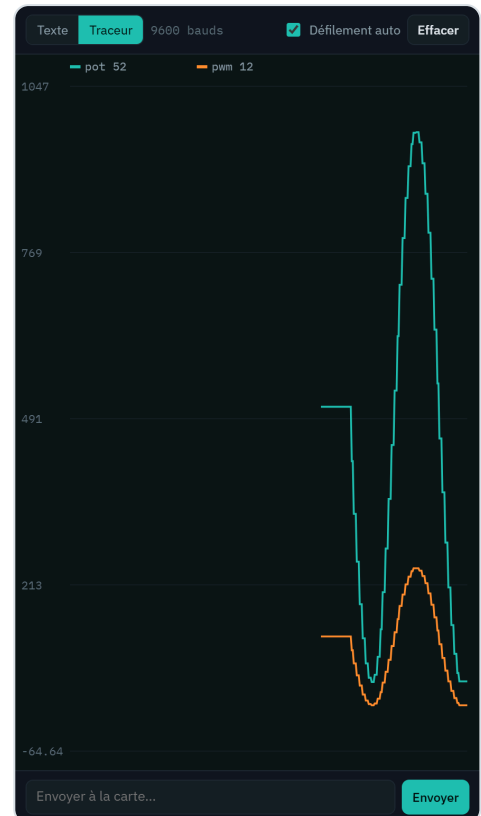


Figure 6.5 — Le traceur série : valeur du potentiomètre et PWM de la LED.

■ Vitesse de simulation

Le menu **×1** de la barre de simulation change la vitesse du temps simulé, de **×0,1** (ralenti, pour suivre un clignotement rapide ou un protocole) à **×10** (pour ne pas attendre une minuterie de plusieurs minutes). Le chronomètre, `millis()` et `delay()` suivent le temps simulé : le programme se comporte exactement comme à vitesse normale.

■ Mémoire et courants : comme sur la vraie carte

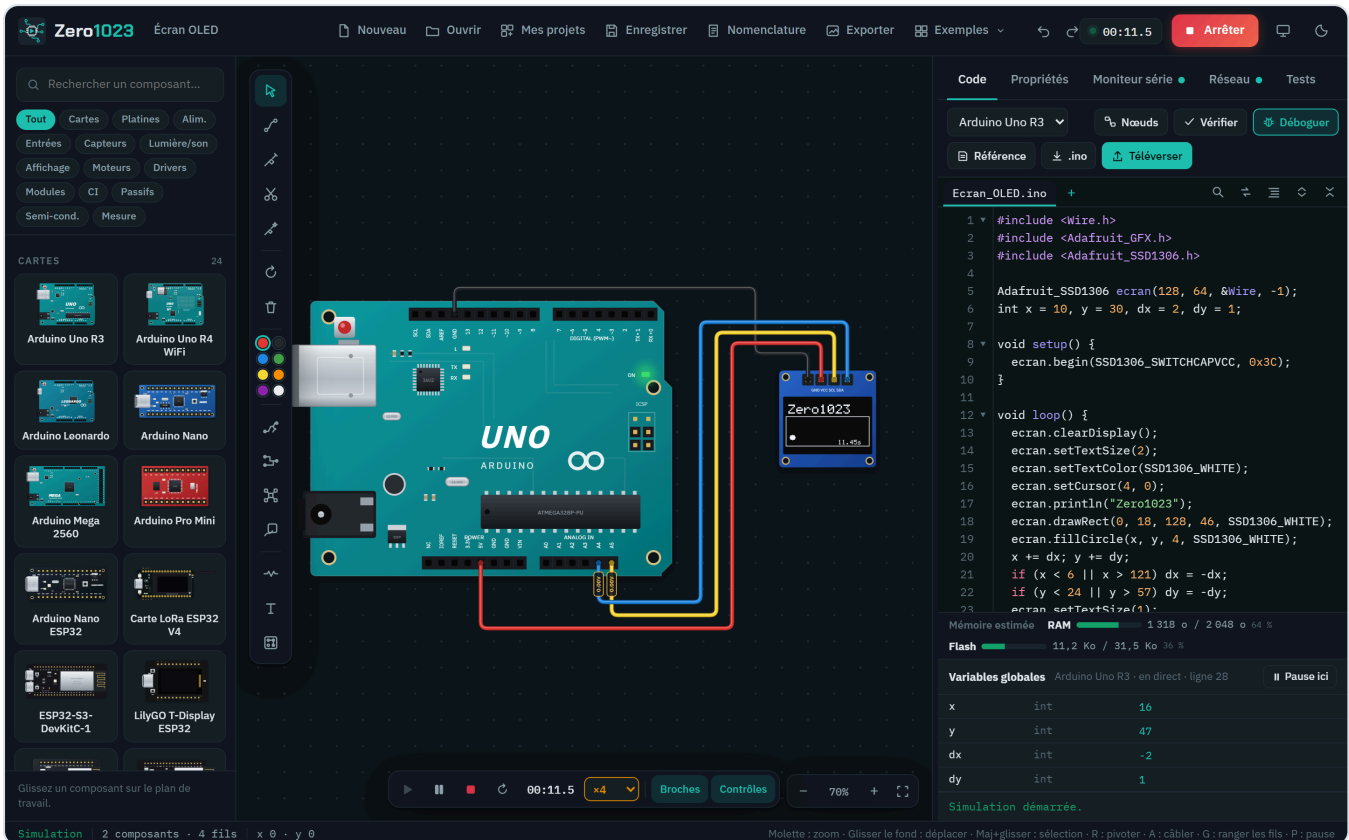


Figure 6.6 — Simulation accélérée x4 ; en bas à droite, la mémoire estimée du programme (RAM et flash).

- **Mémoire estimée** : sous l'éditeur de code, deux jauges estiment la RAM (variables, tableaux, bibliothèques) et la flash (programme) utilisées, comparées à la capacité de la carte. Au-delà de 85 %, un message prévient ; au-delà de 100 %, le programme ne tiendrait pas sur la vraie carte. C'est une estimation : l'IDE Arduino donne le chiffre exact à la compilation.
- **Calculs comme sur la carte** : sur Uno, Nano, Mega..., un `int` tient sur 16 bits et déborde après 32 767 (il repasse à -32 768), comme en vrai. Sur ESP32, il tient sur 32 bits. Les divisions entières et les `unsigned` sont respectés.
- **Courants** : chaque carte a ses limites (par broche, au total, sur les sorties 3,3 V et 5 V). Si un montage les dépasse, par exemple trop de LED sans résistance suffisante, un avertissement le signale.

■ Déboguer pas à pas

- 1 Cliquez dans la **marge de gauche** de l'éditeur, devant un numéro de ligne : un point rouge marque un **point d'arrêt**. Recliquez pour l'enlever.
- 2 Lancez la simulation : la carte s'arrête **juste avant** d'exécuter cette ligne, qui est surlignée en jaune. Le montage reste figé dans l'état du moment.
- 3 Dans le bandeau : **Continuer** (F8) jusqu'au prochain point d'arrêt, **Ligne suivante** (F10) pour avancer d'une seule ligne, **Sans points d'arrêt** pour tous les retirer.
- 4 Le bouton **Déboguer** affiche les **variables globales** et leur valeur en direct. On peut en **modifier une** pendant la pause pour tester un cas. **Pause ici** (F9) arrête la carte là où elle en est.

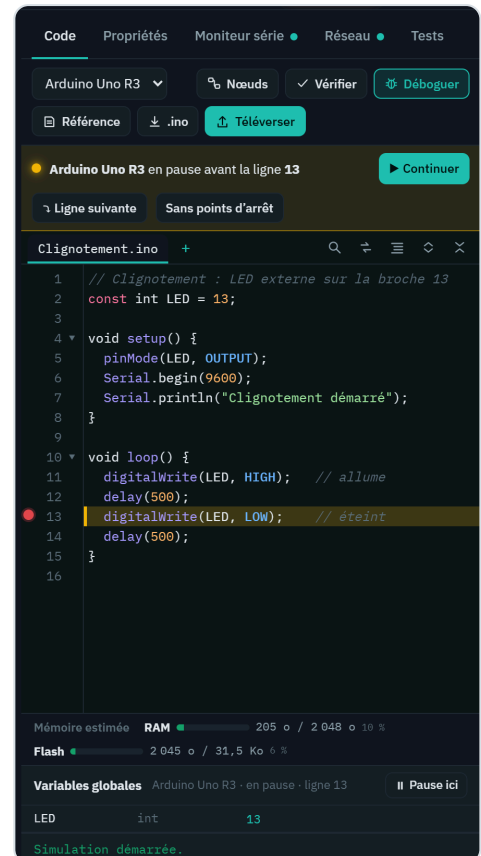


Figure 6.7 — Point d'arrêt ligne 13 : la carte s'est arrêtée avant d'éteindre la LED ; en bas, la valeur des variables globales.

■ Scénarios de test

L'onglet **Tests** rejoue automatiquement une suite d'actions et de contrôles, à des instants précis du temps simulé. Après chaque modification du programme, un clic sur **Lancer les tests** vérifie que tout marche encore.

- **+ Action** : appuyer sur un bouton (pendant une durée), régler un composant (curseur, distance, température...), envoyer un texte au moniteur série, envoyer un SMS à la carte ou l'appeler (module GSM).
- **+ Contrôle** : vérifier un composant (LED allumée, angle d'un servo, relais activé, texte d'un LCD...), une broche (HIGH , LOW ou une tension), le contenu du moniteur série, la valeur d'une variable, ou un SMS reçu par le téléphone.
- **Enregistrer mes gestes** démarre la simulation et transforme vos manipulations (appuis, réglages) en actions : il ne reste qu'à ajouter les contrôles.
- Chaque contrôle affiche ✓ ou X ; survolez un X pour voir la valeur mesurée et la valeur attendue. Le scénario est enregistré avec le projet.

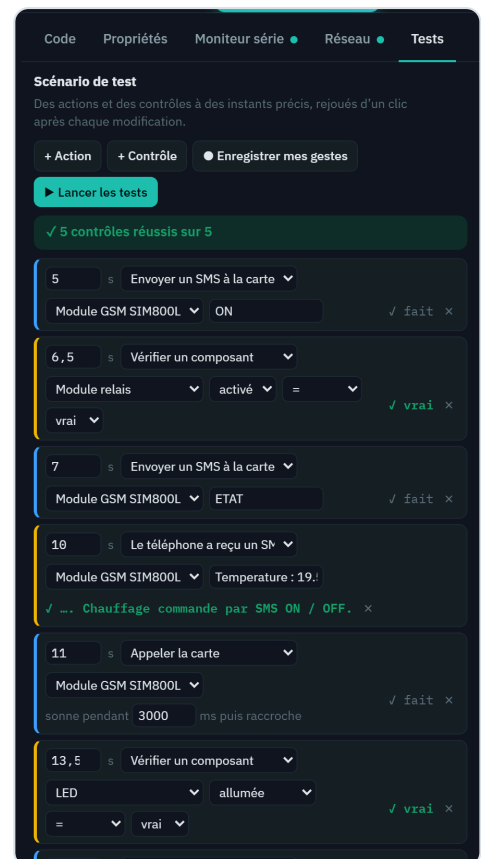


Figure 6.8 — Le scénario de l'alarme GSM : SMS envoyés, appel, détection de mouvement, et cinq contrôles réussis.

À essayer

Les exemples « ESP32-CAM : piège photo », « Deux ESP32 en radio directe (ESP-NOW) », « Alarme GSM », « Radio protégée : portail commandé sans pirate possible », « Serrure à empreinte digitale » et « Aide au stationnement avec un télémètre laser » sont fournis avec leur scénario de test.

7 Plusieurs cartes ensemble

Un même projet peut contenir plusieurs cartes, chacune avec son programme. Elles tournent en même temps et peuvent communiquer.

