

Mesure de la qualité de l'air avec le RP2040

Claus Kühnel (Allemagne)

Début 2021, la Fondation Raspberry annonçait le Raspberry Pi Pico. Les lecteurs du magazine Elektor le savent, il ne s'agit pas d'un autre modèle de la famille RPi, mais d'une carte indépendante dotée du microcontrôleur RP2040. On peut l'utiliser pour mesurer la qualité de l'air. Les données sont acquises à partir d'un capteur de CO₂ puis transmises via un ESP8266 à ThingSpeak, la fameuse plateforme du nuage.

Même après l'annonce du RPi Pico, de nombreuses discussions controversées ont eu lieu sur l'Internet. Alors que certains accueillaient le nouveau microcontrôleur avec enthousiasme, d'autres se focalisaient immédiatement sur son manque de fonctionnalité. Elektor a essayé de donner un aperçu plus neutre dans [1] mais à l'époque le jugement était souvent très bipolaire, et il n'y avait guère de place pour des opinions différenciées.

Entre-temps, Arduino introduisait sur le marché l'Arduino Nano RP2040 Connect. Le microcontrôleur RP2040 amélioré par un module radio ublox NINA-W102 permet de constituer un dispositif IdO complet avec une connectivité Wifi et BLE. Si vous utilisez l'Arduino Nano RP2040 Connect dans un projet, vous utiliserez probablement aussi les EDI Arduino ou PlatformIO. Mais aucun des deux ne prend en charge

certaines spécificités du RP2040, comme le double cœur et le PIO. Autrement, avec le RPi Pico seul, vous avez le choix entre les SDK Raspberry Pi Pico C/C++ et Raspberry Pi Pico Python, qui prennent tous deux en charge toutes les fonctionnalités de ce microcontrôleur. En revanche, c'est la prise en charge de toute communication sans fil qui vous fera alors défaut. Il existe néanmoins suffisamment de modules de communication disponibles aujourd'hui pour ajouter cette fonctionnalité.

Grâce à la prise en charge de multiples microcontrôleurs et des nombreuses bibliothèques disponibles, j'utilise souvent l'EDI Arduino pour mes projets. La nouvelle version majeure de l'EDI Arduino est plus rapide et encore plus puissante. L'Arduino EDI 2.0 est en train de passer au statut stable avec une série de versions préliminaires (RC). En avril 2022, la version RC 2.0.0-rc6 de l'EDI Arduino 2.0 était disponible, téléchargeable à l'adresse [2]. En plus d'un éditeur plus moderne et d'une interface plus réactive, il propose l'autocomplétion, la navigation dans le code, et même un débogueur en direct.

Raspberry Pi Pico

Le microcontrôleur RPi Pico ne sera abordé que brièvement car il a déjà fait l'objet de nombreuses évaluations et la plupart des lecteurs le connaissent. Je n'entrerai dans les détails que de mon point de vue. La carte RPi Pico dispose d'une infrastructure minimale pour faire fonctionner le microcontrôleur RP2040 (**figure 1**). Les broches crénelées de la carte sont identifiables, ce qui permet d'assembler facilement (soudure carte à carte) le RPi Pico sur une carte porteuse. On utilise depuis longtemps les broches crénelées sur divers modules de circuits imprimés (les modules Bluetooth ou WiFi par ex.) pour les monter directement sur une autre carte.

Le RP2040 est un microcontrôleur rentable et puissant doté d'interfaces numériques modulables. Un processeur Cortex-M0+ à double cœur cadencé jusqu'à 133 MHz offre des performances suffisantes pour la plupart des applications IdO.

La **figure 2** montre les multiples affectations des GPIO du RPi Pico, ce qui se traduit par un haut degré de modularité dans la conception du matériel. Vous pouvez voir que les deux UART, les deux bus I2C et les

Tableau 1. Broches et fonctions principales

Broche	Direction E/S	Fonction
GPIO29	entrée	ADC3 capture VSYS / 3
GPIO25	sortie	LED utilisateur
GPIO24	entrée	Détection VBUS - haut si VBUS est présent, bas sinon
GPIO23	sortie	Contrôle du SMPS intégré

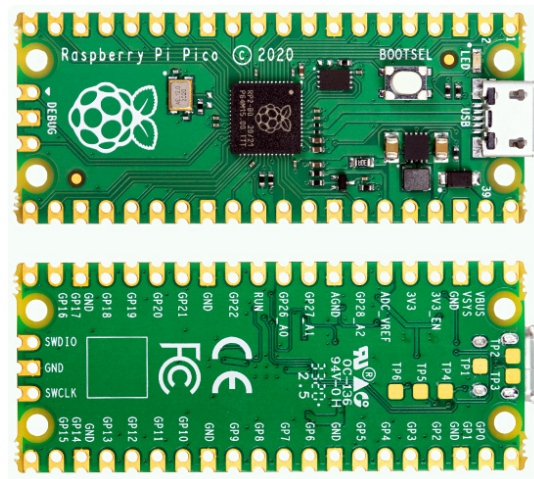


Figure 1. Vue de dessus et de dessous du RPi Pico (Image : fiche technique du RPi Pico).

deux bus SPI sont disponibles plusieurs fois. Seules les connexions CA/N internes sont affectées à des broches fixes. Certains GPIO servent à des fonctions internes. Le **tableau 1** présente les GPIO concernés et leur fonctionnalité. En plus de ces GPIO, sept autres connexions sont disponibles vers l'extérieur. Le **tableau 2** indique la désignation des connexions et leur fonction.

Le 3,3 V commutable (OUT) peut déconnecter les capteurs ou actionneurs de la tension d'alimentation pour économiser de l'énergie. Cela peut être très intéressant pour les applications IdO alimentées par batterie. Si la précision du CA/N ne doit pas être élevée, la tension de référence générée en interne est suffisante. On peut améliorer la

précision avec une tension de référence externe. Celle-ci peut être générée avec un LM4040, par exemple, et appliquée à la connexion ADC_VREF (**figure 3**). Notez que le CA/N de l'actuel RP2040 a encore quelques problèmes de linéarité. Ceci affecte toutes les cartes basées sur le RP2040.

