

Jour 1

Préparer le Raspberry Pi

Le matériel suivant est requis pour la mise en service du Raspberry Pi :

- Clavier USB et souris
- Câble HDMI pour moniteur
- Câble réseau
- Carte microSD avec système d'exploitation
- Chargeur micro USB pour téléphone portable comme bloc d'alimentation

Raccordez le bloc d'alimentation en dernier, le Raspberry Pi démarre alors automatiquement. Il n'est pas équipé d'un interrupteur marche / arrêt.

Installation du système d'exploitation en bref

Pour tous ceux qui doivent encore installer la version actuelle de Raspbian sur leur Raspberry Pi, l'installation du système est décrite ici en 10 étapes :

- 1 Télécharger NOOBS sur l'ordinateur : www.raspberrypi.org/downloads puis extraire l'archive Zip sur le disque dur.
- 2 Si la carte SD a déjà été utilisée, la formater sur l'ordinateur à l'aide de l'application SD-Formatter : www.sdcard.org/downloads/formatter_4. Activer ici **Format Size Adjustment**.
- 3 Copier les fichiers et sous-dossiers sur la carte SD.
- 4 Retirer la carte SD de l'ordinateur, l'insérer dans le Raspberry Pi puis démarrer le système. Tout en bas, sélectionner **Français** comme langue d'installation. Le clavier français est alors également automatiquement sélectionné.
- 5 Cocher la case pour le système d'exploitation Raspbian présélectionné.
- 6 Après la demande de confirmation de l'écrasement de la carte mémoire, l'installation démarre et dure plusieurs minutes.
- 7 Lorsque l'installation est terminée, le Raspberry Pi redémarre et lance automatiquement l'outil de configuration **raspi-config**.
- 8 Sous **3 Enable Boot to Desktop/Scratch**, sélectionner l'option **Desktop Log in as user ,pi' at the graphical desktop**.
- 9 Sous **4 Internationalisation Options**, sélectionner l'option **Change Timezone** puis définir le fuseau horaire **Europe/Paris**.
- 10 Sélectionner **<Finish>** puis redémarrer le Raspberry Pi.

La DEL est allumée

Aucun programme n'est requis pour la première expérience. Le Raspberry Pi est ici uniquement employé comme alimentation électrique pour la DEL. L'expérience montre comment raccorder des DEL.

Composants : 1 tableau de connexions, 1 DEL rouge, 1 résistance 220 $\times$, 2 câbles de raccordement

Jour 2

ScratchGPIO est un module d'extension pour Scratch, un langage de programmation facile à apprendre, qui permet de contrôler l'interface GPIO du Raspberry Pi.

Installer ScratchGPIO

Scratch est fourni avec Raspbian, le module d'extension ScratchGPIO doit encore être installé à partir d'une fenêtre LXTerminal.

- 1 Démarrer LXTerminal à partir du symbole dans la barre des tâches.
- 2 `wget http://bit.ly/1wxrqdp -O isgh7.sh`

3 `sudo bash isgh7.sh`

4 Deux nouveaux symboles s'affichent sur le bureau Raspbian.

5 Pour la plupart des expériences, **ScratchGPIO 7** constitue le choix idéal. La variante **ScratchGPIO 7plus** est requise pour différentes platines d'extension.

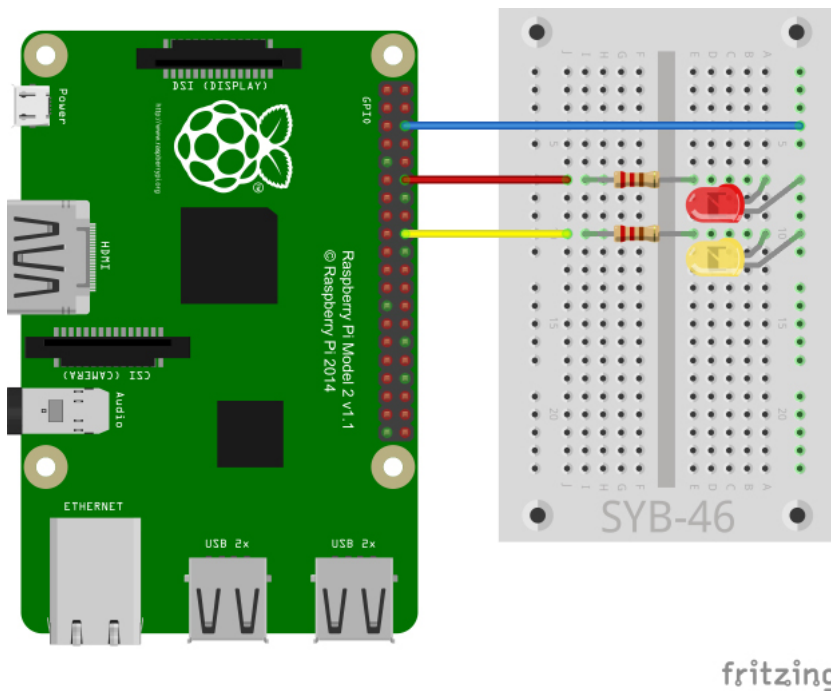
L'affectation des broches dans ScratchGPIO