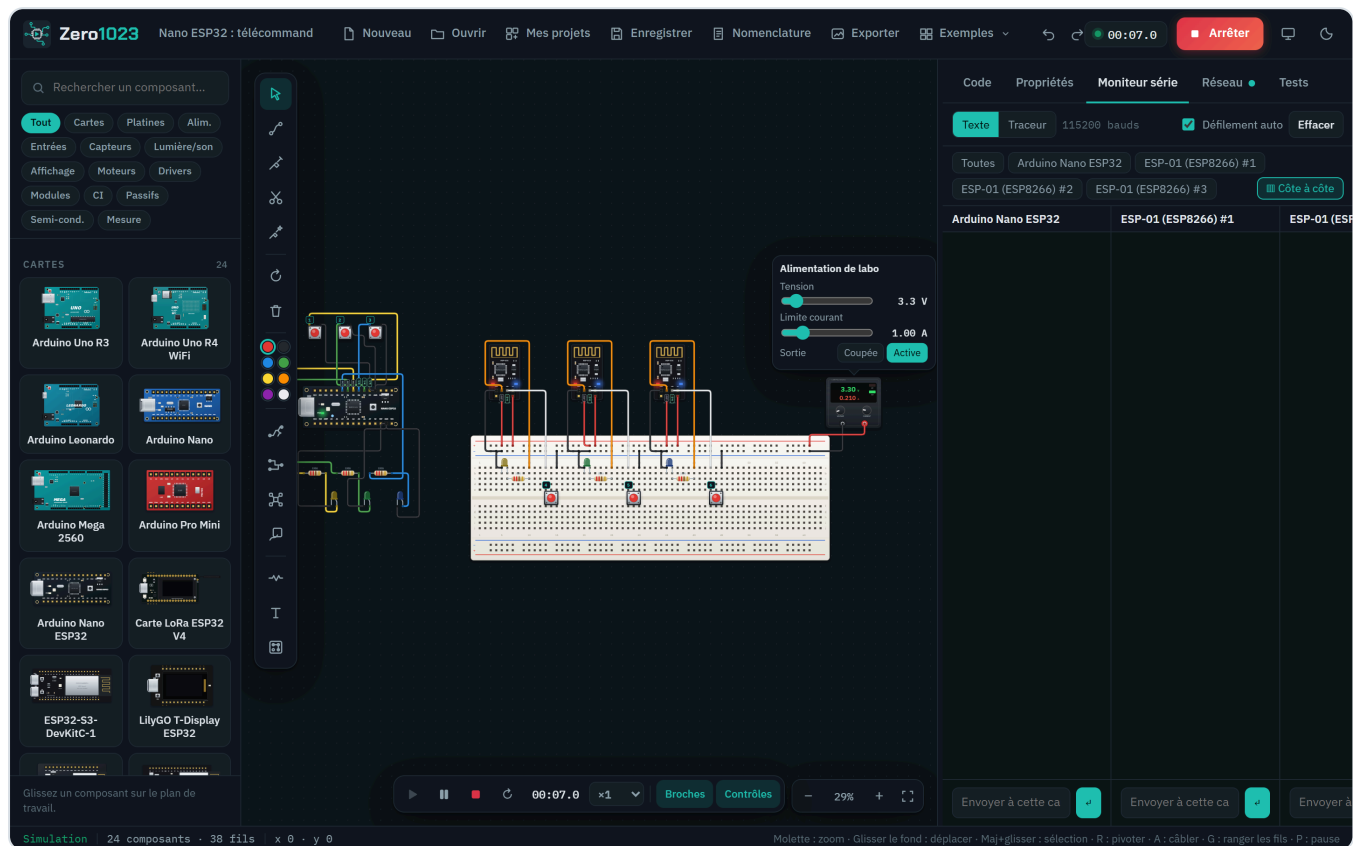


Figure 7.1 — Une Nano ESP32 et trois ESP-01 reliées en Wi-Fi, avec un moniteur série par carte, côte à côte.

Choisir la carte à programmer

Le menu en haut de l'onglet Code sélectionne la carte dont on voit le programme ; double-cliquer sur une carte du plan fait de même.

Les liaisons possibles

LIAISON	CÂBLAGE	DANS LE CODE
Série	TX de l'une → RX de l'autre (et inversement), masses reliées	<code>Serial1</code> , <code>Serial2</code> , <code>SoftwareSerial</code> . Si les vitesses ne correspondent pas, les caractères arrivent illisibles, comme en vrai.
I ² C	SDA ↔ SDA, SCL ↔ SCL, masses reliées	Maître : <code>Wire.beginTransmission</code> , <code>Wire.requestFrom</code> . Esclave : <code>Wire.begin(adresse)</code> , <code>Wire.onReceive</code> , <code>Wire.onRequest</code> .
Radio nRF24L01	Un module sur chaque carte (SPI)	Bibliothèque <code>RF24</code> ; les structures sont transmises telles quelles.

Wi-Fi direct	Aucun	Une carte crée un réseau (<code>WiFi.softAP</code>), les autres le rejoignent (<code>WiFi.begin</code> , adresse fixe avec <code>WiFi.config</code>) et s'interrogent avec <code>HTTPClient</code> et <code>WebServer</code> .
Radio directe ESP-NOW	Aucun (cartes ESP32)	Blocs « Radio directe : envoyer / recevoir » : une valeur nommée passe d'une carte à l'autre sans box, sans courtier ni mot de passe. En code : <code>esp_now_init</code> , <code>esp_now_add_peer</code> , <code>esp_now_send</code> , <code>esp_now_register_rcv_cb</code> ; le chiffrement intégré d'ESP-NOW (<code>esp_now_set_pmk</code> , clé <code>lmk</code> du pair) est simulé.
Entre cartes (Wi-Fi)	Aucun (même réseau Wi-Fi)	Blocs « Entre cartes (Wi-Fi) : envoyer / recevoir » : une valeur nommée est diffusée aux cartes du réseau, sans serveur. En code : <code>WiFiUDP</code> (<code>beginPacket</code> , <code>parsePacket</code> ...).
LoRa	Un module LoRa sur chaque carte (SPI)	Blocs « LoRa : envoyer / recevoir », ou la bibliothèque <code>LoRa</code> (<code>LoRa.begin</code> , <code>beginPacket</code> , <code>parsePacket</code> , <code>packetRssi</code>). Portée de plusieurs kilomètres (chapitre 8).
MQTT	Aucun	<code>PubSubClient</code> ou les blocs MQTT : les cartes publient et s'abonnent à des sujets via le courtier de l'onglet Réseau.

■ Un moniteur par carte

Dès qu'il y a deux cartes, le moniteur série affiche une rangée de boutons : **Toutes** (journal commun), le nom de chaque carte (journal d'une seule carte ; l'envoi part vers elle), et **Côte à côte** : une colonne par carte, chacune avec sa propre zone d'envoi.

À essayer

Exemples « Nano ESP32 : télécommande Wi-Fi de trois ESP-01 », « Deux ESP32 en MQTT : atelier et supervision », « Deux ESP32 en radio directe (ESP-NOW) », « LoRa : station météo sur batterie au fond du jardin » et « Alarme : détecteur en Wi-Fi direct et notification ».

8 Sans fil et page web

Les cartes ESP32 et ESP8266 se connectent à un réseau Wi-Fi simulé, servent des pages web, dialoguent avec un téléphone Bluetooth, échangent des messages MQTT et envoient des notifications. Un module LoRa porte à plusieurs kilomètres ; un module SIM800L envoie des SMS et passe des appels. Et un seul bloc suffit à chiffrer tous ces échanges.

L'onglet Réseau

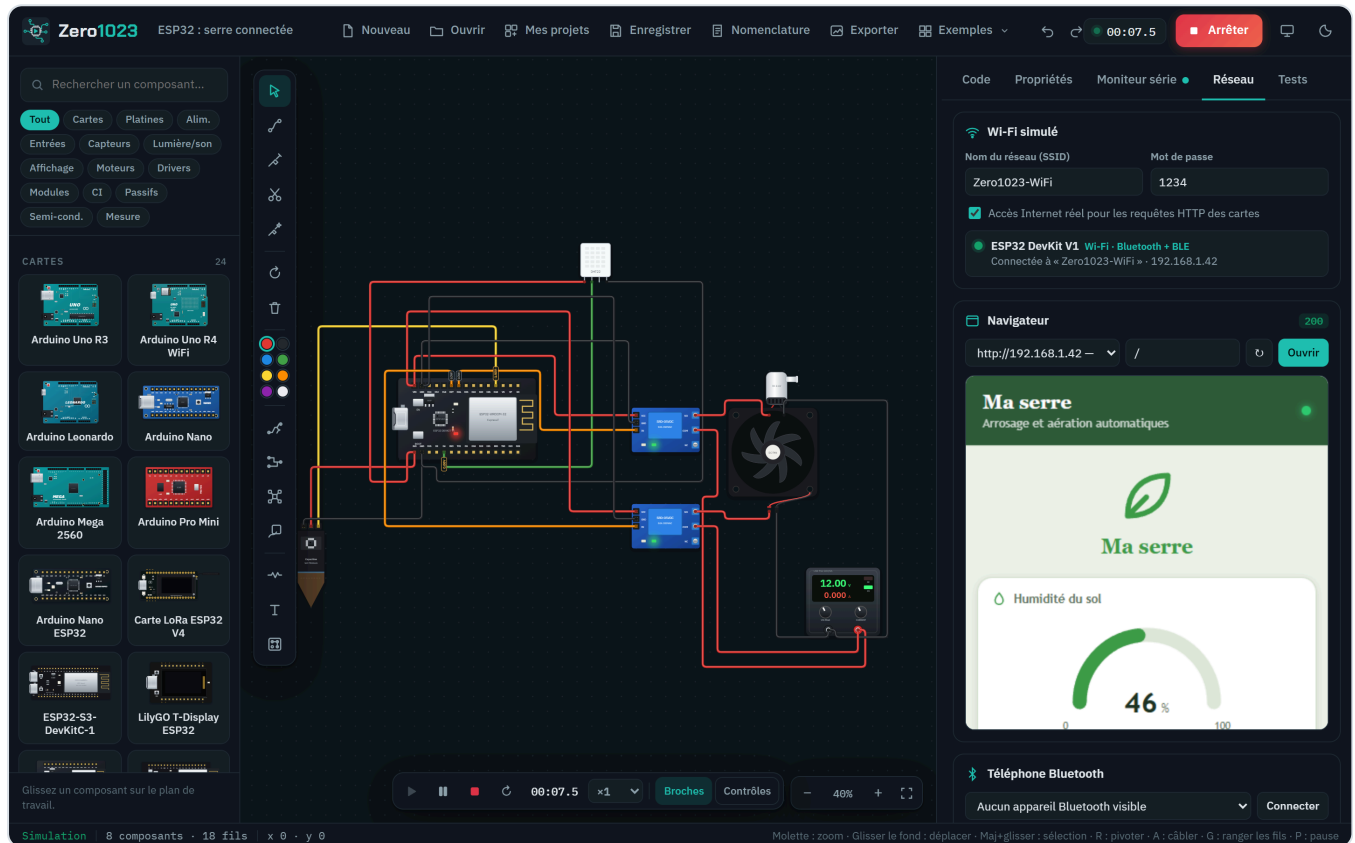


Figure 8.1 — La serre connectée : la page web servie par l'ESP32 s'affiche dans le navigateur intégré.

- **Wi-Fi simulé** : nom du réseau et mot de passe acceptés (par défaut `Zero1023-WiFi` / `1234`), accès Internet réel pour les requêtes HTTP des cartes, et liste des cartes connectées avec leur adresse IP.
- **Navigateur** : affiche la page servie par une carte. On peut changer d'adresse, recharger, cliquer sur ses boutons.
- **Téléphone Bluetooth** : se connecte à une carte qui utilise `BluetoothSerial` ou le BLE série, et échange des messages ; des boutons rapides (ON, OFF...) se personnalisent d'un double-clic.
- **Téléphone GSM** : apparaît dès qu'un module SIM800L est dans le montage ; vous jouez le téléphone (voir plus loin).
- **Courtier MQTT** : montre les connexions et tous les messages publiés, les abonnements et les messages conservés. Vous pouvez publier vous-même un message sur un sujet.
- **Notifications du téléphone** : les notifications envoyées par les cartes, comme sur l'écran d'un smartphone.

- **Espion radio** : tout ce que les cartes échangent sans fil, tel qu'un pirate à proximité le capterait (voir « Protéger les échanges »).

■ Deux façons de se connecter

Mode box : la carte rejoint le réseau (`WiFi.begin`) et reçoit une adresse du type 192.168.1.42.

Propre réseau : la carte devient un point d'accès (`WiFi.softAP`), joignable à 192.168.4.1 sans box. D'autres cartes peuvent le rejoindre.

Le bloc **Connexion Wi-Fi** propose par défaut le mot de passe 1234. Sur une vraie carte, un point d'accès exige au moins 8 caractères : avec un mot de passe plus court, le bloc crée un réseau **ouvert**, sans mot de passe, et le signale. Choisissez un mot de passe d'au moins 8 caractères pour un montage durable.