Maker Pi Pico

Le RPi Pico n'est pas la seule carte basée sur le RP2040. Adafruit, Sparkfun, Pimoroni et d'autres fabricants réputés sont présents sur le marché avec leurs variantes de cartes à microcontrôleur basées sur le RP2040. Vous pouvez mesurer la qualité de l'air avec n'importe

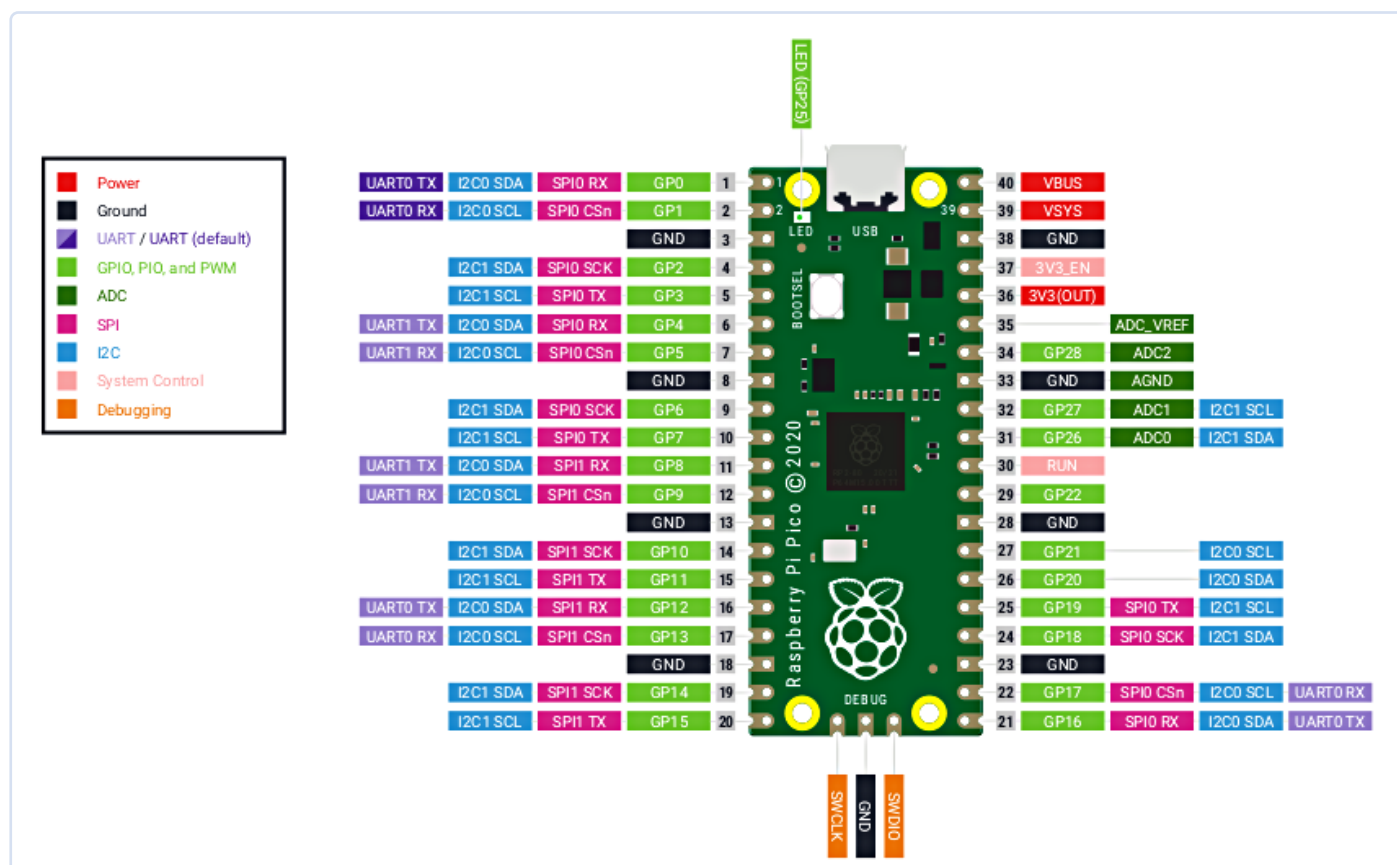


Figure 2. Brochage du RPi Pico (Image : [9]).

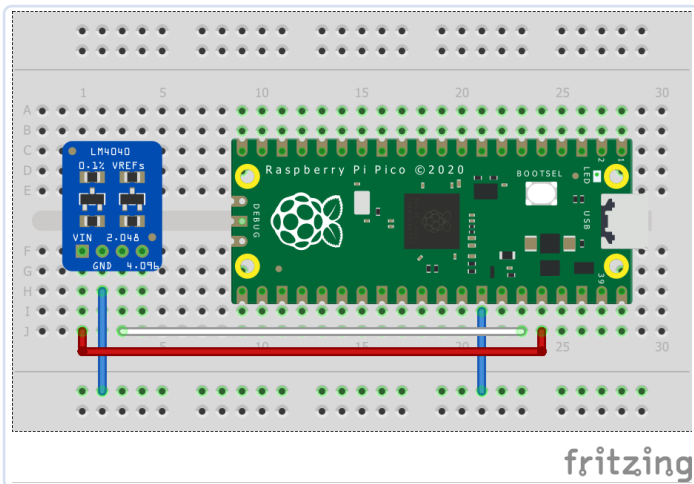


Figure 3. ADC_VREF externe avec le LM4040.

laquelle de ces cartes. J'utilise le Maker Pi Pico à cette fin en raison de l'excellente base expérimentale fournie par Cytron Technologies. Dans la boutique Elektor, vous trouverez d'autres cartes ainsi que des modules complémentaires pour vos projets [3].

