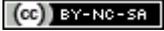
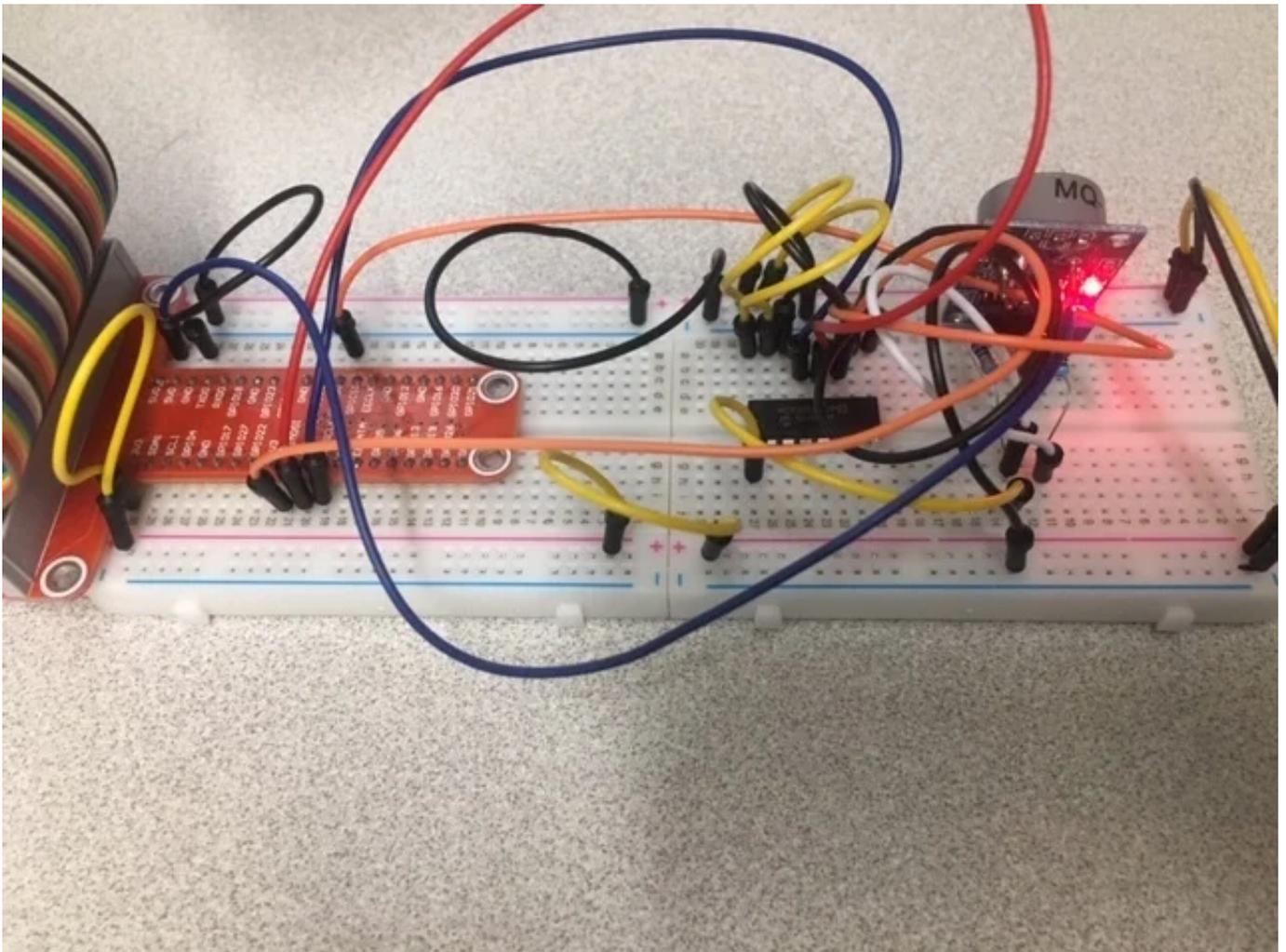


Capteur de monoxyde de carbone de la framboise Pi

Par [jeanet.g1](#) dans les [circuits Raspberry Pi](#)



Introduction: Capteur de monoxyde de carbone de Raspberry Pi



L'Internet des objets est celui qui est connecté à l'internet et qui présente les caractéristiques de l'auto-adaptabilité, de l'auto-configuration, des protocoles de communication interopérables et qui ont les entités physiques et virtuelles uniques. Les capteurs sont les dispositifs qui mesurent certaines caractéristiques physiques et environnementales et sont principalement utilisés pour collecter des données dans les dispositifs IdO. Aux fins de ce projet, nous avons choisi le

Nous utilisons des cookies pour améliorer votre expérience et personnaliser le contenu. Voir notre [déclaration sur les cookies](#) pour plus de détails. [Gérer vos préférences en matière de cookies.](#)