MQTT en vrai

Dans le simulateur, le courtier est intégré : l'adresse écrite dans le code n'a pas d'importance. Sur de vraies cartes, il faut un courtier : un service public ou Mosquitto installé sur un ordinateur ou un Raspberry Pi.

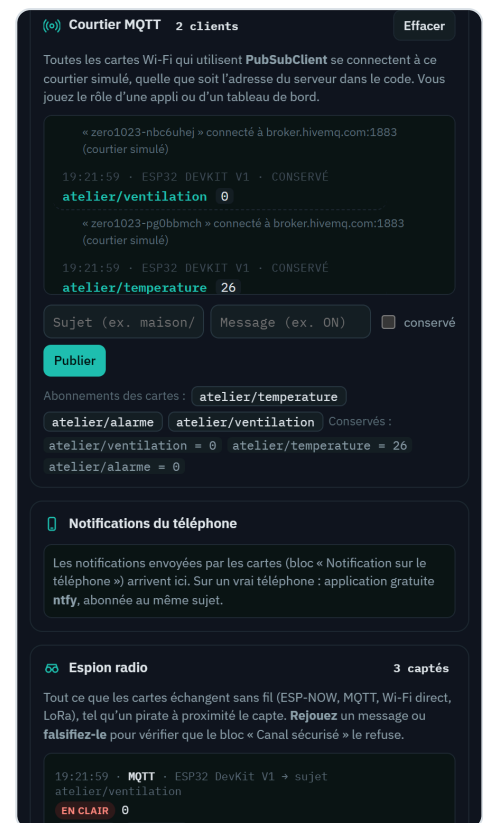


Figure 8.2 — Le courtier MQTT simulé.

■ Le configurateur de page web

Avec les blocs de la catégorie **Page web**, la carte sert automatiquement une page de pilotage. Le configurateur en règle l'apparence : ouvrez-le avec `Page web` dans la barre de l'éditeur de nœuds, ou `Personnaliser la page web...` dans un bloc web.

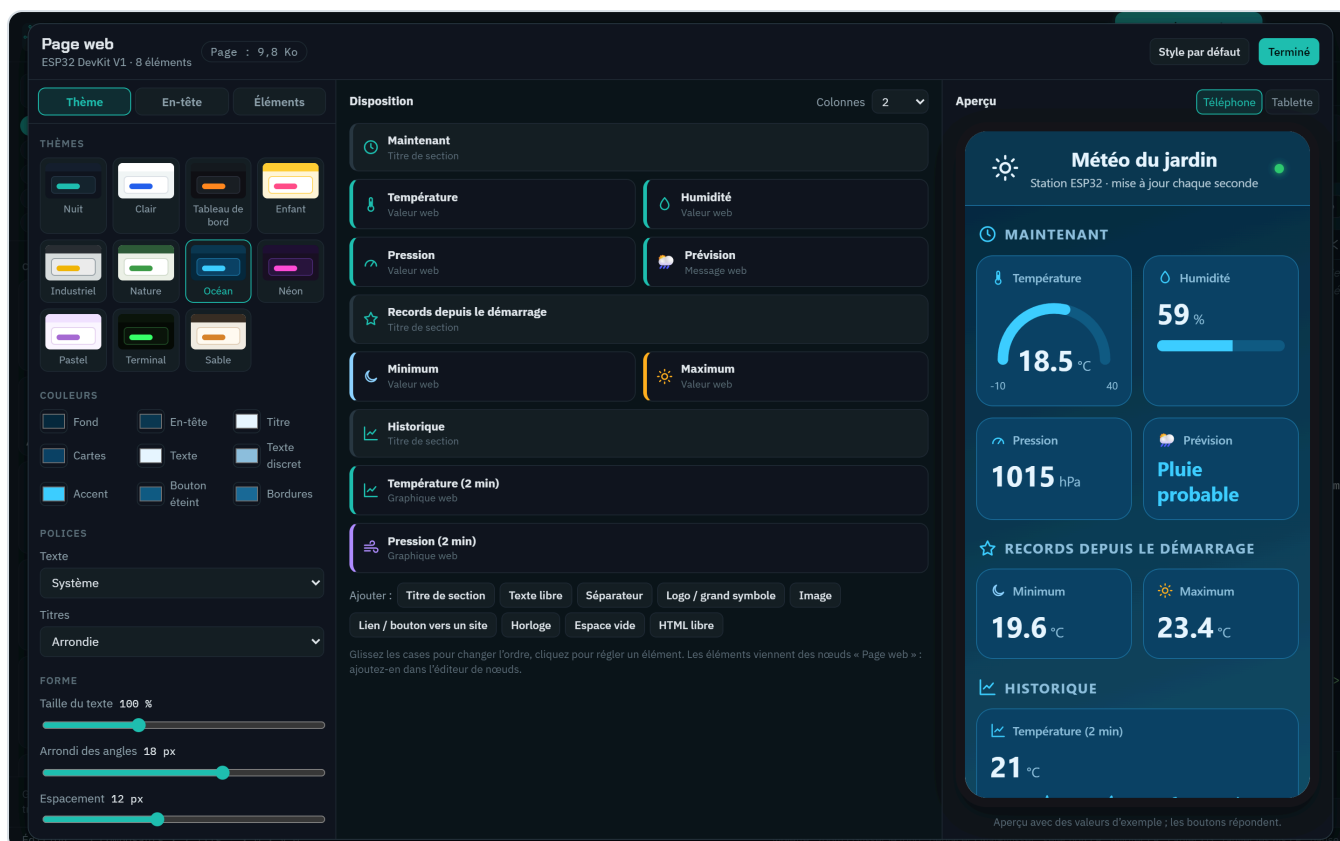


Figure 8.3 — Réglages à gauche, disposition au centre, aperçu en format téléphone à droite.

Onglet Thème

Des thèmes prêts (Nuit, Clair, Tableau de bord, Enfant, Industriel, Nature, Océan, Néon, Pastel, Terminal, Sable...), puis réglage fin : couleurs, police du texte et des titres, taille du texte, arrondi des angles, espacement, ombre, fond dégradé, bordures, largeur maximale.

Onglet En-tête

Titre, sous-titre, symbole, alignement, pastille verte / rouge « connecté », pied de page.

Onglet Éléments

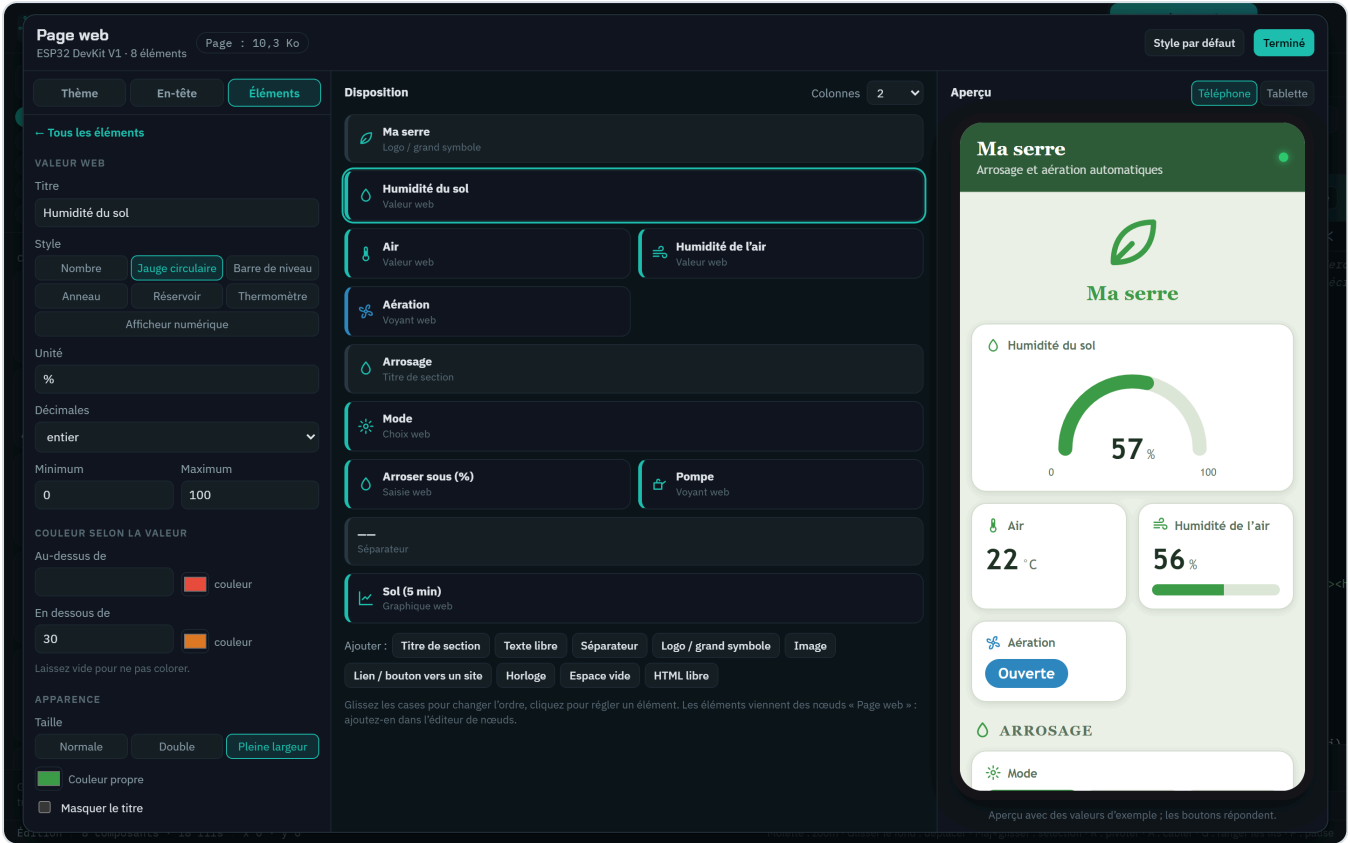


Figure 8.4 — Réglage d’une valeur : jauge circulaire, pleine largeur, orange sous 30 %.

Pour chaque élément : titre, style, taille (normale, double, pleine largeur), couleur propre, titre masqué, symbole (une icône de la collection ou un emoji). Les valeurs et graphiques changent de couleur au-dessus ou en dessous d’un seuil.

BLOC	SUR LA PAGE	STYLES
Bouton web	Interrupteur ou poussoir	Bouton, interrupteur à glissière, gros bouton carré, bouton rond
Curseur web	Réglage d’une valeur	Horizontal, vertical
Valeur web	Affichage d’une mesure	Nombre, jauge circulaire, barre de niveau, anneau, réservoir, thermomètre, afficheur numérique
Voyant web	État allumé / éteint avec texte	Voyant + texte, pastille, ampoule
Message web	L’un de deux textes selon une condition	—
Graphique web	Courbe des dernières mesures	—
Saisie web	Case numérique + bouton	—
Choix web	Boutons à choix unique (Auto / Marche / Arrêt)	—
Couleur web	Palette de couleurs du téléphone	—
Pavé directionnel	Quatre flèches et arrêt, tenus tant qu’on appuie	—

Joystick web	Joystick au doigt : X et Y de –100 à 100, force, direction ; retour au centre au choix, arrêt de sécurité si la liaison coupe	—
Roue chromatique	Choix d'une couleur (rouge, vert, bleu) ; la LED suit le doigt en direct	Roue + luminosité, pastilles
Bouton rotatif	Bouton à tourner entre un minimum et un maximum (aussi à la molette)	—
Compteur + / –	Deux gros boutons qui changent une valeur pas à pas	—
Heure web	Sélecteur d'heure du téléphone : heure, minute	—
Journal web	Liste horodatée des derniers événements	—
Texte web	Affiche un texte quelconque : message reçu, état composé...	—
Alerte web	Bandeau rouge clignotant, bip et vibration tant qu'une condition est vraie	Bandeau, carte seule

Disposition et aperçu

Au centre, choisissez le nombre de colonnes et **glissez les cases** pour changer l'ordre. Ajoutez des décorations : titre de section, texte libre, séparateur, logo, **image** (par son adresse Internet), **lien ou bouton vers un site**, **horloge** du téléphone, **espace vide**, **HTML libre**. À droite, l'aperçu (téléphone ou tablette) utilise des valeurs d'exemple et ses boutons réagissent. La taille de la page est indiquée en haut, avec une alerte si elle devient lourde pour un ESP8266.