Le fabricant Cytron Technologies utilise le slogan « Simplifier le RPi Pico pour les débutants » pour faire la publicité de sa carte Maker Pi Pico (figure 4). Il est toujours important de jouer sur un terrain connu et stable lorsqu'on a affaire à un nouveau microcontrôleur. Le nouvel environnement et les nouveaux outils offrent suffisamment d'occasions de commettre des erreurs.

Le RPi Pico est déjà soudé, et on peut voir les connecteurs Grove à gauche et à droite de la carte. Il y a un emplacement pour une carte microSD. La connexion pour un module ESP-01 en bas à droite est essentielle pour obtenir une connexion à l'Internet. Face aux nombreux modules Wi-Fi plus récents, l'ESP-01 basé sur l'ESP8266 d'Espressif reste un bon vieux module pour cet usage.

La figure 5 montre le brochage de la carte Maker Pi Pico. Vous y trouverez également les broches correspondantes du RPi Pico. Le bouton de réinitialisation de la carte (absent sur le RPi Pico) est essentiel faciliter le travail. Une LED RVB Neopixel permet de signaler les différents états.

Tableau 2. Autres broches et fonctions

Broche	Nom	Fonction
40	VBUS	Tension USB (5 V). Une alimentation externe via USB peut être détectée via GPIO24 (haut : tension disponible ; bas : pas de tension disponible).
39	VSYS	Tension d'entrée du système ; plage de 1,8 à 5,5 VCC. Le SMPS intégré génère les 3,3 V pour alimenter le RP2040 et ses GPIO.
37	3V3_ EN	Fixé au niveau haut par une résistance de rappel. Connecté à GND, le 3,3 V est désactivé.
36	3V3 (OUT)	Alimentation 3,3 V pour les composants externes. La charge doit être ≤ 300 mA.
35	ADC_ VREF	Tension d'alimentation et référence pour le CA/N. L'utilisation d'une référence de tension externe pour une meilleure précision est possible.
33	AGND	Masse analogique pour GPIO26 à 29. Si le CA/N n'est pas utilisé ou si sa précision n'est pas critique, cette broche peut être connectée à la masse numérique (GND).
30	RUN	Mis à l'état haut via une résistance de rappel. Un niveau bas réinitialise le RP2040. Il n'y a pas de bouton de réinitialisation.

La figure 5 montre également les multiples affectations des GPIO du RPi Pico aux connecteurs Grove. Vous pouvez voir que les deux UART, les deux bus I²C et les deux bus SPI sont disponibles plusieurs fois. Seuls les deux canaux CA/N sont affectés de manière fixe au connecteur Grove6. Trois GPIO sont connectés à des boutons programmables et plusieurs sorties à la LED RVB NeoPixel, au buzzer piézoélectrique et à l'audio. Le module Wi-Fi ESP-01 a déjà été mentionné. L'ajout d'un module Wi-Fi ESP-01 élimine le principal défaut du RP2040. La combinaison d'une carte RP2040 et d'un module Wi-Fi ESP8266 coûte généralement moins de 12 €. Comparez cela aux 30 € que vous devez payer normalement pour la carte Arduino Nano RP2040 Connect.

Mesurer la qualité de l'air

La qualité de l'air est une question essentielle en ces temps de Corona. Le risque d'infection est plus faible pendant l'été, mais les prévisions varient pour cet automne et cet hiver. Les polluants tels que la poussière fine, les fibres d'amiante, le formaldéhyde, le PCB, le radon, les produits de nettoyage, les moisissures, la poussière, la fumée de tabac et les composés organiques volatiles (COV) affectent la qualité de l'air intérieur, en plus du CO₂.

La proportion de dioxyde de carbone dans l'air que nous respirons aujourd'hui est d'environ 415 ppm ($\approx 0,04$ %). L'air expiré par une personne a une teneur en CO₂ d'environ 40 000 ppm. Par conséquent, dans les chambres à coucher non ventilées, les salles de classe entièrement occupées ou les salles de réunion, nous pouvons rapidement mesurer jusqu'à 5 000 ppm. Ces concentrations élevées de CO₂ sont néfastes pour l'attention, les performances et la santé en général. Si vous souhaitez évaluer la qualité de l'air de manière exhaustive, vous devez prévoir un système complexe de capteurs. Dans un système

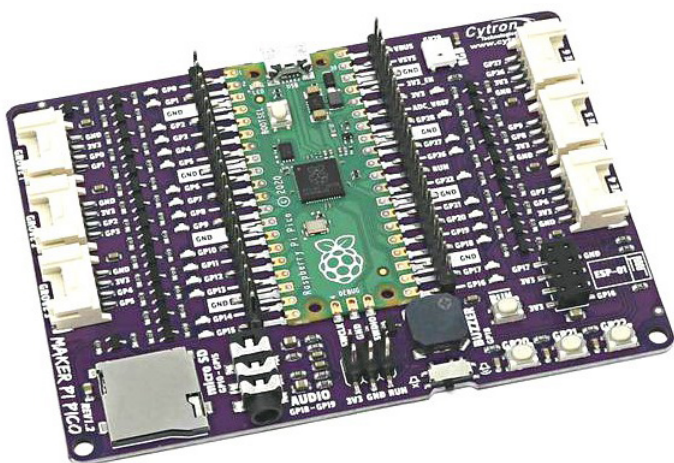


Figure 4. Vue de dessus du Maker Pi Pico (Image : Cytron).