Une broche GPIO peut soit être employée comme entrée soit comme sortie. ScratchGPIO désigne les broches GPIO d'une autre manière que tous les autres programmes et langages de programmation sur le Raspberry Pi. L'illustration tout en bas à droite montre les broches définies comme entrée (7, 8, 10, 19, 21-24 et 26) ainsi que les broches définies comme sortie (11-13, 15, 16, 18). Les broches avec les numéros 29, 31-33, 35-38 et 40 peuvent aussi bien être employées comme entrée que comme sortie.

Les DEL clignotent

L'expérience du deuxième jour permet de faire clignoter deux DEL en alternance. L'ensemble de la procédure se pilote à l'aide d'un programme dans ScratchGPIO.

Composants : 1 tableau de connexions, 1 DEL rouge, 1 DEL jaune, 2 résistances 220 $\times$, 3 câbles de raccordement

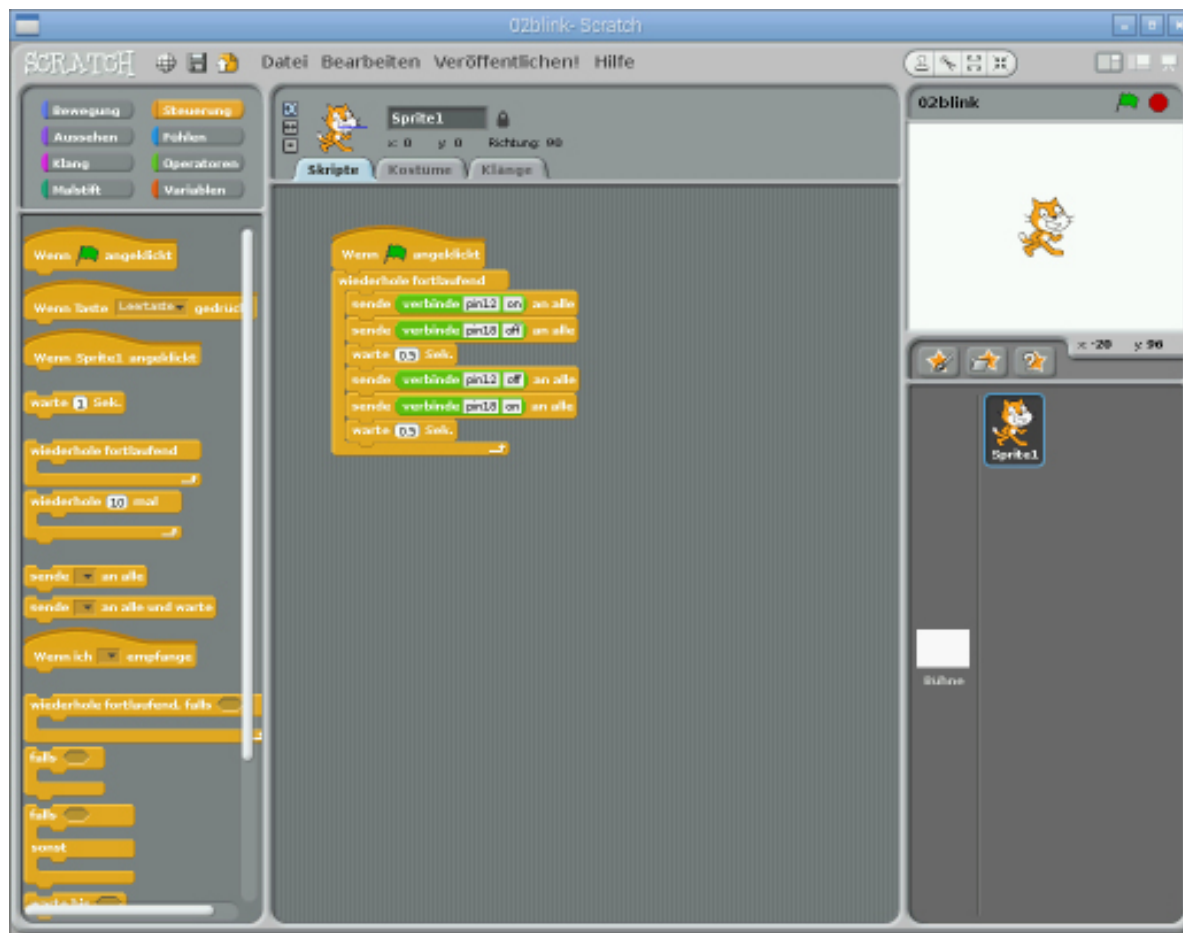


fritzing

Deux DEL clignotent sur le Raspberry Pi

Le programme

Dans Scratch, il n'est pas nécessaire de saisir le code du programme pendant la programmation. Il suffit d'ajouter les blocs par glisser-déposer. La palette de blocs dans la partie gauche de la fenêtre Scratch contient les blocs disponibles, classés par thèmes.



Ce programme Scratch 02blink fait clignoter les DEL en alternance

Le programme démarre lorsque l'utilisateur clique en haut à droite sur le drapeau vert dans la fenêtre Scratch.

Une boucle **repeated continuous** assure le clignotement en continu des DEL jusqu'à ce que l'utilisateur clique sur le symbole Stop rouge en haut à droite dans Scratch.

Les instructions GPIO sont générées via le bloc Scratch **send...to all**. Dans le champ de texte, un bloc **connect** de la palette de blocs verte **operators** associe la désignation respective de la broche et le mot-clé **on** ou **off** au message.

Après la mise en circuit de la DEL rouge sur la broche 12 et la mise hors circuit de la DEL jaune sur la broche 18, le programme attend une demi-seconde. (Remarque : comme de nombreux autres programmes américains, Scratch emploie le point comme séparateur décimal et non pas la virgule courante en France).