Options avancées

En bas de l'onglet Thème : **image de fond**, **opacité des cartes** (en dessous de 100 %, effet verre dépoli sur l'image), **apparition animée** des cartes, et **CSS personnalisé** pour aller plus loin (classes utiles : `.hd` en-tête, `.c` carte, `.v` valeur, `.b` bouton ; couleurs du thème : `var(--ac)`, `var(--card)` ...).

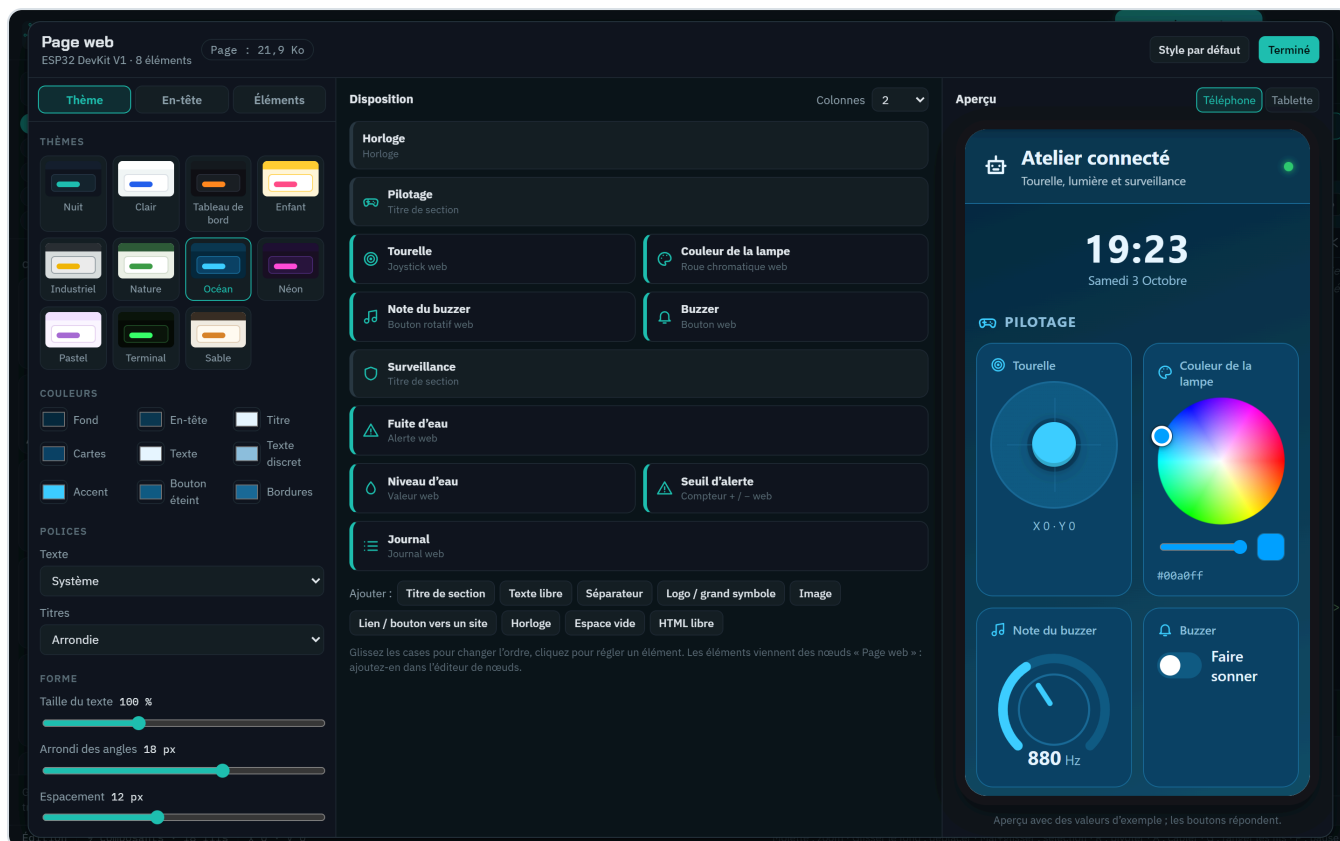


Figure 8.5 — L'exemple « Page web avancée » : horloge, joystick de la tourelle, roue chromatique de la lampe, bouton rotatif du buzzer.

Ce que vous voyez est ce que la carte servira

Les réglages sont enregistrés dans le projet et intégrés au code généré : la vraie carte sert exactement la même page. Relancez la simulation pour voir une modification dans l'onglet Réseau.

La caméra du téléphone

Quatre blocs de la catégorie Page web transforment le téléphone en capteur : la page servie par la carte affiche l'image de l'appareil photo et envoie le résultat à la carte.

BLOC	CE QUE FAIT LA PAGE	CE QUE REÇOIT LA CARTE
Photo du téléphone	Aperçu et bouton « Prendre la photo » ; la photo est convertie en noir et blanc 128 × 64, avec un tramage qui imite les gris	L'image, affichée sur l'écran OLED ; une impulsion « nouvelle photo »
Pipette à couleur (caméra)	Un rond de visée : « Capturer la couleur » mesure la couleur au centre (option : couleurs renforcées pour les LED, mesure en continu)	Rouge, vert, bleu (0–255) et le code #RRVVBB ; la roue chromatique de la page peut prendre la même couleur
Détecteur de mouvement (caméra)	Le téléphone posé compare les images successives	La quantité de mouvement et la luminosité (0–100), « mouvement » au-dessus du seuil

Lecteur de QR code (caméra)

Lit un QR code présenté à la caméra
(Chrome ou Edge sur Android, macOS,
ChromeOS)

Le texte lu et une impulsion «
nouveau code »

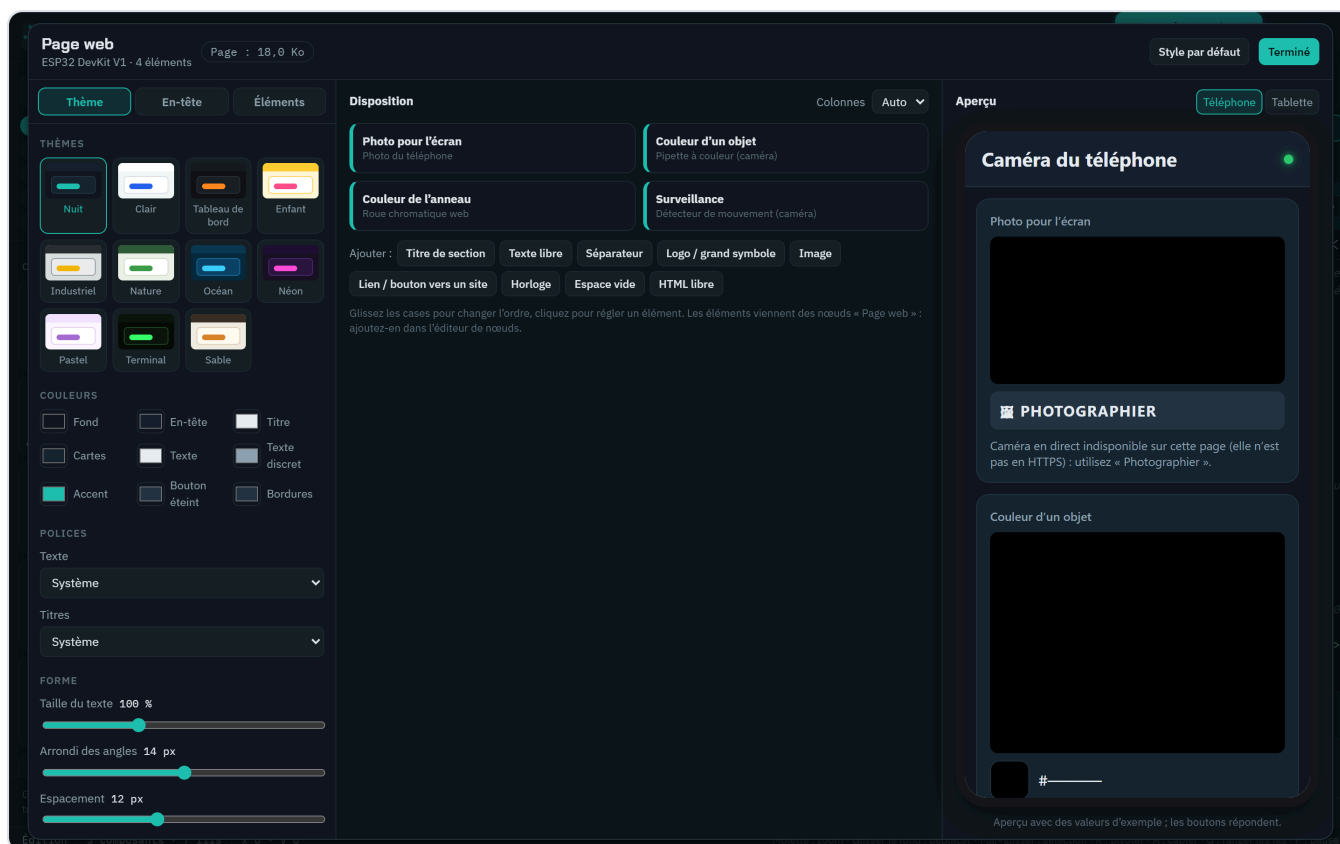


Figure 8.6 — L'exemple « Nœuds : la caméra du téléphone » dans le configurateur : photo pour l'écran OLED, couleur d'un objet, roue chromatique et surveillance.

- **Dans le simulateur**, la webcam de l'ordinateur joue le rôle de la caméra du téléphone ; sans webcam, la page reçoit une image de démonstration (objet coloré, balle qui passe).
- **Sur une vraie carte**, un navigateur n'ouvre l'appareil photo en direct que sur une page **https**. Téléversez avec la case **Page web en https** cochée et installez une fois le certificat sur le téléphone (chapitre 11). En http, le bouton **Photographier** ouvre l'appareil photo pour une seule prise ; le détecteur de mouvement, lui, a besoin du direct.

À essayer

L'exemple « Nœuds : la caméra du téléphone » : photographiez un objet, capturez sa couleur pour l'anneau NeoPixel, puis posez le téléphone en surveillance.

La radio LoRa

LoRa est une radio lente mais qui porte loin : quelques centaines de mètres en ville, plusieurs kilomètres en terrain dégagé, avec très peu d'énergie. Idéale pour une station au fond du jardin, une ruche ou une boîte aux lettres.

- **Câblage** : le module (SX1276 / SX1278, Ra-02, RFM95) se relie en SPI, plus NSS, RST et DIO0. Il s'alimente en **3,3 V**.

- **Propriétés** : bande (433, 868 ou 915 MHz : la même sur tous les modules), **distance** jusqu'aux autres modules et **environnement** (terrain dégagé, ville, à travers des bâtiments). Le signal reçu (RSSI, en dBm) s'affiche près du module ; trop loin, les paquets n'arrivent plus.
- **Blocs** « LoRa : envoyer » et « LoRa : recevoir » transmettent une valeur nommée ; en code, la bibliothèque `LoRa` de Sandeep Mistry.

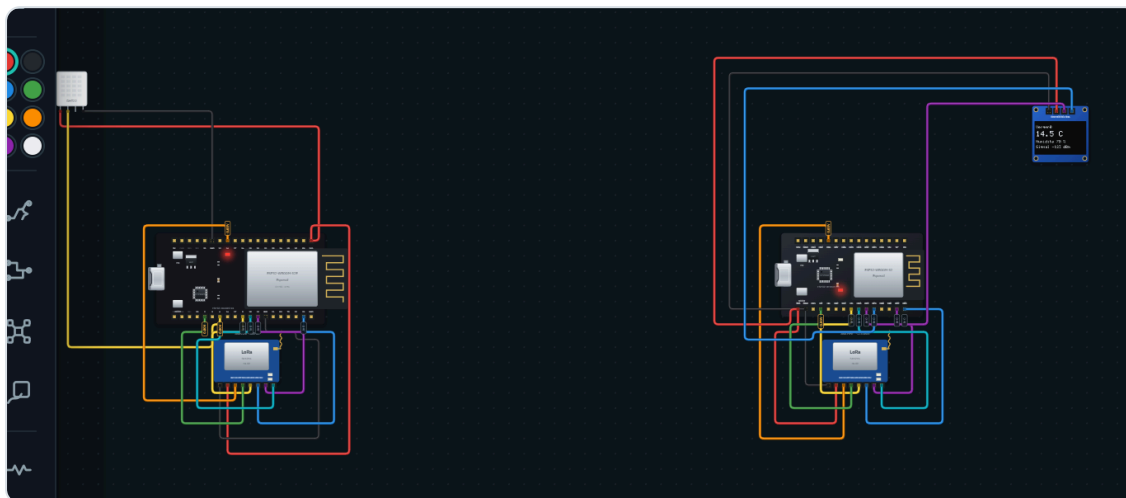


Figure 8.7 — La station météo du jardin : mesure, envoi LoRa, puis veille profonde.