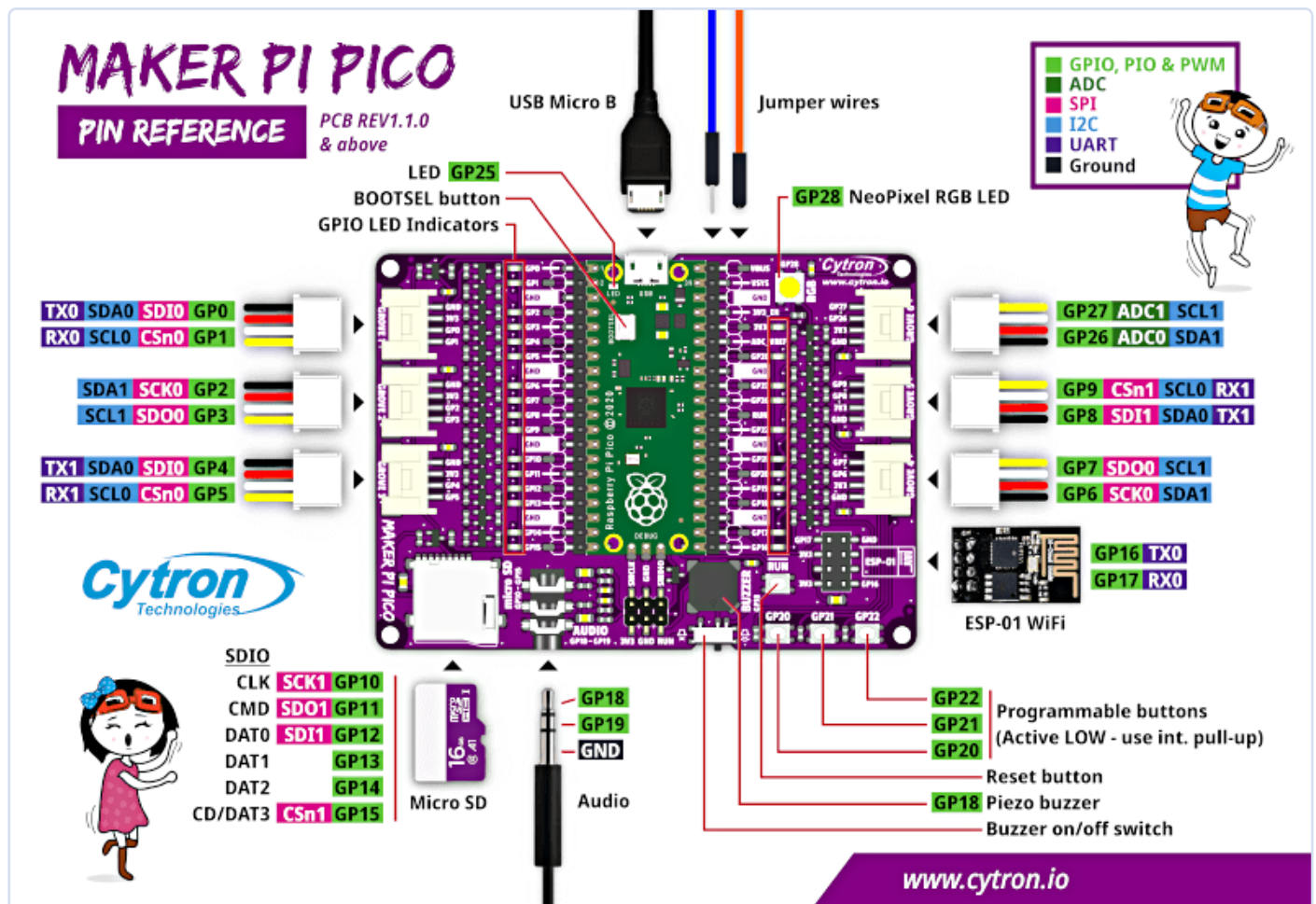


Figure 5. Brochage de la carte Maker Pi Pico (Image : Cytron).

de mesure professionnel, par exemple, différents capteurs mesurent les valeurs de CO₂, de poussières fines (catégories de taille pertinentes : PM1, PM2.5 et PM10), de NO₂, de CO, le bruit, la température, l'humidité et la pression atmosphérique. Cet article se concentre sur la mesure de deux valeurs importantes, le CO₂ et les COV, qui peuvent être mesurés par deux types de capteurs :

- Les capteurs NDIR (Non-Dispersive Infrared Sensors) utilisent l'absorption du rayonnement électromagnétique dans la gamme infrarouge en fonction de la concentration. L'absorption maximale de CO₂ se produit à une longueur d'onde de 4,3 µm, sans grande influence des autres gaz. La concentration de CO₂ peut donc être mesurée simplement et de manière très sélective.
- Avec les capteurs MOX, le gaz qui circule provoque une modification d'une couche d'oxyde métallique sensible au gaz. La modification de la résistance mesure la concentration des COV enregistrés dans leur intégralité car insolubles dans une substance particulière. Avec les capteurs MOX à large bande mesurant les COV, on dispose de capteurs qui détectent toute une série de substances dangereuses pour la santé à des concentrations spécifiques.

Comme indiqué, cet article se concentre sur la mesure du CO₂ et des COV. Supposons que nous utilisons la concentration de CO₂ comme mesure de la qualité de l'air. Nous avons alors une bonne indication de la pollution de la qualité de l'air par l'air expiré par les personnes

présentes et du risque associé d'infection par des virus transmis par les aérosols.

Supposons que nous utilisons des capteurs MOX à plus large bande pour mesurer la qualité de l'air. Dans ce cas, nous avons une bonne indication de la pollution de la qualité de l'air par divers polluants, y compris les vapeurs et les odeurs humaines. Dans une étude approfondie, j'ai examiné le comportement des capteurs NDIR et MOX lors de la mesure de la qualité de l'air [4]. Désolé, le texte est en allemand, mais la **figure 6** montre un résultat important. Dans l'une de ces expériences, j'ai fait des mesures avec deux capteurs CO₂ et un capteur MOX. Les capteurs de CO₂ utilisés étaient le SCD30 (NDIR) et le SCD41 (PASense), développés par la société suisse Sensirion. Le capteur de MOX était un SGP30 également de Sensirion. Comme il n'y avait pas de substances supplémentaires en plus des vapeurs humaines, les valeurs mesurées par le capteur MOX SGP30 suivent également les deux capteurs de CO₂ SCD30 et SCD41.