De la même manière, la DEL rouge sur la broche 12 est ensuite mise hors circuit et la DEL jaune sur la broche 18 mise en circuit.

Jour 3

Feu tricolore

L'expérience du troisième jour commute la séquence typique d'un feu tricolore avec trois DEL : du rouge au vert en passant par rouge / jaune puis au rouge en passant par le jaune.

Composants : 1 tableau de connexions, 1 DEL rouge, 1 DEL jaune, 1 DEL verte, 3 résistances 220 $\times$, 4 câbles de raccordement

Le programme

Le programme fonctionne de manière similaire au programme d'hier. Ici aussi, différentes combinaisons des DEL sont mises en circuit et hors circuit au moyen d'une boucle infinie. Au cours des phases intermédiaires rouge / jaune et jaune, le feu tricolore reste respectivement allumé pendant 0,5 seconde et, au cours des phases rouge et verte, respectivement 2 secondes. Ces durées peuvent également être configurées d'une

autre manière dans les blocs **wait for...sec.**



Le programme 03ampe1 pilote le feu tricolore

Jour 4

DEL RVB

Une DEL normale ne brille toujours qu'avec une seule couleur. Avec les DEL du calendrier de l'Avent, il est même possible de reconnaître la couleur lorsque la DEL est éteinte. Mais il existe également des DEL transparentes dont la couleur est uniquement visible lorsque les DEL sont alimentées en courant. Les DEL RVB peuvent, au choix, briller en différentes couleurs. En principe, trois DEL de différentes couleurs sont ici montées dans un boîtier transparent. Chacune de ces trois DEL dispose de sa propre anode, par le biais de laquelle elles sont respectivement raccordées à une broche GPIO. La cathode raccordée au câble de terre n'existe qu'une seule fois. C'est la raison pour laquelle une DEL RVB est munie de quatre fils de raccordement.

Afin de pouvoir mieux les reconnaître, les fils de raccordement des DEL RVB n'ont pas la même longueur. Contrairement aux DEL normales, la cathode a ici le fil le plus long.

Les DEL RVB fonctionnent comme trois DEL individuelles et nécessitent donc également trois résistances série.

La DEL RVB clignote avec différentes couleurs

L'expérience du quatrième jour fait clignoter la DEL RVB de différentes couleurs de manière aléatoire.