■ Protéger les échanges : le canal sécurisé

Sans précaution, un message radio (ESP-NOW, MQTT, Wi-Fi, LoRa) peut être **lu** par n'importe qui à proximité, **rejoué** (renvoyer l'ordre « ouvrir » capté la veille) ou **falsifié**. Le bloc **Canal sécurisé** (catégorie Sans fil) règle les trois problèmes d'un coup :

- 1 Posez-le sur **chaque carte** qui communique, avec la **même phrase secrète** (au moins 12 caractères, difficile à deviner).
- 2 C'est tout : les blocs Radio directe, MQTT, Entre cartes (Wi-Fi) et LoRa de la carte l'utilisent automatiquement. Chaque message est chiffré (illisible), signé (toute modification est détectée) et numéroté (un vieux message rejoué est refusé), en **AES-128-GCM**, sans bibliothèque à installer.
- 3 Ses sorties « messages refusés » et « attaque détectée » permettent d'allumer une alarme quand quelqu'un tente de tricher.

Jouer le pirate : l'espion radio

Pendant la simulation, la section **Espion radio** de l'onglet Réseau liste chaque message capté dans les airs, avec son heure, le moyen utilisé et l'expéditeur. Un message protégé apparaît « chiffré + signé » : on ne voit qu'un numéro et des octets illisibles.

- **Rejouer** renvoie un message tel quel : c'est l'attaque la plus simple.
- **Falsifier** permet de le modifier avant de l'envoyer ;
Inventer un message... en fabrique un de toutes pièces (par exemple `portail=1`).

Sans canal sécurisé, la carte obéit au pirate. Avec, elle refuse et compte l'attaque.

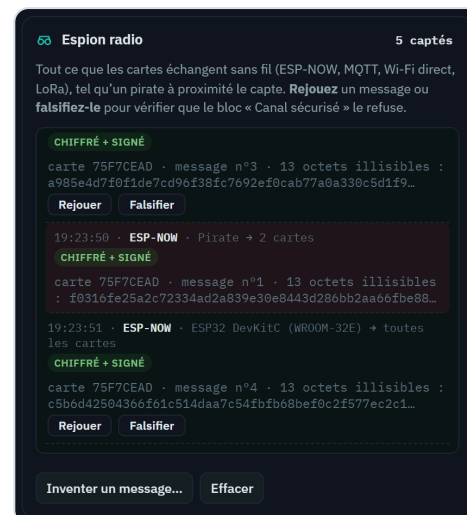


Figure 8.8 — Le portail protégé : un message rejoué par le pirate est refusé.

Les limites, comme en vrai

La phrase secrète est écrite dans le programme : quelqu'un qui a la carte en main peut la lire. Une carte qui redémarre oublie les numéros déjà reçus. Le bloc est prévu pour les cartes ESP32 (il utilise la bibliothèque mbedtls intégrée).

À essayer

L'exemple « Radio protégée : portail commandé sans pirate possible » : lancez la simulation, appuyez sur le bouton, puis rejouez et falsifiez les messages dans l'espion radio.

■ Notifications sur le téléphone

Le bloc **Notification sur le téléphone** envoie une alerte sur votre smartphone quand son entrée « envoyer » devient vraie : intrusion, gamelle vide, serre trop chaude... Il passe par le service gratuit **ntfy.sh** :

- 1 Sur le téléphone, installez l'application gratuite **ntfy** et abonnez-vous à un **sujet**, un nom unique et difficile à deviner (il sert de mot de passe).
- 2 Dans le bloc, indiquez le même sujet, un titre, le message (`{v}` insère une valeur) et l'importance, de « discrète » à « urgente ». Un délai minimum évite d'inonder le téléphone.
- 3 La carte doit être reliée à Internet par la box (bloc « Connexion Wi-Fi » en mode box).

Dans le simulateur, les notifications s'affichent dans la section **Notifications du téléphone** de l'onglet Réseau ; elles partent vraiment vers votre téléphone si l'accès Internet réel est coché.



Figure 8.9 — La gamelle connectée prévient qu'il ne reste presque plus de croquettes.

La veille profonde (ESP32)

Une carte qui ne fait que mesurer de temps en temps peut dormir le reste du temps et tenir des mois sur une batterie. Le bloc **Veille profonde (ESP32)** garde la carte éveillée quelques secondes (le temps de mesurer et d'envoyer), puis l'endort pour la durée choisie. Au réveil, le programme **repart du début**, comme en vrai. Un bouton peut aussi la réveiller (sur une broche compatible). Le moniteur série affiche l'autonomie estimée ; en code, `esp_sleep_enable_timer_wakeup`, `esp_sleep_enable_ext0_wakeup` et `esp_deep_sleep_start` sont pris en charge.

La caméra ESP32-CAM

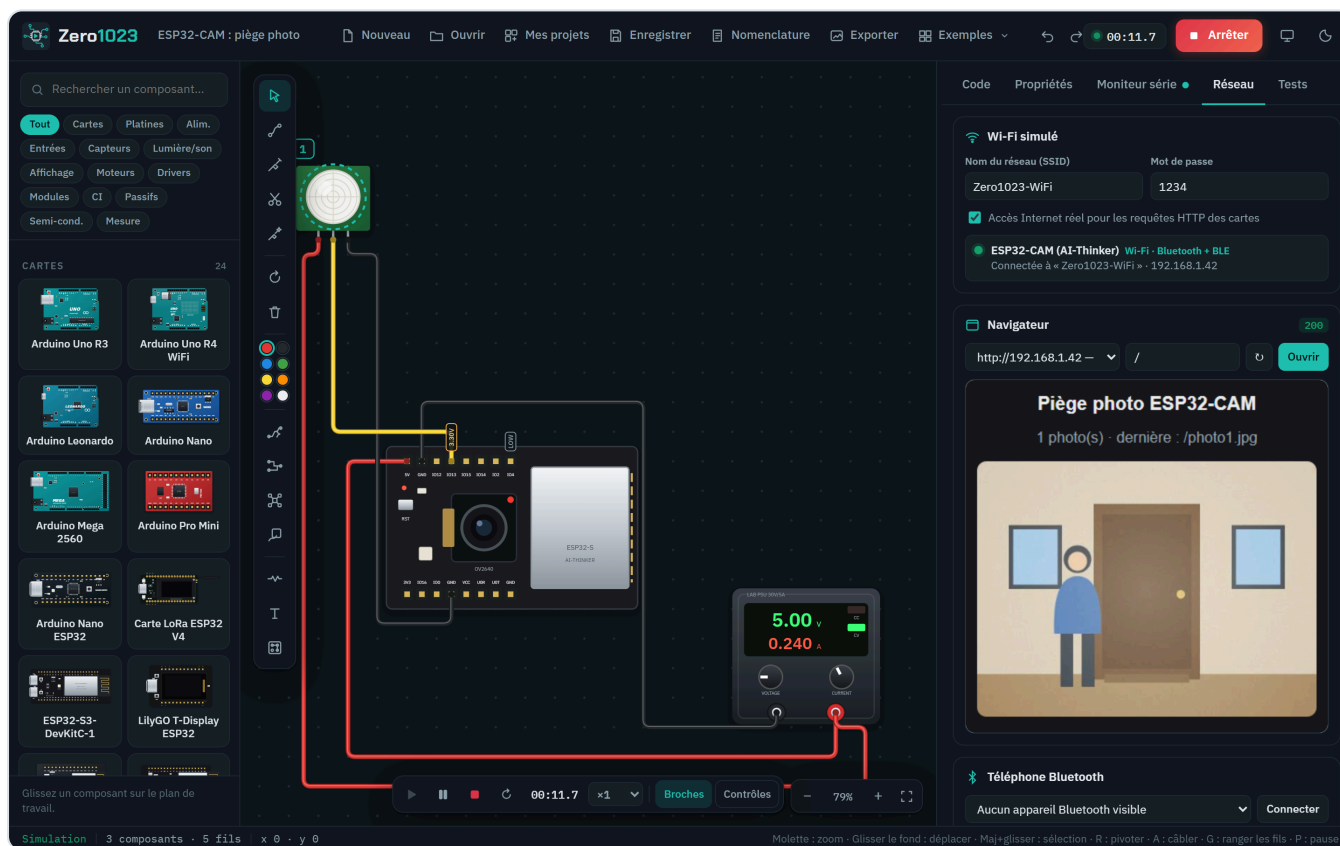


Figure 8.10 — Le piège photo : le détecteur de mouvement a déclenché le flash et une photo, visible sur la page web de la carte.

La carte **ESP32-CAM (AI-Thinker)** a une caméra, une LED flash (IO4) et un lecteur de carte micro-SD. Dans ses propriétés, choisissez la **scène** filmée (salon, entrée, jardin, atelier, mire), la présence d'une **personne** et la **lumière** ; la personne apparaît aussi quand un détecteur de mouvement relié à la carte la voit.

- Le programme utilise les vraies fonctions : `esp_camera_init` , `esp_camera_fb_get` , `esp_camera_fb_return` , et `SD_MMC` pour enregistrer les photos.
- La page web peut servir une image (`send_P` avec `image/jpeg` , `streamFile`) : elle s'affiche dans le navigateur de l'onglet Réseau.
- La carte n'a pas de port USB : en vrai, elle se programme avec un adaptateur USB-série (ou une carte « ESP32-CAM-MB »), et s'alimente en 5 V.

À essayer

L'exemple « ESP32-CAM : piège photo avec page web » et son scénario de test.

Le téléphone GSM (module SIM800L)

Le **SIM800L** est un téléphone portable 2G sans écran : la carte lui parle par une liaison série (TXD du module → RX de la carte, RXD → TX, masses reliées) avec des **commandes AT**. Il fonctionne avec toutes les cartes, sans Wi-Fi.

- **Propriétés du module** : version (carte EVB en 5 V, ou module nu rouge en 3,4–4,4 V), carte SIM présente, opérateur, force du signal, **votre numéro** de téléphone simulé et celui de la carte SIM.
- **Section « Téléphone GSM »** de l'onglet Réseau : envoyez un SMS à la carte (boutons rapides ON, OFF, ETAT), **Appeler la carte** , et lisez les SMS qu'elle vous envoie. Quand la carte vous appelle, un bandeau propose de décrocher ou de refuser.
- **Blocs GSM** : « Module GSM SIM800L » (broches, numéro principal, réseau et signal), « SMS : envoyer » (avec `{v}` pour insérer une valeur, réponse à l'expéditeur, limite anti-rafale), « SMS : recevoir un ordre », « Appel : téléphoner », « Appel : appel entrant » (qui peut raccrocher aussitôt : une télécommande gratuite) et « GPRS : envoyer sur Internet ».
- En code, le module comprend les commandes AT courantes : `AT+CMGF` , `AT+CMGS` , `AT+CNMI` , `AT+CMGR` , `ATD` , `ATA` , `ATH` , `AT+CLIP` , `AT+CSQ` , `AT+CREG?` , `AT+SAPBR` , `AT+HTTPACTION` ...

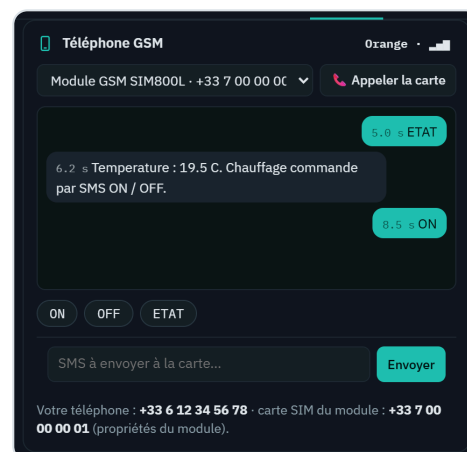


Figure 8.11 — L'alarme GSM : SMS « ETAT », réponse de la carte avec la température, puis SMS « ON » pour le chauffage.

Comme en vrai

Le module demande des pics de 2 A pendant l'émission : alimenté par la broche 3,3 V d'une carte, il redémarre en boucle, et le module nu est détruit en 5 V. Zero1023 vous prévient dans les deux cas. En vrai, vérifiez aussi que votre opérateur a encore un réseau 2G.