La mesure de la qualité de l'air des capteurs MOX, plus économiques, donne des résultats comparables à la concentration de CO₂ pur partout où l'air respiré et les vapeurs humaines sont dominants. Des écarts apparaissent dans les environnements pollués (par exemple, avec le formaldéhyde).

Qualité de l'air via le capteur MOX

Comme un capteur MOX suffit pour mesurer la qualité de l'air, j'ai utilisé le Sensirion SGP30 dans mon projet. Ce capteur mesure les concentrations de composés organiques volatils totaux (COVT) et

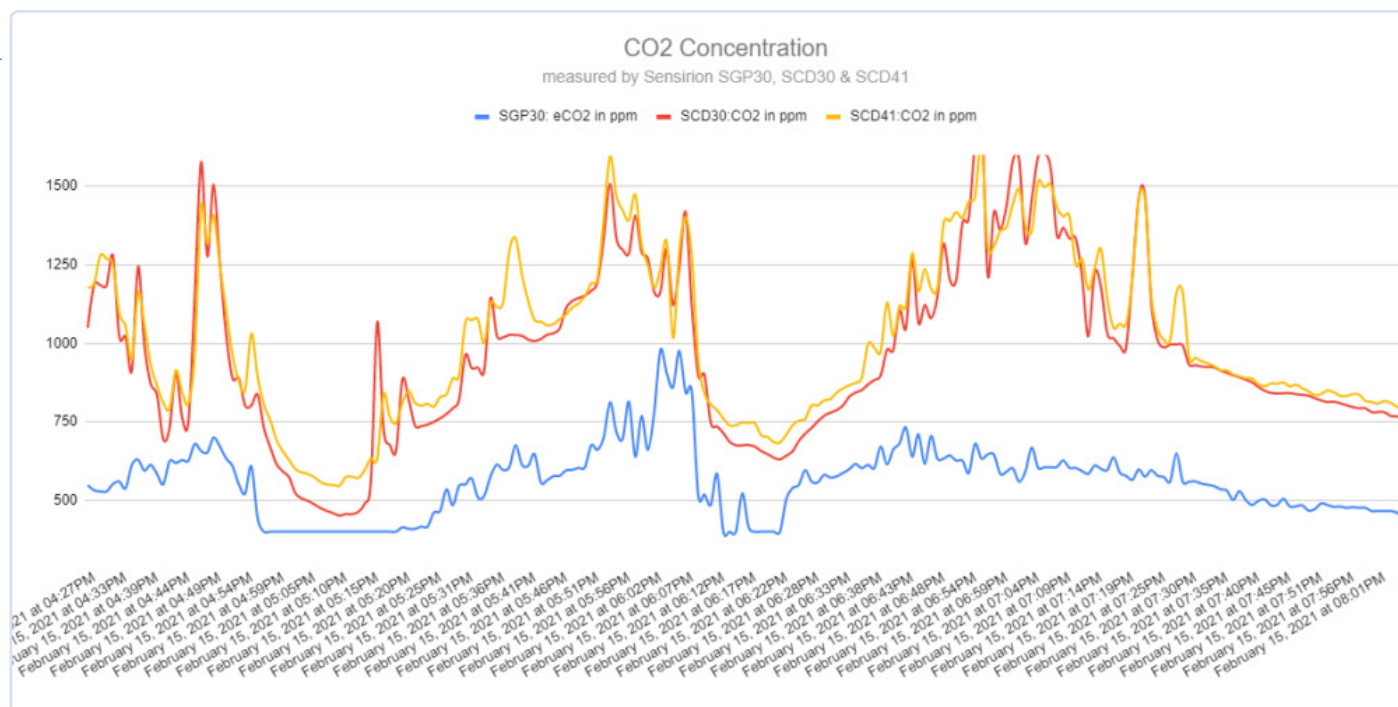


Figure 6. Concentration de CO₂ mesurée par le NDIR et le capteur MOX.



Figure 7. SGP30 (Image : fiche technique Sensirion SGP30).

les concentrations calculées équivalentes de CO₂ (eCO₂). eCO₂ est calculé sur la base de la concentration de H₂ et n'est donc pas aussi précis que la mesure directe du CO₂, mais cela suffit pour mon objectif. Ne pouvant pas manipuler la puce (figure 7) elle-même, j'ai utilisé l'unité M5Stack TVOC/eCO₂. Il s'agit d'un SGP30 mis en boîtier, connectable via un connecteur Grove I²C (figure 8).

Un écran OLED de 2,4 cm connecté via I²C affiche les valeurs mesurées pour TVOC et eCO₂. L'ESP-01 se charge de l'accès à l'Internet et est connecté via l'interface UART Serial1 au RPi Pico. La figure 9 montre tous les périphériques utilisés connectés à la carte Maker Pi Pico. Le NeoPixel sert de feu de signalisation. Le vert signifie que la qualité de l'air est bonne, le jaune qu'elle l'est moins et le rouge qu'elle est mauvaise (il est alors indispensable de ventiler).

La figure 10 montre le circuit complet. Le contrôleur interroge le capteur SGP30 toutes les cinq secondes et transmet les dernières



Figure 8. Capteur TVOC/eCO M5Stack2 (Image : M5Stack).