Composants : 1 tableau de connexions, 1 DEL RVB, 3 résistances 220 $\times$, 4 câbles de raccordement

Le programme

La DEL RVB est raccordée à trois broches GPIO dont les numéros se suivent. La commande est ainsi possible par le biais d'un nombre aléatoire compris entre 11 et 13. Pendant chaque boucle, une des trois broches de la DEL RVB est mise en circuit de manière aléatoire. À cet effet, le programme génère un nombre aléatoire dans la plage comprise entre 11 et 13. Deux blocs **connect** imbriqués l'un dans l'autre génèrent l'un de ces textes à partir de ce nombre : **pin11on**, **pin12on**, **pin13on**. Dans Scratch, les nombres et chaînes de caractères peuvent librement être combinés en chaînes de caractères. Avec d'autres langages de programmation, cela nécessite bien souvent une conversion préalable. Un bloc similaire commute ensuite de manière aléatoire l'une des trois broches de la DEL RVB. Comme l'état de commutation de la broche correspondante n'est pas préalablement contrôlé, cela permet de générer toutes les combinaisons imaginables de couleurs allumées et éteintes après quelques séquences de boucles.

Comment les nombres aléatoires sont-ils générés ?

D'ordinaire, on pourrait penser qu'un programme ne laisse rien au hasard. – Comment est-il alors possible qu'un programme soit en mesure de générer des nombres aléatoires ? Lorsque l'on divise un nombre premier par une valeur quelconque, on obtient, à partir de la *énième* décimale, des nombres qui ne sont plus prévisibles. En augmentant régulièrement le diviseur, ces nombres varient de manière absolument irrégulière. Ce résultat peut sembler fortuit, mais il peut à tout moment être reproduit avec un programme identique ou en exécutant plusieurs fois le même programme. Mais si l'on prend plusieurs chiffres de ce nombre compilé et que l'on le divise à nouveau par un nombre qui résulte des secondes actuelles de l'heure ou du contenu d'une plage de mémoire quelconque de l'ordinateur, on obtient un résultat non reproductible et que l'on peut donc appeler nombre aléatoire.

Jour 5

Commuer la DEL à l'aide de boutons-poussoirs

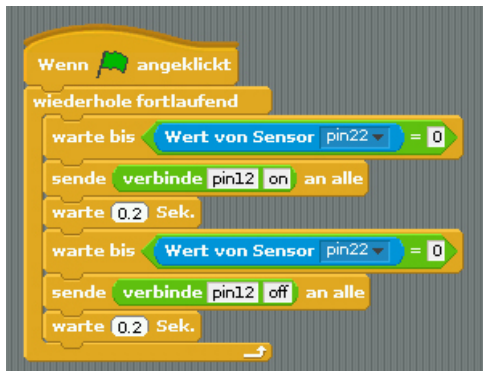
Tant que le bouton est enfoncé, un bouton-poussoir établit une liaison entre les deux barrettes à plots. Dès que le bouton est relâché, la liaison est à nouveau interrompue. Pour toujours allumer une DEL lorsque l'un des boutons-poussoirs est enfoncé, il suffit de fermer un circuit électrique et pas un programme. L'expérience du cinquième jour allume la DEL par simple pression sur le bouton et l'éteint en cas de nouvelle pression.

Composants : 1 tableau de connexions, 1 DEL rouge, 1 résistance 220 $\times$, 1 bouton-poussoir, 4 câbles de raccordement

ScratchGPIO utilise les résistances de tirage internes du Raspberry Pi ; à l'état non commuté, les entrées GPIO ont clairement la valeur 1. En cas de raccordement d'une telle entrée GPIO à GND, la valeur passe à 0. Les entrées GPIO peuvent uniquement prendre la valeur 1 ou 0.

Le programme

Au cours d'une boucle infinie, le programme attend jusqu'à ce que l'utilisateur appuie sur le bouton-poussoir. La DEL est alors allumée. Le programme attend ensuite à nouveau jusqu'à ce que l'utilisateur appuie sur le bouton-poussoir. La DEL est alors éteinte et la boucle infinie recommence à zéro.



Le programme 05taster commute la DEL à l'aide d'un bouton-poussoir

Le bloc Scratch **wait until...** met le programme en attente jusqu'à ce qu'une condition définie soit remplie. Les conditions sont des blocs avec

des crochets, la plupart d'entre eux se trouve sur la palette de blocs **operators**.

La valeur actuelle d'une entrée GPIO se contrôle à l'aide du bloc **value of sensor...** de la palette de blocs **feel**. Toutes les entrées GPIO peuvent être sélectionnées dans le champ de liste.

Après avoir allumé ou éteint la DEL, le programme attend 0,2 seconde. Ces « timeouts » ou « temporisations » en français sont toujours ajoutés lorsque des programmes communiquent directement avec le matériel. Ils permettent – en d'autres termes – d'éviter qu'un programme ne « s'emballe » et ne rate un événement matériel quelconque.