À essayer

L'exemple « Alarme GSM : SMS et appels (SIM800L) » : appelez la carte pour armer l'alarme, envoyez ON / OFF / ETAT.

9 Schéma, nomenclature et images

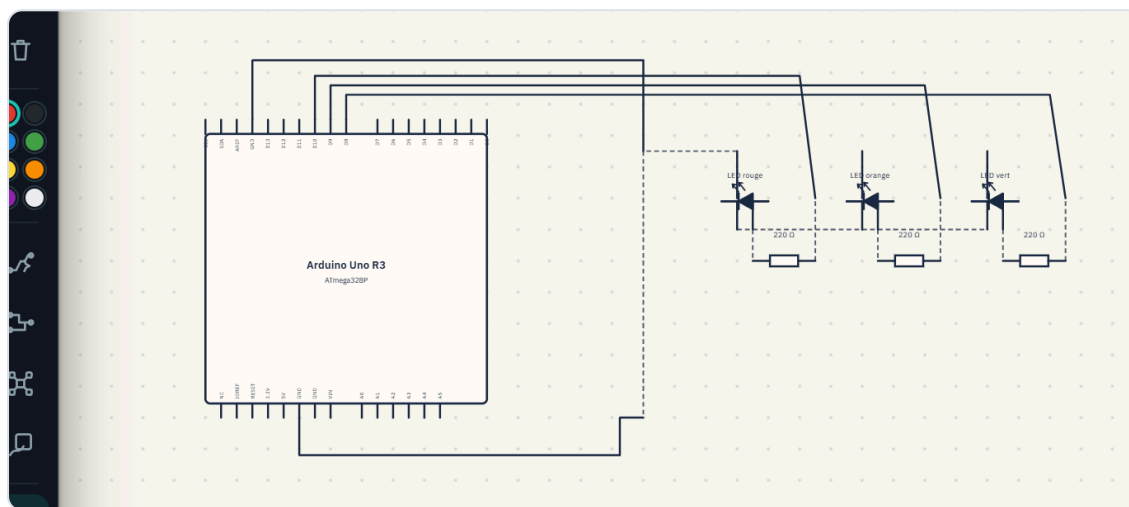
Trois outils pour documenter un montage : le schéma électrique, la liste des composants et l'export en image.

■ Vue schéma

L'outil **Vue schéma** (E) redessine le même montage avec des symboles normalisés : résistances, LED, diodes, transistors, condensateurs... Cartes et modules sont représentés par des blocs à broches.

Les deux vues restent **synchronisées** : un composant ajouté ou déplacé dans l'une l'est aussi dans l'autre. Les liaisons faites par une platine d'essai apparaissent en pointillés.

Appuyez de nouveau sur (E) pour revenir à la vue réaliste. (O) réorganise le montage pour aérer le schéma.



Nomenclature

Nomenclature

Coffre-fort à code · 35 éléments, 10 références

QTÉ	COMPOSANT	VALEUR	REPÈRES	CATÉGORIE
1	Écran LCD 16×2 I²C		MOD1	Affichage
1	Arduino Uno R3		A1	Cartes
1	Clavier matriciel 4×4		MOD2	Entrées
1	Buzzer piézo		BZ1	Lumière et son
1	LED	Vert	D1	Lumière et son
1	LED	Rouge	D2	Lumière et son
1	Servomoteur SG90		M1	Moteurs et actionneurs
2	Résistance	220 Ω	R1, R2	Passifs
1	Platine 400 points		P1	Prototypage
25	Fils de liaison (Dupont / straps)			Câblage

[Exporter en CSV \(tableur\)](#)[Imprimer / PDF](#)[Copier](#)

Figure 9.2 — La liste des composants à acheter, regroupée et repérée (R1, D1, M1...).

Le bouton **Nomenclature** liste tous les composants, regroupés par référence et valeur, avec leurs repères et le nombre de fils. **Exporter en CSV** l'ouvre dans un tableur ; **Imprimer / PDF** et **Copier** servent à la joindre à un dossier.

Exporter une image

Le menu **Exporter** enregistre le montage en **PNG** (normal ou haute définition, pour un diaporama ou une impression) ou en **SVG** (image vectorielle retouchable). Il donne aussi accès à l'historique des versions.

10 Projets, versions et exemples

Un projet contient le montage, le code de chaque carte, les schémas de nœuds, la page web et même les données simulées (EEPROM, fichiers de carte SD).

■ Enregistrer et ouvrir

- **Enregistrer** (**Ctrl** + **S**) télécharge le projet en fichier `.json`. Le nom du projet se modifie en haut à gauche.
- **Ouvrir** recharge un fichier `.json`.
- **Nouveau** repart d'un plan vide.

■ Mes projets

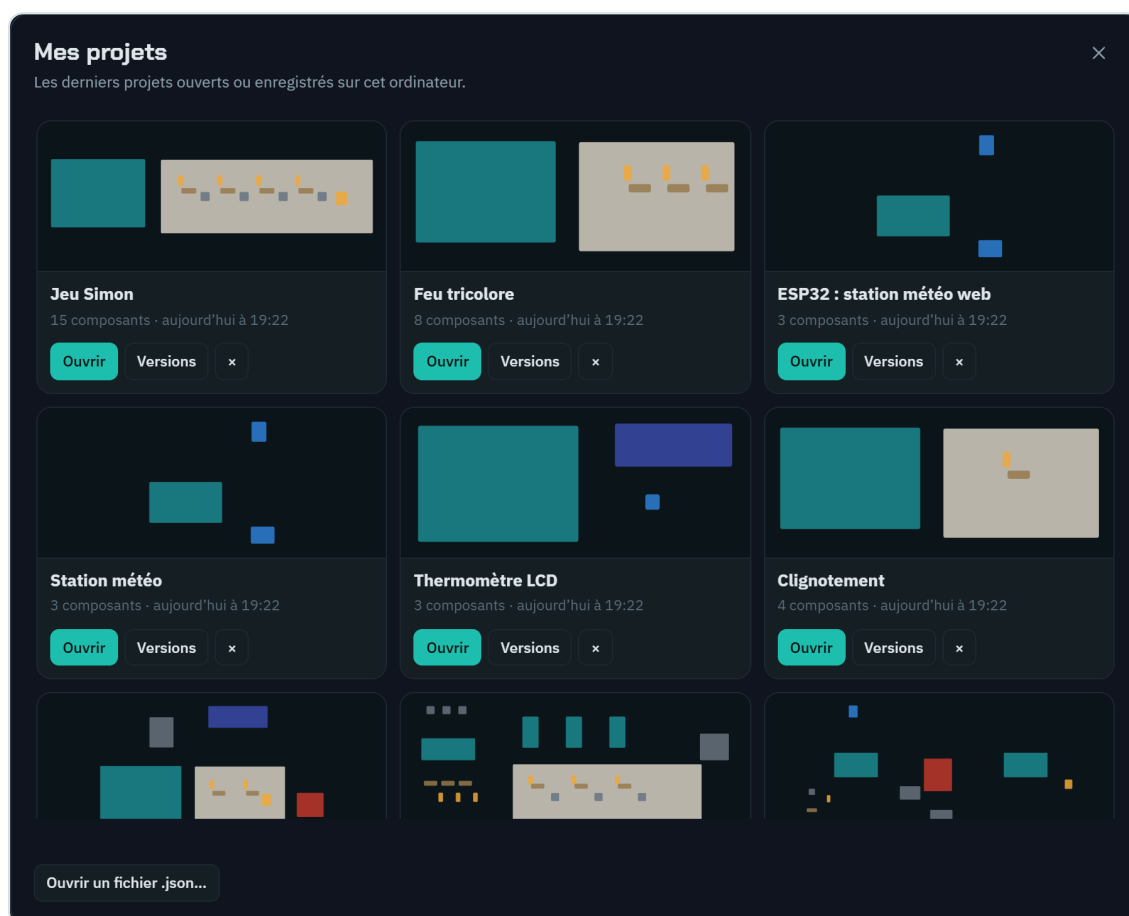


Figure 10.1 — Les derniers projets, avec une vignette du montage.

Les derniers projets ouverts ou enregistrés sont gardés avec une vignette. **Ouvrir** les recharge, **Versions** montre leur historique, **x** les retire de la liste.

■ Historique des versions

Une version est gardée automatiquement toutes les 5 minutes de travail, à chaque enregistrement et avant chaque restauration. Pour revenir en arrière : **Exporter** → **Historique des versions**, ou **Versions** dans Mes projets, puis **Restaurer**.

Une sauvegarde de secours, pas un coffre-fort

Mes projets et les versions sont conservés par le navigateur de cet ordinateur. Vider les données du navigateur les efface : enregistrez vos projets importants en `.json`.

La bibliothèque d'exemples

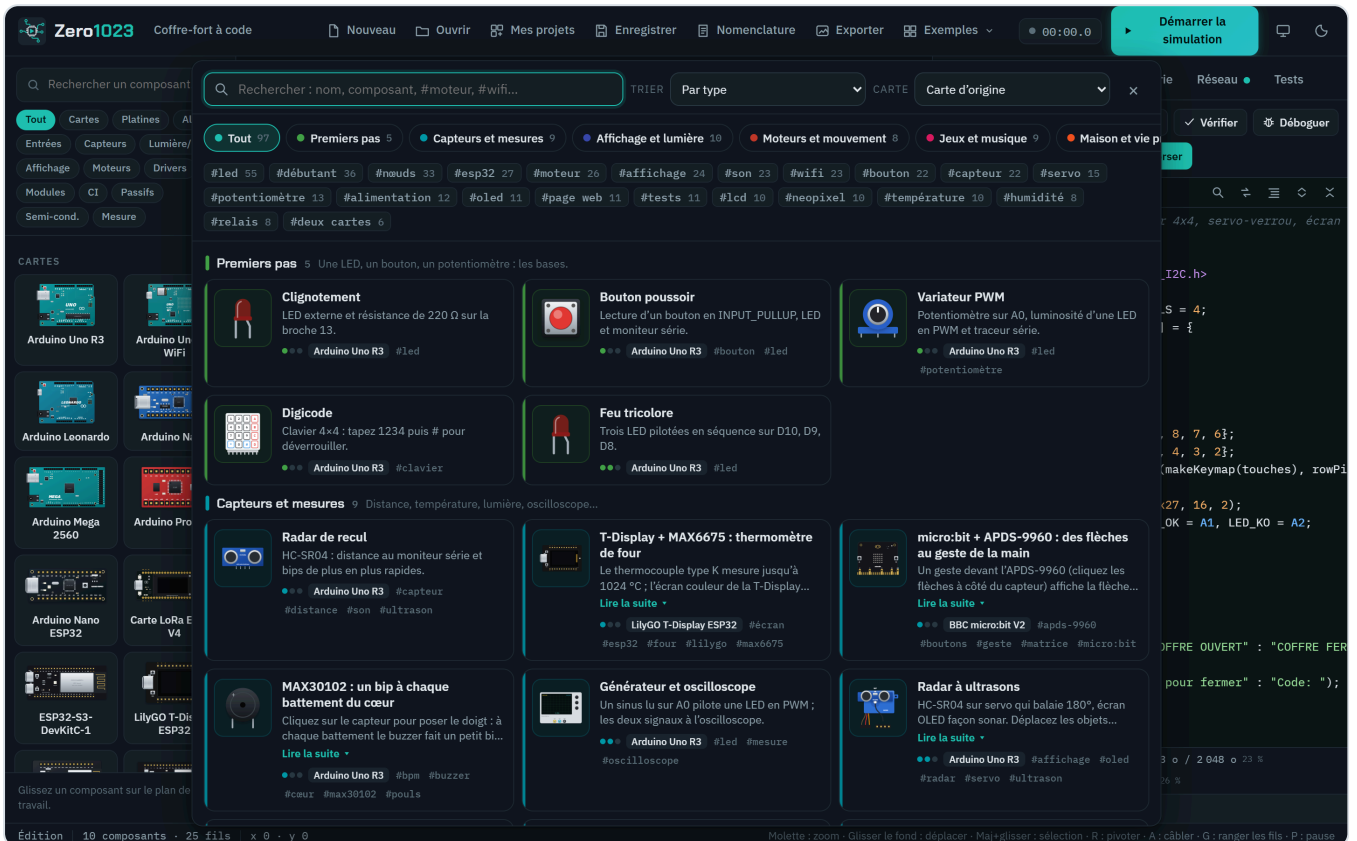


Figure 10.2 — Les exemples classés par thème, avec recherche et mots-clés.