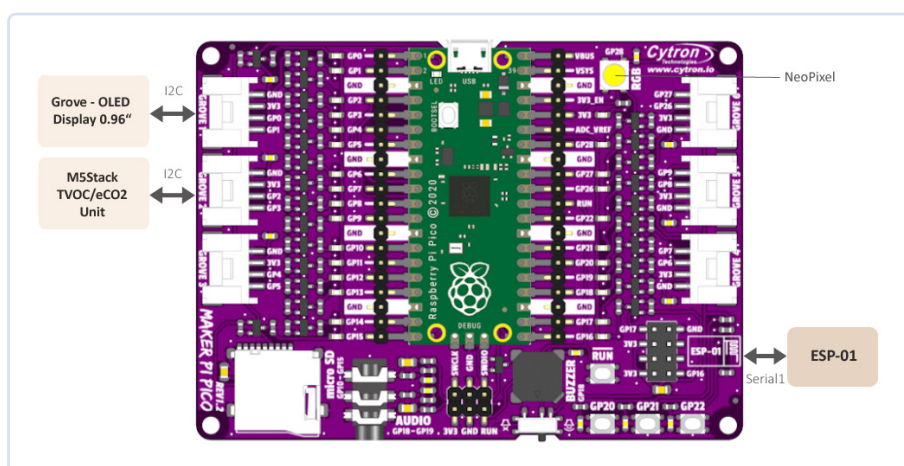


Figure 9. Périphériques utilisés connectés au Maker Pi Pico (Image : Cytron/Auteur).

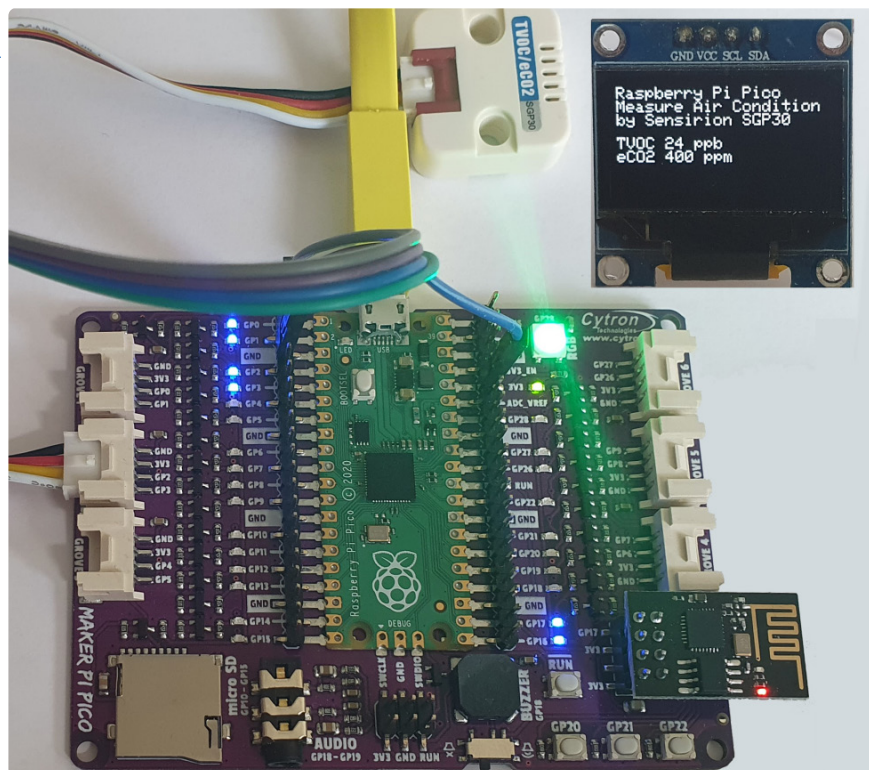


Figure 10. Mesure de la qualité de l'air avec Sensirion SGP30 et RPi Pico.

valeurs mesurées au serveur ThingSpeak une fois par minute. La couleur de la LED NeoPixel correspond à la valeur mesurée d'eCO₂ et peut être facilement ajustée (**figure 11**).

Pour visualiser les valeurs mesurées sur ThingSpeak, les données sont envoyées via un accès HTTP GET à l'API ThingSpeak. La **figure 12** montre la visualisation des valeurs mesurées dans ThingSpeak [5].

Logiciel

Pour cette application, j'ai utilisé l'EDI Arduino 2.0, comme mentionné dans la préface. Earle F. Philhower III a effectué le portage de la programmation du RPi Pico dans l'EDI Arduino. Grâce à son travail, on peut maintenant programmer toutes les différentes cartes RP2040 avec l'EDI Arduino. Ce portage du RP2040 (processeur RPi Pico) vers l'écosystème Arduino utilise le SDK Raspberry Pi

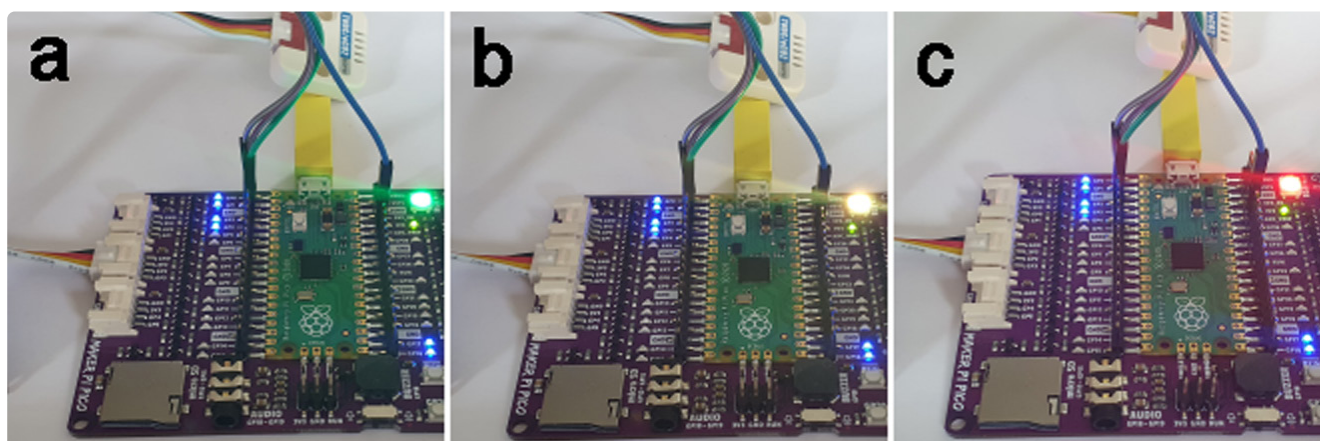


Figure 11. Signalisation de la valeur mesurée d'eCO₂ par la couleur de la LED RVB NeoPixel.