Jour 6

Piloter la DEL par capteur à contact

Pas seulement les feux tricolores, les ouvre-portes, les interrupteurs et les automates sont de nos jours souvent pilotés par capteurs à contact qu'il suffit d'actionner. Les boutons-poussoirs qui doivent réellement être enfoncés deviennent toujours plus rares. L'expérience du sixième jour pilote une DEL par le biais d'un simple capteur à contact.

Composants : 1 tableau de connexions, 1 DEL rouge, 1 résistance 220 Ω , 1 résistance 20 M Ω , 2 straps (capteurs à contact), 4 câbles de raccordement

Les deux straps représentés en gris sont réalisés au moyen de petits morceaux de fil de connexion dénudé. Le contact du haut est enfiché sur le câble de terre, le contact du bas est relié à la broche GPIO 22. Au cours des jours à venir, ce fil sera encore employé à plusieurs reprises pour réaliser des ponts de raccordement sur le tableau de connexions.

Principe de fonctionnement des capteurs à contact

La broche GPIO configurée comme entrée est raccordée à une tension +3,3 V par le biais d'une résistance à très haute impédance (20 M Ω) de sorte qu'un signal faible, mais dont le niveau High est clairement défini, soit disponible. Une personne qui ne flotte pas dans les airs est toujours reliée à la terre et délivre un niveau Low par le biais de la peau à conduction électrique. Lorsque cette personne touche un capteur à contact, le faible signal High brouille le niveau Low nettement plus puissant au bout des doigts et bascule la broche GPIO sur un niveau Low.

Le niveau réel de la résistance entre la main et la terre dépend néanmoins de nombreux facteurs, notamment des chaussures et de la nature du sol. Lorsque l'on marche pieds nus dans l'herbe humide, la liaison à la terre est optimale, mais cela fonctionne également généralement plutôt bien sur les sols en pierre. Les planchers en bois sont mieux isolés, les revêtements de sol en plastique ont même souvent une charge positive. Pour que le circuit fonctionne toujours, un contact de terre supplémentaire est installé, comme pour les touches à effleurement des ascenseurs et portes. Lorsque l'on touche simultanément ce contact et le capteur en soi, la connexion à la terre est systématiquement établie.

Le programme

Afin que les contacts du capteur fonctionnent, les résistances de tirage internes doivent d'abord être désactivées sur les broches GPIO, qui activent par défaut toujours ScratchGPIO. L'instruction GPIO **SetPinsNone** au début du programme se charge de cette tâche.

Au cours d'une boucle infinie, une requête **if...else** contrôle si la valeur du capteur **pin22** est égale à 0. La broche GPIO est alors reliée à la terre, le capteur a donc été touché.

En tel cas, la DEL sur la broche GPIO 12 est allumée. Le cas contraire, c.-à-d. lorsque le capteur n'est pas touché, elle est éteinte.

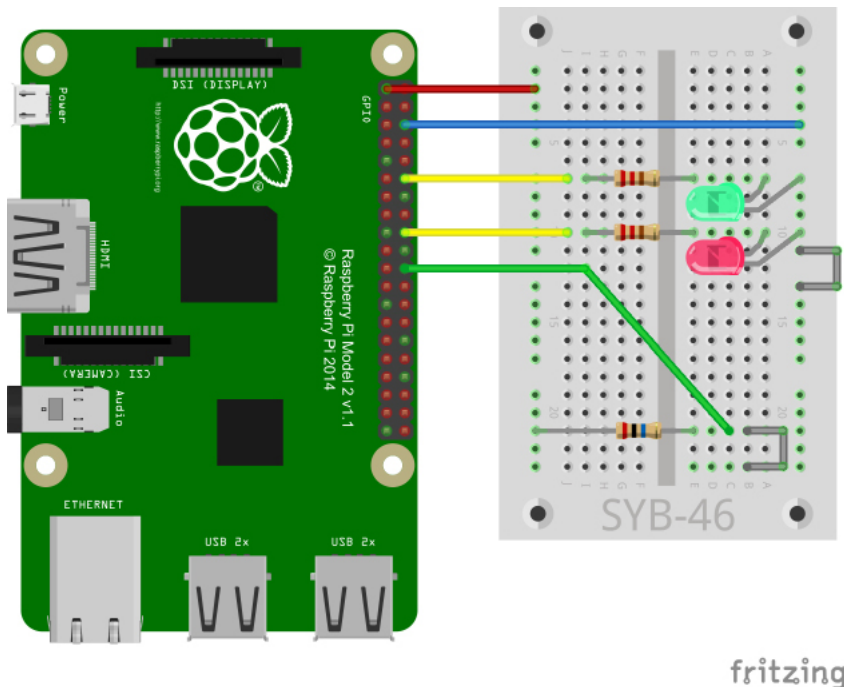
Jour 7

Cuillère en guise de capteur

Une cuillère ou un autre objet métallique constituent de bons capteurs à contact. La cuillère est reliée à un strap sur le tableau de connexions au moyen d'un câble à pinces crocodile.