Le menu **Exemples** ouvre la bibliothèque : recherche par nom ou composant, mots-clés (`#moteur`, `#wifi`, `#configurateur` ...), catégories (Premiers pas, Capteurs, Affichage, Moteurs, Jeux, Maison, Sans fil, Nœuds, Électronique), tri, et choix de la carte : un exemple prévu pour une Uno peut être chargé sur une autre carte compatible.

- Les **trois points** en bas de chaque carte indiquent le niveau : ●○○ débutant, ●●○ intermédiaire, ●●● avancé (d'après le nombre de composants et la longueur du programme). Le tri « Du plus simple au plus avancé » les utilise.
- Quand un descriptif est trop long, « **Lire la suite** ▼ » l'affiche en entier sans ouvrir l'exemple ; « Réduire ▲ » le replie. Cliquer ailleurs sur la carte ouvre l'exemple.

11 Téléverser sur une vraie carte

Un navigateur ne peut ni compiler du C++ ni accéder librement aux ports USB. Le petit « pont » Zero1023 s'en charge avec `arduino-cli`, l'outil officiel d'Arduino.

■ Installer le pont (une seule fois, Windows)

- 1 Récupérez le pont : la fenêtre **Téléverser** propose de le télécharger (fichier `zero1023-pont.zip`) tant qu'il n'est pas détecté. Décompressez-le.
- 2 Double-cliquez sur **Installer le pont.bat**. Il installe **Node.js** et **arduino-cli** s'ils manquent (Windows peut demander l'autorisation), copie le pont dans `%LOCALAPPDATA%\Zero1023\Pont` et l'ajoute au démarrage de Windows.
- 3 C'est tout : le pont tourne en arrière-plan, sans fenêtre, à chaque ouverture de session. Pour le **mettre à jour**, relancez simplement l'installateur d'une version plus récente.
- 4 Pour les cartes à puce USB **CH340** (Nano, Keystudio ESP32 PLUS...), installez si besoin le pilote CH340 : sans lui, aucun port n'apparaît.

Désinstaller, ou utiliser un Mac ou Linux

Désinstaller le pont.bat l'arrête et le retire du démarrage. Sans l'installateur (macOS, Linux) : installez Node.js et `arduino-cli` (`brew install arduino-cli`), puis lancez `node zero1023-bridge.js` et laissez la fenêtre ouverte. En cas de souci, le journal du pont est dans `%LOCALAPPDATA%\Zero1023\Pont\pont.log`.

■ Téléverser

- 1 Branchez la carte en USB.
- 2 Dans Zero1023, onglet Code, cliquez sur **Téléverser**. La pastille verte « Pont connecté » confirme que le pont est détecté.
- 3 Choisissez le **type de carte** et le **port USB** (bouton $\cup$ pour actualiser la liste). Un bandeau rappelle les particularités de la carte (bouton BOOT d'un ESP32, BOOTSEL d'un Pico...).
- 4 **Compiler seulement** vérifie le programme ; **Téléverser** compile et envoie.

Au premier envoi pour une famille de cartes (AVR, ESP32, ESP8266, RP2040...), le pont télécharge le cœur correspondant : cela peut prendre quelques minutes. Les bibliothèques utilisées (`#include`) sont installées automatiquement.

Moniteur série de la carte

En bas de la fenêtre, **Connecter** affiche ce que la vraie carte envoie, à la vitesse choisie, et permet de lui écrire. Il fonctionne **par le pont**, ou **par le navigateur** sans pont (Chrome, Edge : choisissez le port dans la liste proposée par le navigateur).

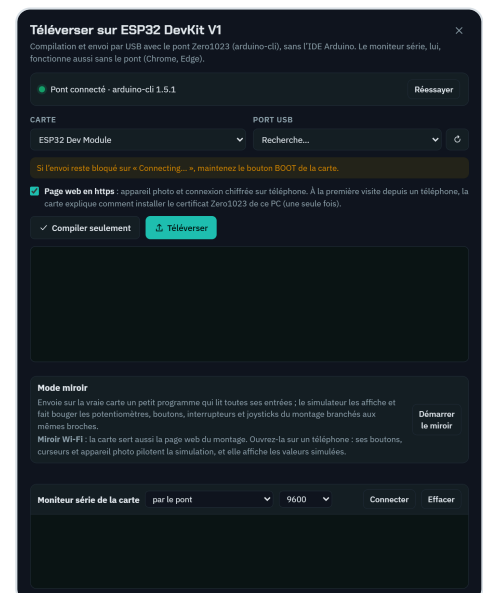


Figure 11.1 — La fenêtre de téléversement, avec la case « Page web en https » et le mode miroir.

■ Pages web en https sur le téléphone

Une page servie en simple **http** fonctionne, mais un téléphone y refuse l'appareil photo en direct et signale la connexion comme « non sécurisée ». Pour les cartes **ESP32** et **ESP8266**, le pont passe automatiquement la page en **https** au téléversement (case **Page web en https**, cochée par défaut). Il suffit d'installer **une fois**, sur chaque téléphone, le certificat Zero1023 de votre ordinateur.

Protocole

- 1 Branchez l'ordinateur au **même réseau** (box ou point d'accès) que la carte et le téléphone : le certificat couvre les adresses du réseau de l'ordinateur au moment du téléversement.
- 2 Téléversez le programme avec la case **Page web en https** cochée. Le journal confirme « Page web en https ».
- 3 Sur le téléphone, ouvrez l'adresse affichée par la carte dans le moniteur série, en **http://** (par exemple `http://192.168.1.42`). La première fois, la carte affiche la page « Première visite depuis ce téléphone ».
- 4 Choisissez l'onglet de votre téléphone, touchez **Télécharger le certificat** et suivez les étapes (voir ci-dessous).
- 5 Revenez sur la page et touchez **Ouvrir la page sécurisée (https)**. Les fois suivantes, l'adresse http bascule toute seule en https.

Première visite depuis ce téléphone

Pour ouvrir la page en https (appareil photo, connexion chiffrée), installez une seule fois le certificat **Zero1023 v2** (PC-SALON). Il servira ensuite pour toutes les cartes programmées depuis ce PC.

iPhone / iPad

Android

1. Dans **Safari**, touchez « Télécharger le certificat » puis **Autoriser**.
2. Ouvrez **Réglages** → **Profil téléchargé** (en haut) → **Installer**.
3. **Réglages** → **Général** → **Informations** → **Réglages des certificats** (tout en bas) : activez **Zero1023 v2 (PC-SALON)**. Un ancien « Zero1023 » sans « v2 » peut être supprimé (**Réglages** → **Général** → **VPN et gestion de l'appareil**).
4. Revenez ici et touchez **Ouvrir la page sécurisée**.

Télécharger le certificat

Ouvrir la page sécurisée (https)

Si le navigateur affiche « Cette connexion n'est pas privée », le certificat n'est pas encore approuvé : vérifiez l'étape 3, fermez complètement le navigateur (balayez-le vers le haut), rouvrez-le et revenez sur cette adresse.

Continuer sans https (sans appareil photo)

Ce certificat est propre à votre PC : seules les cartes programmées depuis ce PC l'utilisent, et sa clé secrète ne quitte jamais l'ordinateur.

Figure 11.2 — La page d'aide servie par la carte à la première visite (ici sur iPhone).

iPhone / iPad

1. Dans **Safari**, touchez « Télécharger le certificat » puis **Autoriser**.
2. **Réglages** → **Profil téléchargé** (en haut) → **Installer**.
3. **Réglages** → **Général** → **Informations** → **Réglages des certificats** (tout en bas) : activez **Zero1023 v2 (nom du PC)**. Cette étape est indispensable : sans elle, la page sécurisée reste refusée.
4. Revenez sur la page et touchez « Ouvrir la page sécurisée ».

Android

1. Touchez « Télécharger le certificat » (fichier `zero1023.crt`).
2. Ouvrez **Paramètres** et cherchez « **certificat CA** » (ou : **Sécurité** → **Chiffrement et identifiants** → **Installer un certificat** → **Certificat CA**), touchez **Installer quand même** et choisissez `zero1023.crt` .
3. Revenez sur la page et touchez « Ouvrir la page sécurisée ».

« Cette connexion n'est pas privée »

Le certificat n'est pas encore approuvé : sur iPhone, vérifiez l'étape « Réglages des certificats ». Fermez complètement le navigateur (balayez-le vers le haut), rouvrez-le et revenez sur l'adresse. Si la page signale que l'adresse n'est pas couverte, téléversez à nouveau depuis un ordinateur branché au même réseau que la carte.

Un certificat par ordinateur

Le certificat est créé une fois pour votre ordinateur : seules les cartes programmées depuis cet ordinateur l'utilisent. Sa clé secrète reste dans %LOCALAPPDATA%\Zero1023\Certificats : ne partagez jamais ce dossier. Un téléphone qui travaille avec plusieurs ordinateurs installe le certificat de chacun. L'Uno R4 WiFi et les autres cartes restent en http ; décochez la case pour revenir en http sur un ESP32.

■ Le mode miroir

Le mode miroir fait piloter la simulation par la vraie carte : on teste un programme simulé avec les vrais boutons en main.

- 1 Câblez les potentiomètres, boutons, interrupteurs ou joysticks sur la vraie carte, sur les **mêmes broches** que dans le montage simulé.
- 2 Ouvrez **Téléverser**, choisissez la carte et le port, puis **Démarrer le miroir**. Un petit programme est envoyé sur la carte : il lit toutes ses entrées, sans rien piloter (aucun risque pour le montage).
- 3 Un panneau **Miroir** s'ouvre sur le plan de travail : il affiche les lectures en direct, et les composants simulés branchés aux mêmes broches bougent avec les vrais.
- 4 Le **x** du panneau (ou un nouveau clic sur **Démarrer le miroir**) termine le mode miroir. Téléversez ensuite votre programme pour que la carte fonctionne seule.

■ Le miroir Wi-Fi : piloter la simulation depuis le téléphone

Avec une carte **ESP32** ou **ESP8266** dont le programme en blocs a une **page web**, le miroir devient un **miroir Wi-Fi** : la vraie carte sert la page web du montage, et ce que vous touchez sur le téléphone (boutons, curseurs, roue chromatique, appareil photo...) pilote la simulation ; la page affiche en retour les valeurs simulées.

- 1 Programmez le montage en blocs avec des blocs **Page web** et un bloc **Connexion Wi-Fi** : en mode « propre réseau », la carte crée son Wi-Fi ; en mode box, elle rejoint votre box (nom et mot de passe de la vraie box).
- 2 Branchez la carte en USB, ouvrez **Téléverser** : la mention **Miroir Wi-Fi** apparaît sous le mode miroir. Laissez cochée **Page web en https** si le téléphone doit utiliser l'appareil photo.
- 3 Cliquez sur **Démarrer le miroir**. Le programme miroir est envoyé et la simulation démarre toute seule.
- 4 Le panneau **Miroir Wi-Fi** indique le réseau à rejoindre (et son mot de passe), l'adresse de la carte et un **QR code** : scannez-le avec le téléphone, ou tapez l'adresse.
- 5 Touchez les commandes de la page : chaque action est transmise par la carte, puis par le câble USB, à la simulation. Le panneau compte les **actions reçues du téléphone**.



Figure 11.3 — Le panneau du miroir Wi-Fi : réseau, adresse, QR code et état de la liaison.

- Le câble USB doit rester branché : c'est lui qui relie la vraie carte au simulateur. Si la liaison est coupée (carte redémarrée, câble débranché), elle se rétablit d'elle-même ; le panneau affiche « Liaison USB avec la carte :

active » ou « coupée, reconnexion... ».

- Si la simulation est arrêtée, la prochaine action du téléphone la relance ; les actions reçues entre-temps sont gardées et rejouées.
- Un point d'accès exige un mot de passe d'au moins 8 caractères : plus court, la carte crée un réseau ouvert (le panneau l'indique).
- La page passe en https si le téléphone a le certificat Zero1023 (voir plus haut) ; sinon elle explique comment l'installer.

Sécurité

Le pont n'écoute que sur cet ordinateur (adresse 127.0.0.1, port 8787) et n'accepte que les requêtes de Zero1023, ouvert en local ou depuis son site officiel.

12 Personnaliser l'affichage

Le bouton en forme d'écran, en haut à droite, ouvre le menu Affichage ; celui en forme de lune bascule entre thème sombre et clair.