Déclin

Accepter

Fournitures

Raspberry Pi 3

MQ-7 Capteur de monoxyde de carbone

Panier à pain

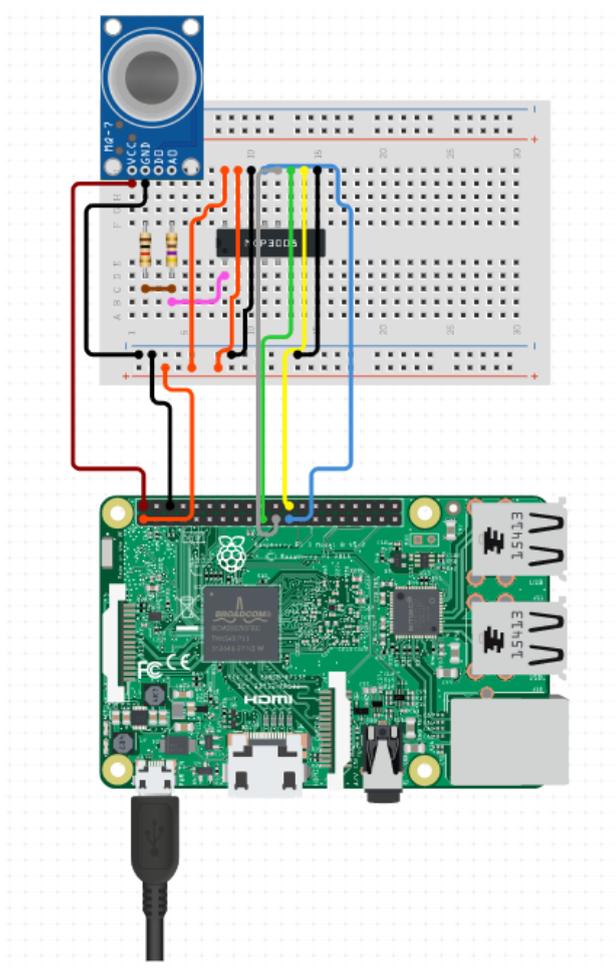
1K Résisteur Ohm

Résisteur de 470 Ohm

MCP3008 8 Canal, ADC 10 bits avec interface SPI

Ordinateur

Étape 1: Mise en place du Pi



Suivez le schéma de principe pour configurer votre circuit. Pour une version interactive de l'image, visitez [le diagramme de circuito.io](#)

Nous utilisons des cookies pour améliorer votre expérience et personnaliser le contenu. Voir notre [déclaration sur les cookies](#) pour plus de détails. [Gérer vos préférences en matière de cookies.](#)

Déclin

Accepter

Étape 3 : Code

Vous pouvez trouver tout le code source pour ce projet dans notre [dépôt de github](#). Assurez-vous que lorsque vous le téléchargez sur votre ordinateur local et gardez la structure du fichier de la même manière que vous le transférez au pi.

Ouvrez le fichier `sensor.py` et changez les lignes de code contenant `localhost` pour être l'adresse IP de votre ordinateur. Vous pouvez trouver des instructions sur la façon de trouver l'adresse IP de votre ordinateur [ici](#).

Nous devons déplacer le fichier **sensor.py** vers le framboise pi, alors exécutez cette commande à partir du répertoire COSensor

```
scp sensor.py pi" 'enter pi ip address ici':
```

Nous utilisons des cookies pour améliorer votre expérience et personnaliser le contenu. Voir notre [déclaration sur les cookies](#) pour plus de détails. [Gérer vos préférences en matière de cookies](#).

Déclin

Accepter

Étape 4: Configuration des messages de texte

Puisque notre programme nous alerte à travers des messages textuels lorsque les niveaux de monoxyde de carbone deviennent trop élevés, nous devons être en mesure d'envoyer des messages à partir d'un numéro central. Pour ce faire, nous utiliserons une plateforme appelée Twilio. Tout d'abord, [inscrivez-vous](#) pour un compte d'essai gratuit. Ensuite, nous devons télécharger quelques paquets. Si vous avez déjà Node.js installé avec la version v8.0.0 ou supérieure, sautez à l'étape 2. Vous pouvez vérifier votre version avec:

```
node -v
```

Le premier paquet est Node.js qui peut être téléchargé à partir de leur site [ici](#). Assurez-vous d'installer votre système d'exploitation correct, puis de l'ouvrir et de suivre les instructions fournies.