En cas de contact avec la cuillère, la DEL verte s'allume au cours de l'expérience du septième jour. Tant que l'on ne touche pas la cuillère, la DEL rouge reste allumée.

Composants : 1 tableau de connexions, 1 DEL rouge, 1 DEL verte, 1 résistance 220 $\times$, 1 résistance 20 M $\times$, 2 straps (capteurs à contact), 5 câbles de raccordement, 1 câble à pinces crocodile



Le câble à pinces crocodile se raccorde au strap du bas. Le contact de terre sur le strap du haut est uniquement requis lorsque la mise à la terre est insuffisante

Le programme

Le programme est similaire au programme d'hier. Lorsque le capteur sur la broche GPIO 22 fournit la valeur 0, c.-à-d. lorsqu'il est relié à la terre, la DEL verte raccordée à la broche GPIO 18 est allumée et la DEL rouge raccordée à la broche GPIO 12 est éteinte. Dans l'autre cas de figure, la DEL rouge est allumée et la DEL verte éteinte.

Jour 8

Piloter la DEL RVB par capteur à contact

Trois capteurs à contact peuvent piloter les trois couleurs d'une DEL RVB indépendamment les unes des autres. Un contact simultané avec deux contacts génère une couleur mélangée. En cas de contact simultané avec les trois contacts, la DEL RVB vire au blanc.

Composants : 1 tableau de connexions, 1 DEL RVB, 3 résistances 220 $\times$, 3 résistances 20 M $\times$, 4 straps (capteurs à contact), 1 strap (fil de raccordement), 8 câbles de raccordement

Le programme

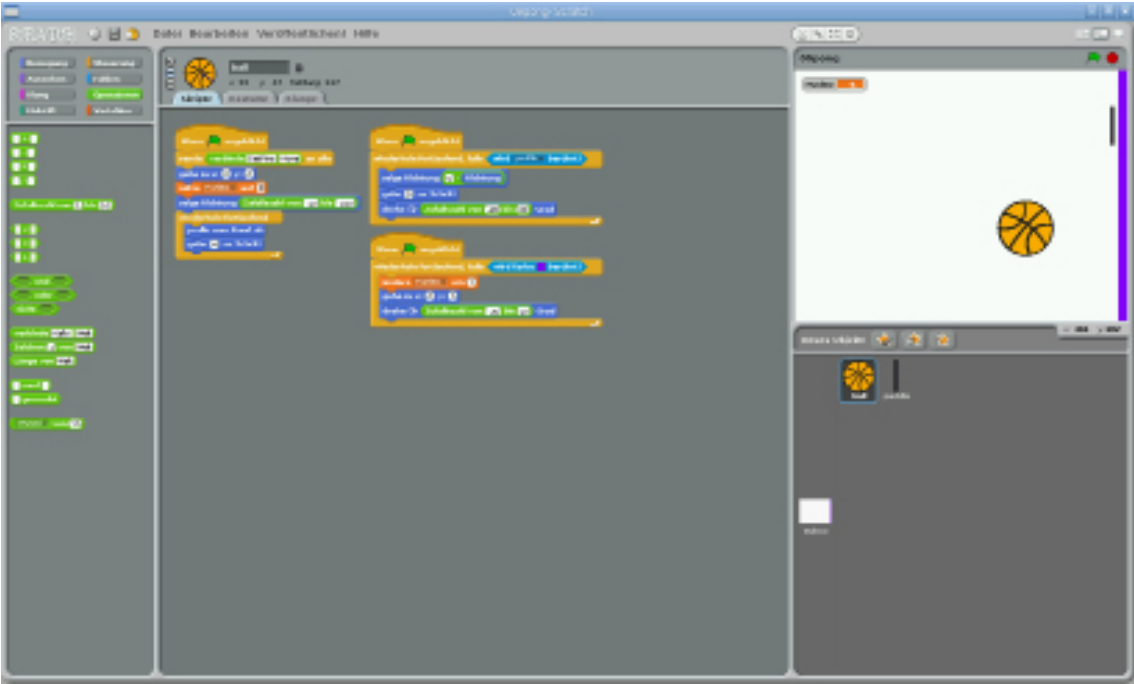
Au cours d'une boucle infinie, le programme interroge constamment les trois capteurs à contact sur les broches GPIO 11, 12 et 13. Lorsqu'un capteur fournit la valeur 0, la composante chromatique correspondante de la DEL RVB est allumée. Le cas contraire, elle est éteinte.