- **Plein écran** (**Ctrl** + **Maj** + **F**) : masque le navigateur et la barre des tâches.
- **Mode concentration** (**M**) : plan de travail seul, sans bibliothèque ni panneau.
- **Panneau dans une autre fenêtre** : code, moniteur série et réseau sur un deuxième écran, le plan de travail prend toute la place.
- **Vue présentation** : le montage en direct, en grand, dans une fenêtre à placer sur un vidéoprojecteur.
- **Style graphique** : Classique, Plan d'architecte ou Néon rétro.
- **Grille** du plan de travail : selon le style, points, lignes ou aucune.
- **Langue** : français ou anglais. En anglais, tout est traduit : interface, messages, composants, blocs, exemples, aides, référence Arduino et commentaires du code généré. Zero1023 s'ouvre en anglais quand le navigateur n'est pas réglé en français ; le choix est ensuite mémorisé.

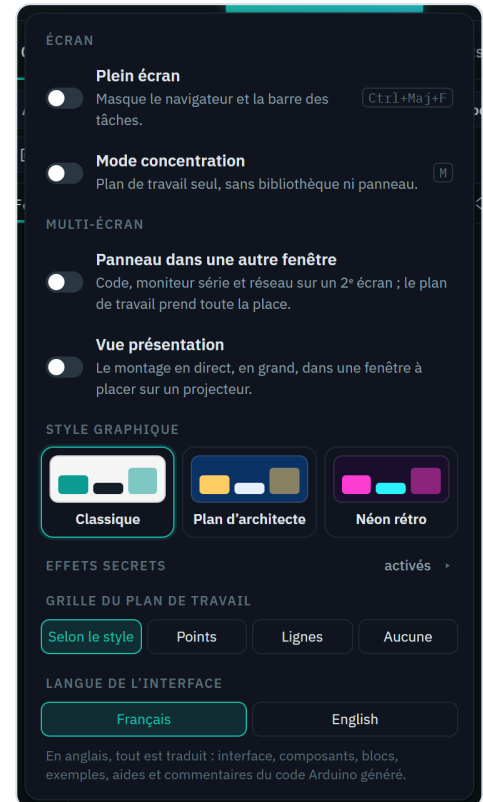


Figure 12.1 — Le menu Affichage.

Le compagnon

Un petit chat noir en pixel art, **Réglisse**, vit sur le plan de travail. Il se promène, fait sa toilette, s'étire, fait la sieste quand vous ne touchez à rien, et de temps en temps une bêtise : un coup de patte dans un composant, une tasse poussée hors du bord, une pelote qui roule. Ses bêtises sont purement visuelles : il ne modifie **jamais** le montage ni le code.

- **Caressez-le** d'un clic : il ronronne. **Portez-le** en le faisant glisser : il se balance au bout de votre souris, puis se réceptionne en douceur.
- Il est posé **sur le plan** : quand vous déplacez la vue ou zoomez, il suit.
- Il vit avec vous : il suit des yeux le composant que vous déplacez et vient le renifler, sursaute à un clic trop proche, jette un œil au code que vous tapez, réagit au démarrage de la simulation et aux erreurs.

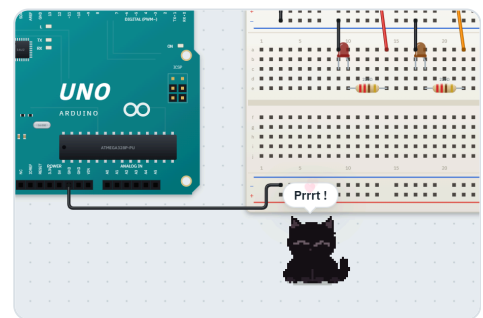


Figure 12.2 — Réglisse, caressé, à côté du montage.

- Son pelage s'ébouriffe quand il a peur ou qu'il court, et redevient lisse après sa toilette.

Ses réglages

Dans le menu Affichage, la section repliable **Compagnon** l'active ou le désactive et règle son **nom**, sa **taille** (petit par défaut), sa **robe** (noir, noir et blanc, gris, tigré, roux, crème), ses **yeux**, son **collier** (aucun par défaut), son **pelage** (selon son humeur, toujours lisse ou toujours ébouriffé), son **caractère** (calme, normal, joueur) et la fréquence de ses **bêtises**. [Faire une bêtise maintenant](#) le lance aussitôt.

L'aperçu animé montre le chat assis, puis en train de marcher de profil, de dos et de face, avec les réglages choisis.

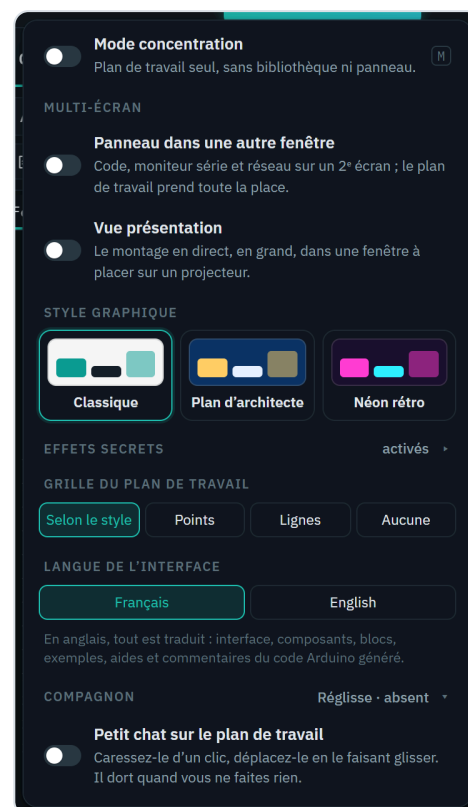


Figure 12.3 — Les réglages du compagnon, dans le menu Affichage.

13 Raccourcis clavier

Les raccourcis à une lettre fonctionnent quand le curseur n'est pas dans un champ de texte ou dans l'éditeur de code.

Plan de travail

V	Sélection
W	Fil
S	Fer à souder
X	Coupe de piste / dessoudage
R	Pivoter la sélection
Suppr	Supprimer la sélection
A	Câblage automatique
G	Ranger les fils
O	Organiser le montage
B	Fils en arrière-plan
L	Changer de face de plaque
E	Vue schéma
T	Étiquette de texte
F	Tout afficher
+ / -	Zoomer / dézoomer
Espace + glisser	Déplacer la vue
Échap	Annuler le fil en cours, désélectionner

Débogueur

F8	Continuer jusqu'au prochain point d'arrêt
F10	Ligne suivante
F9	Mettre en pause ici

Général

Ctrl + Z	Annuler
Ctrl + Y	Rétablir
Ctrl + D	Dupliquer
Ctrl + A	Tout sélectionner
Ctrl + S	Enregistrer
Maj + Entrée	Démarrer / arrêter la simulation
P	Pause / reprise
N	Éditeur de nœuds
M	Mode concentration
Ctrl + Maj + F	Plein écran

Éditeur de code

Ctrl + Espace	Autocomplétion
Maj + Alt + F	Mettre en forme
Ctrl + F	Rechercher
Ctrl + H	Remplacer
Ctrl + Q	Plier / déplier
Maj + Tab	Désindenter

Pendant la simulation

1 ... 9	Boutons et interrupteurs
A / Q , Z / S ...	Potentiomètres (un sens par touche)
Flèches	Joystick, inclinaison MPU6050

14 Questions fréquentes et limites

Les situations les plus courantes, et ce qu'il faut savoir sur les différences entre simulation et réalité.

■ Questions fréquentes

La simulation ne démarre pas, une ligne est surlignée en rouge.

Le programme contient une erreur : lisez le message sous l'éditeur. **Vérifier** la localise sans lancer la simulation, et **Référence** indique si une fonction est prise en charge.

Ma LED ne s'allume pas.

Vérifiez le sens (patte longue, l'anode, vers la broche), la résistance, et que la masse est bien reliée. Activez **Broches** pour voir l'état de la broche, et survolez la broche pour lire sa tension.

Le moniteur série affiche des caractères bizarres.

Entre deux cartes, les vitesses de la liaison série ne correspondent pas : utilisez la même valeur dans `begin()` des deux côtés.

La page web n'apparaît pas dans l'onglet Réseau.

La carte doit être connectée au Wi-Fi (nom et mot de passe identiques à ceux de l'onglet Réseau), et le programme doit appeler `server.begin()` puis `server.handleClient()` dans `loop()`. Avec les blocs, ajoutez un bloc « Connexion Wi-Fi ».

Le téléversement propose seulement « Autre carte (FQBN) ».

Vérifiez que le pont est lancé et à jour, puis choisissez la carte dans la liste. Pour une carte inconnue, indiquez son identifiant arduino-cli (par exemple `esp32:esp32:esp32`).

Le module GSM ne répond pas.

Vérifiez l'alimentation (la section « Téléphone GSM » indique « hors tension »), que TXD du module va bien sur la broche RX de la carte, et que la carte SIM est insérée avec du signal (propriétés du module). Le module met environ 3 secondes à s'inscrire sur le réseau après la mise sous tension.

Un test échoue alors que le montage marche.

Survolez le X pour lire la valeur mesurée. Souvent, le contrôle est trop tôt : laissez à la carte le temps de démarrer (Wi-Fi, réseau GSM) et décalez l'instant du contrôle.

La notification n'arrive pas sur mon téléphone.

Dans le simulateur, elle apparaît dans l'onglet Réseau ; elle ne part vers le téléphone que si « Accès Internet réel » est coché. Vérifiez que l'application ntfy est abonnée exactement au même sujet, et que la carte est connectée à la box (pas en point d'accès).

Mes cartes LoRa ne se reçoivent pas.

Les modules doivent utiliser la même bande (433, 868 ou 915 MHz). Réduisez la distance ou passez en terrain dégagé dans leurs propriétés : le RSSI affiché près du module montre la force du signal. Avec un canal sécurisé, vérifiez que la phrase secrète est identique sur toutes les cartes.

Sur l'iPhone, « Ouvrir la page sécurisée » me ramène à la page du certificat.

Le certificat est installé mais pas encore **approuvé** : Réglages → Général → Informations → Réglages des certificats, activez « Zero1023 v2 ». Fermez ensuite complètement Safari et revenez sur l'adresse en http.

En miroir Wi-Fi, mes actions sur le téléphone ne font rien.

Regardez le panneau Miroir Wi-Fi : la liaison USB doit être « active » et le compteur d'actions doit augmenter quand vous touchez la page. Si le compteur reste à 0, le téléphone n'est pas sur le bon Wi-Fi ou la page n'est pas celle de la carte (rechargez-la). Si la liaison est coupée, vérifiez le câble et fermez tout autre moniteur série qui utiliserait le port (IDE Arduino, autre onglet).

Le téléversement ne trouve pas le pont.

Relancez **Installer le pont.bat** : il redémarre le pont et le met à jour. Le journal

%LOCALAPPDATA%\Zero1023\Pont\pont.log indique la cause d'une erreur.

J'ai perdu mon projet.

Regardez dans **Mes projets** et dans l'historique des versions. À l'avenir, enregistrez régulièrement avec **Ctrl + S**.

■ Limites connues

- La simulation reproduit le comportement logique et électrique courant (tensions, courants, protections) mais pas tous les phénomènes physiques : parasites, rebonds de contacts très rapides, échauffement, précision exacte des horloges.
- Les bibliothèques simulées couvrent les fonctions les plus utilisées ; certaines fonctions avancées manquent. La **Référence** fait foi.
- Les polices spéciales des écrans graphiques (FreeSans...) sont rendues de façon approchée.
- Le courtier MQTT est intégré au simulateur ; les requêtes HTTP vers Internet dépendent de ce que le site autorise depuis un navigateur.
- Le mode miroir et le téléversement demandent le pont Zero1023 et arduino-cli installés sur l'ordinateur.
- La mémoire affichée est une estimation ; le débogueur montre les variables globales (pas les variables locales des fonctions).
- Les SMS et appels restent dans le simulateur ; les requêtes GPRS ne partent vraiment que si « Accès Internet réel » est coché. Les images de l'ESP32-CAM sont des scènes dessinées, pas une vraie caméra.
- La radio directe ESP-NOW, le canal sécurisé et la veille profonde sont réservés aux cartes ESP32 ; le montage automatique ne câble pas l'intérieur des blocs réutilisables.
- La portée LoRa suit un modèle simplifié (distance et environnement) : en vrai, le relief, les antennes et la météo comptent aussi.
- Dans le simulateur, la caméra du téléphone est remplacée par la webcam de l'ordinateur (ou une image de démonstration). Les pages en https et le miroir Wi-Fi concernent les cartes ESP32 et ESP8266.
- Dans la version découverte, les éléments marqués Pro ne sont pas utilisables (chapitre 1).