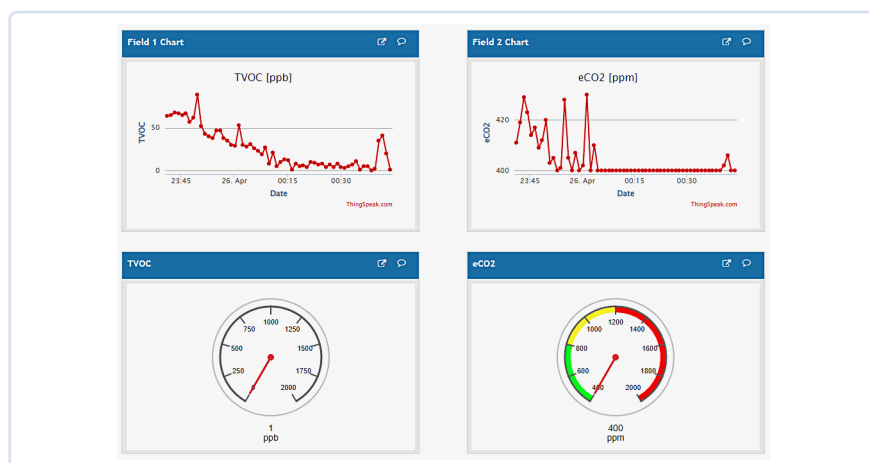


Figure 12. Affichage de Thingspeak.

Pico nu et une chaîne d'outils GCC 10.3/Newlib 4.0 personnalisée.

En mode d'amorçage, les cartes sont automatiquement reconnues et affichées dans l'EDI Arduino. La commande de téléchargement fonctionne avec l'outil Microsoft UF2 (inclus). Vous pouvez trouver les instructions d'installation et d'utilisation sur GitHub [6].

La **figure 13** montre le programme d'exemple *RP2040_WiFi.ino*. Il utilise deux fichiers supplémentaires, *sendMessage.ino* et *arduino_secrets.h*. Ils sont disponibles en téléchargement sur GitHub [7]. Lors des premières compilations de cette version du logiciel, le compilateur a

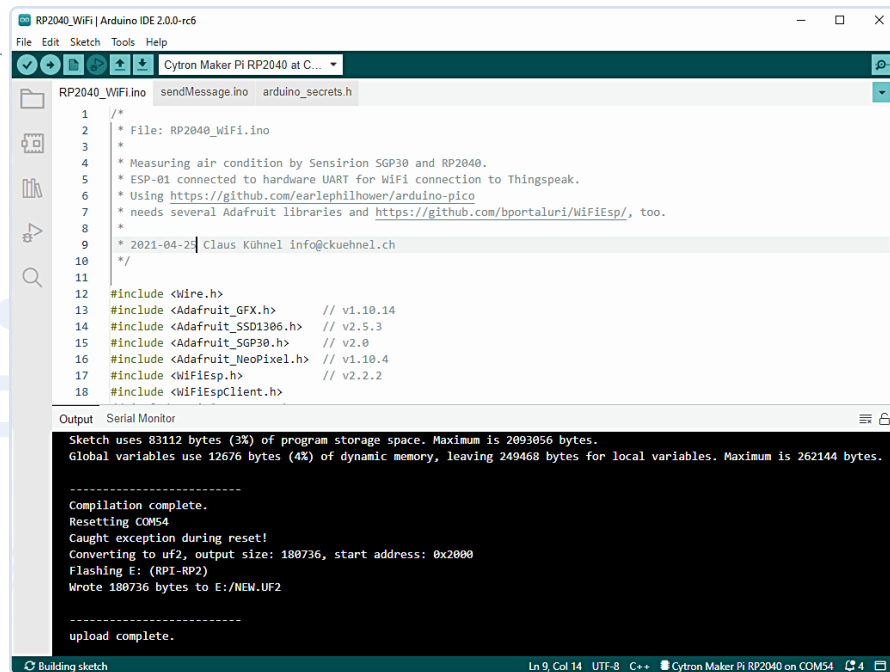


Figure 13. Exemple de programme RP2040_WiFi.ino dans l'EDI Arduino 2.0.

trouvé deux erreurs dans le fichier WiFiEspClient.cpp de la bibliothèque WiFiEsp v2.2.2. Après avoir ajouté les déclarations de retour manquantes, cela a fonctionné.

```
size_t WiFiEspClient::print(const __FlashStringHelper
    *ifsh)
{
    return printFSH(ifsh, false); // 2022-04-25 CK
}

// si nous surchargeons ceci, le println standard
// appellera deux fois la méthode d'impression
size_t WiFiEspClient::println(const __
    FlashStringHelper *ifsh)
{
    return printFSH(ifsh, true); // 2022-04-25 CK
}
```

Le programme *RP2040_WiFi.ino* comprend toutes les bibliothèques nécessaires. J'ai indiqué leurs versions utilisées en commentaire. Les déclarations et initialisations suivent avant la fonction `setup()`. L'initialisation des périphériques est intéressante et souligne la modularité du RP2040. Le code suivant est un extrait du code source commenté montrant l'affectation des fonctions aux GPIO et leur initialisation.

```
//I2C0 pour la connexion du SSD1306
Wire.setSDA(0);
Wire.setSCL(1);
Wire.begin();
...
// I2C1 pour la connexion du SGP30
Wire1.setSDA(2);
Wire1.setSCL(3);
Wire1.begin();
...
```

```
//Serial pour la connexion de l'ESP-01
Serial1.setRX(17);
Serial1.setTX(16);
Serial1.begin(115200);
WiFi.init(&Serial1);
```

Les valeurs eCO₂ inférieures à 800 sont indiquées par une LED verte. La LED s'allume en jaune entre 800 et 1200. Au-delà de 1200, la LED devient rouge (**figure 11c**). Modifiez ces niveaux en fonction de vos besoins.