Jour 9

Jeu Pong avec deux capteurs à contact

Scratch ne permet bien sûr pas seulement de programmer les DEL. À l'origine, le langage de programmation était prévu pour la compilation de jeux simples sur l'écran. La version actuelle du programme pilote un jeu Pong simple au look rétro avec deux objets métalliques, qui sont rac-

cordés aux straps sur le tableau de connexions au moyen du câble à pinces crocodile.



Le jeu Pong et les scripts Scratch pour la balle

L'objectif de ce jeu est de renvoyer la balle qui vole dans l'espace à l'aide de la raquette. Lorsque la balle touche la ligne de couleur, le joueur reçoit un point négatif et la balle redémarre au centre de l'écran. Deux capteurs à contact permettent de déplacer la raquette vers le haut et le bas. Les DEL signalisent les contacts des capteurs.

Le programme pour la balle

Les deux objets du jeu, la balle et la raquette, disposent respectivement de leurs propres scripts Scratch, qui démarrent tous lorsque vous cliquez sur le drapeau vert.

Au début du jeu, la balle est placée au centre de l'estrade et le compteur de points est remis à 0. La balle commence dans un sens aléatoire avec un angle compris entre -20 et 160 degrés. Elle vole ensuite au cours d'une boucle infinie et change uniquement de direction lorsqu'elle rebondit contre le bord.



Ce script pilote le mouvement de la balle en cas de contact avec la raquette

Lorsque la balle touche la raquette, le sens de déplacement est inversé en sens négatif. La balle poursuit son déplacement avec le même angle en bas à gauche avec lequel elle est arrivée en haut à gauche ou, à l'inverse, elle poursuit son déplacement en haut à gauche lorsqu'elle est arrivée d'en bas à gauche. La balle poursuit encore brièvement son vol afin de ne toucher en aucun cas la raquette. Afin de rendre le déplacement légèrement plus prévisible, la direction de vol est inversée avec une valeur aléatoire comprise entre -20 et 20 degrés.

Lorsque la balle touche la barre violette, le joueur reçoit un point négatif. La variable **points** est augmentée de 1. La balle est ensuite à nouveau positionnée au centre du terrain avant que le jeu ne redémarre. La direction de vol est tournée avec une valeur aléatoire comprise entre -20 et 20 degrés afin que la trajectoire de la balle ne soit pas à nouveau exactement identique, mais que sa direction corresponde approximativement à la direction dans laquelle elle volait auparavant.

L'électronique pour la raquette

Composants : 1 tableau de connexions, 1 DEL rouge, 1 DEL jaune, 2 résistances 220 $\times$, 2 résistances 20 M $\times$, 3 straps (capteurs à contact), 6 câbles de raccordement, 2 câbles à pinces crocodile

Le programme pour la raquette

La raquette dispose de son propre script Scratch qui démarre lorsque vous cliquez sur le drapeau vert et qui interroge et pilote l'électronique.



Ce script pilote la raquette

Les deux capteurs sur les broches GPIO 24 et 26 sont interrogées au cours d'une boucle infinie. En cas de contact avec un capteur, la DEL correspondante est brièvement allumée pendant 0,1 seconde. Le déplacement de la raquette de 20 incréments de coordonnées vers le haut ou le bas est bien plus important que le scintillement bref de la DEL.

La raquette ne doit pas poursuivre sa course lorsqu'elle atteint le bord supérieur. Nous employons ici une requête **if** qui contrôle si la position sur l'axe des y est supérieure à 200. Le cas échéant, la position sur l'axe des y est simplement fixée à 200. Selon le même principe, la position est fixée à -200 lorsque la raquette atteint le bord inférieur.

Jour 10

Piloter le feu clignotant à l'aide de pâte à modeler

La pâte à modeler est un aussi bon conducteur que la peau humaine. Elle peut facilement être modelée dans tous les formes et un contact en pâte à modeler peut bien plus facilement être saisi qu'un morceau de fil. La surface avec laquelle la peau touche le contact est nettement plus grande. Cela provoque bien moins souvent un « faux contact ».



Un capteur en pâte à modeler peut être fabriqué à l'aide d'un morceau de pâte à modeler, d'un fil de fer et d'un câble à pinces crocodile.