Ensuite, nous devons installer le Twilio CLI. Celui-ci est installé et mis à jour avec le gestionnaire de paquets Node avec les commandes suivantes:

```
npm installation twilio-cli -g
npm installation twilio-cli-latest-g
```

À ce stade, nous devons connecter le Twilio CLI à notre compte. Pour ce faire, nous avons besoin de deux éléments d'information : notre compte SID et notre jeton Auth de la [console Twilio](#). Puis exécutez `twilio login` et entrez l'information induite.

```
twilio Login
```

Nous avons donc maintenant connecté notre compte mais nous avons encore besoin d'un numéro de téléphone. Vous pouvez en acheter un via Twilio avec l'argent que vous avez reçu. Après avoir tapé la commande ci-dessous, un groupe de nombre apparaîtra; choisir un.

```
twilio numéros de téléphone:buy:local --country-code US --sms-enabled
```

Maintenant, pour que Twilio travaille dans notre programme, nous devons installer ses paquets. Type

```
pip install twilio
```

À l'intérieur du fichier `keys.py`, nous devons entrer dans notre compte SID et notre Jeton Auth pour une utilisation ultérieure. Il devrait déjà y avoir un point blanc pour copier et coller ces valeurs.

```
mots de passe - "twilio": "compte-sid": "collez votre sid ici",
                "Jouet-en-jambes": "collez votre jeton ici"/p.
```

Tout cela étant fait, il est maintenant temps de mettre en place la compatibilité des e-mails avec notre programme via l'API Gmail.

Nous utilisons des cookies pour améliorer votre expérience et personnaliser le contenu. Voir notre [déclaration sur les cookies](#) pour plus de détails. [Gérer vos préférences en matière de cookies.](#)

Déclin

Accepter

Étape 5: Configuration de l'API Gmail

Pour configurer l'API gmail, vous devez tout d'abord visiter le [tableau de bord google](#). Vous pouvez ici enregistrer le nouveau projet en utilisant l'option «créer le projet». Après la création du nouveau projet, il y aura une invite qui dira: «Vous n'avez pas encore d'API disponible à utiliser. Pour commencer, veuillez consulter la bibliothèque de l'API ».

Puis visitez [ici](#). Dans la zone de recherche, sélectionnez l'API Gmail. Après avoir cliqué sur l'option Gmail API, il y aura une option pour activer cette API. Une fois que vous aurez activé l'API Gmail, vous devrez créer des identifiants pour que vous puissiez l'utiliser. Par conséquent, cliquez sur « Créer des accréditations », cela vous mènera à la fenêtre vous demandant de sélectionner l'API. Là, sélectionnez l'API Gmail, puis sélectionnez la bonne option où vous appellerez cette API Gmail à partir de. Après cela, vous aurez besoin de choisir votre rôle: quelque chose comme le propriétaire du produit. Ensuite, le fichier json sera téléchargé sur votre ordinateur, qui sera votre compte de service, copier et coller ceci à l'intérieur du dossier cred dans votre répertoire de projet. Hurray puis votre API est activée et votre compte est enregistré pour utiliser cette API Gmail. Maintenant vient la partie amusante, disons que nous voulons envoyer un e-mail en utilisant votre compte enregistré avec Gmail API. Consultez [ce site web](#) pour référence sur le code et la manière dont le code fonctionne pour envoyer des e-mails à partir du compte enregistré. La première chose à retenir est de définir le SCOPE qui vous permet d'envoyer des e-mails. Le champ d'envoi de courriels ressemble à: «https://www.googleapis.com/auth/gmail.send». Vous pouvez trouver la liste des champs d'autorisation [ici](#)

Tout ce que vous faites en utilisant l'API Gmail comme accéder aux étiquettes du courrier électronique, ou envoyer l'e-mail, le nouveau jeton de cornichon est créé, cela se produit pour la première fois que vous exécutez l'application. Chaque fois que vous ajoutez un nouveau champ, le nouveau cornichon de jeton est créé, ce qui permet toutes les fonctions que vous pouvez effectuer en utilisant l'API gmail. Chaque fois que vous exécutez votre application en changeant le champ d'application, le nouveau jeton de cornichon est créé.

Étape 6: Courir le capteur

Maintenant, nous pouvons enfin exécuter notre programme. Ouvrez une séance ssh à votre framboise pi et en une seule course:

```
capteur python3.py
```

Sur votre ordinateur, exécutez

```
python api.py
```

Maintenant, nous pouvons accéder aux données et nous abonner pour recevoir des notifications à partir de la page web. Ouvrez un navigateur web et tapez <http://host:5000> pour voir les niveaux actuels de CO. Naviguez jusqu'à la page de l'abonnement et entrez dans vos informations pour recevoir des notifications.

Nous utilisons des cookies pour améliorer votre expérience et personnaliser le contenu. Voir notre [déclaration sur les cookies](#) pour plus de détails. [Gérer vos préférences en matière de cookies.](#)

Déclin

Accepter