La fonction `sendMessage()` se charge de l'accès HTTP GET à l'API de Thingspeak. La ligne

```
client.println("GET /update?api_key=" + api_key +
    "&field1=" + String(TVOC) + "&field2=" +
    String(eCO2) + " HTTP/1.1");
```

envoie la clé API, les valeurs TVOC et eCO₂ à l'API de Thingspeak (**figure 14**). En outre, vous pouvez voir dans cette capture d'écran la sortie série du programme après la réinitialisation.

Pour l'accès wifi, il faut, dans le fichier `arduino_secrets.h`, préciser votre SSID et votre mot de passe et pour Thingspeak, le jeton d'écriture. L'inscription à Thingspeak est requise. Les données dans le code suivant sont des exemples non fonctionnels.

```
// Accès aux données pour les services

// wifi
const char* ssid = "Sunrise_2.4GHz_8xxxxx0";
const char* pass = "u2uxxxxxDs";

// Thingspeak
// GET https://api.thingspeak.com/
```

```

21 void sendMessage()
22 {
23     // close any connection before send a new request
24     // this will free the socket on the WiFi shield
25     client.stop();
26
27     Serial.println("Starting connection to server...");
28     // if you get a connection, report back via serial
29     if (client.connect(server, 80))
30     {
31         Serial.println("Connecting...");
32         // Make a HTTP request
33         client.println("GET /update?api_key=" + api_key + "&field1=" + String(TVOC) + "&field2=" + String(eCO2) + " HTTP/1.1");
34         client.println("Host: api.thingspeak.com");
35         client.println("Connection: close");
36         client.println();
37     }
38 }

```

Output Serial Monitor x

Message (Ctrl + Enter to send message to 'Cytron Maker Pi RP2040' on 'COM54')

```

RP2040 Air Condition Measurement
Found SGP30 serial #01684D23
[WiFiEsp] Initializing ESP module
[WiFiEsp] Initialization successful - 1.5.4
Attempting to connect to WPA SSID: Sunrise_2.4GHz_8AC2A0
[WiFiEsp] Connected to Sunrise_2.4GHz_8AC2A0
You're connected to the network
SSID: Sunrise_2.4GHz_8AC2A0
IP Address: 192.168.1.217
Signal strength (RSSI): -77 dBm
Initialization finished.
TVOC 0 ppb      eCO2 400 ppm
TVOC 0 ppb      eCO2 400 ppm
TVOC 0 ppb      eCO2 400 ppm

```

Figure 14. Fonction `sendMessage()` dans le fichier `sendMessage.ino` et sortie série.

```

update?api_key=33xxxxxxSZE&field1=0&field2=0
const char* server = «api.thingspeak.com»;
String api_key      = «33xxxxxxSZE»;

```

Vous pouvez étendre cette application à un système en circuit fermé basé sur le matériel utilisé ici. Utilisez d'autres unités Grove pour contrôler la ventilation d'une pièce selon la valeur d'eCO₂, par exemple. En fonction de la température et de l'humidité, la régulation d'un système de climatisation est également possible. Vous devriez maintenant être

en mesure de développer d'autres idées pour mettre en œuvre l'automatisation dans votre environnement. La consultation d'un autre article d'Elektor [8] pourrait vous fournir des informations complémentaires sur la mesure de l'air et vous inspirer pour d'autres projets.

210636-04



PRODUITS

- **RP2040 controller boards and more @ Elektor**
www.elektor.com/raspberry-pi/rp2040
- **Sensors @ Elektor**
www.elektor.com/catalogsearch/result/?q=Sensors

À propos de l'auteur

Claus Kühnel a étudié les technologies de l'information à l'Université technique de Dresde et a développé pendant de nombreuses années des systèmes embarqués pour des appareils de diagnostic de laboratoire, entre autres. Dans ce domaine interdisciplinaire, il est entré en contact avec la scène maker. Il a publié de nombreux articles et livres sur le matériel et les logiciels de microcontrôleurs en Allemagne et à l'étranger. Il est passionné par les nouvelles technologies autour des microcontrôleurs.

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

LIENS

- [1] L. Lemmens, M. Claussen, « Carte Raspberry Pi Pico à RP2040 », Elektor 5-6/2021: www.elektormagazine.fr/magazine/elektor-176/59572
- [2] Téléchargement EDI Arduino 2.0 (Windows 64 bits): https://downloads.arduino.cc/arduino-ide/arduino-ide_2.0.0-rc6_Windows_64bit.exe
- [3] Cartes microcontrôleur RP2040 et plus à Elektor: www.elektor.fr/raspberry-pi/rp2040
- [4] Mesurer la qualité de l'air avec des capteurs NDIR et MOX (allemand): <https://tinyurl.com/yvj9z5sb>
- [5] Résultats via Thingspeak: <https://thingspeak.com/channels/1392977>
- [6] Noyau Arduino Raspberry Pi Pico pour RP2040: <https://github.com/earlephilhower/arduino-pico>
- [7] Exemple RP2040_WiFi.ino: https://github.com/ckuehnel/Arduino2020/tree/master/RP2040/RP2040_WiFi
- [8] L. Lemmens, M. Claussen, « Détecteur de CO₂ : CO₂ Guard », Elektor 5-6/2022: www.elektormagazine.fr/magazine/elektor-259/60520
- [9] Brochure du RPi Pico: <https://datasheets.raspberrypi.com/pico/Pico-R3-A4-Pinout.pdf>