Composants : 1 tableau de connexions, 1 DEL rouge, 1 résistance 220 $\times$, 1 résistance 20 M $\times$, 2 straps (capteurs à contact), 4 câbles de raccordement, 1 câble à pinces crocodile, 1 contact en pâte à modeler

Pendant l'expérience du dixième jour, le capteur à contact n'allume ou n'éteint pas simplement la DEL, elle la fait clignoter à une vitesse réglable tant que vous touchez le capteur en pâte à modeler.

Le programme

Le programme utilise une variable qui est représentée sur l'estrade par un curseur de réglage. Ce curseur permet de modifier la fréquence de clignotement pendant l'exécution du programme.



Le programme 10blinklicht pilote un feu clignotant à l'aide d'un capteur en pâte à modeler

Lors du démarrage du programme en cliquant sur le drapeau vert, la variable *z*, qui régule la vitesse de clignotement, est définie à 0.1. Une boucle infinie interroge le capteur sur la broche GPIO 24. En cas de contact avec ce capteur, la DEL s'allume pour la durée définie dans la variable *z* et reste ensuite éteinte pendant la même durée.

En cas de contact prolongé avec le capteur en pâte à modeler, un effet de clignotement est généré. À la fin de la boucle, la DEL est toujours éteinte. Elle reste ainsi éteinte en l'absence de contact avec le capteur en pâte à modeler.

Jour 11

Feu pour piétons avec feu clignotant

Dans certains pays, les feux aux passages piétons ne sont pas rouges et verts, mais clignotent lorsque le feu tricolore pour les voitures est rouge et que les piétons peuvent traverser la route.

Composants : 1 tableau de connexions, 1 DEL rouge, 1 DEL jaune, 1 DEL verte, 1 DEL bleue, 4 résistances 220 $\times$, 1 bouton-poussoir, 1 straps (fils de raccordement), 6 câbles de raccordement

Le programme

Le programme pilote le cycle du feu dans une boucle infinie. Lorsque le feu tricolore passe au vert, la boucle attend que la touche soit enfoncée. Le cycle du feu se poursuit ensuite jusqu'à ce que le feu tricolore passe au rouge. La DEL bleue scintille ensuite 20 fois. Après la mise hors circuit de cette DEL, aucun temps d'attente n'est implémenté. La DEL reste éteinte un court instant afin que le script puisse démarrer la séquence de la prochaine boucle. Contrairement au clignotement régulier durant les projets antérieurs, cela produit un scintillement.



Le programme 11fussgaengerblau pilote un feu pour piétons avec feu clignotant

Jour 12

Feu pour piétons avec feu rouge / vert

L'expérience du douzième jour est une autre version du feu pour piétons avec laquelle une DEL RVB génère le signal rouge ou vert pour les piétons.

Composants : 1 tableau de connexions, 1 DEL rouge, 1 DEL jaune, 1 DEL verte, 1 DEL RVB, 5 résistances 220 Ω, 1 bouton-poussoir, 2 straps (fils de raccordement), 7 câbles de raccordement

Le programme

Le programme fonctionne de la même manière que la version précédente, mais active en plus la broche GPIO 15 sur laquelle la couleur rouge de la DEL RVB est raccordée. Lorsque le feu tricolore pour les voitures passe au rouge, la couleur rouge de la DEL RVB sur la broche GPIO 15 est éteinte pendant 2 secondes et la couleur verte de la DEL RVB sur la broche GPIO 18 est alors allumée.



Le programme 12fussgaengerrot pilote un feu pour piétons avec DEL RVB

Jour 13

Effets lumineux RVB

L'expérience du treizième jour permet de faire clignoter deux DEL RVB en alternance. Les couleurs peuvent être activées de manière interactive sur l'écran.

Composants : 1 tableau de connexions, 2 DEL RVB, 6 résistances 220 Ω , 1 strap (fil de raccordement), 7 câbles de raccordement

Le programme

Le programme montre une autre méthode pour la définition des broches de sortie dans ScratchGPIO. Il emploie trois variables, une pour chaque couleur. Les variables r, g et b sont représentées sur l'estrade par des boutons de réglage. L'utilisateur peut ici activer ou désactiver les trois composantes chromatiques de manière interactive. À cela s'ajoutent six variables pour les broches employées des deux DEL RVB. Ces variables sont définies aux valeurs correspondantes.

Au cours d'une boucle infinie, les trois broches GPIO 11, 12 et 13 de la première DEL RVB sont définies aux valeurs chromatiques configurées et les trois broches GPIO 16, 18 et 15 de la seconde DEL RVB sont désactivées. Après un bref temps d'attente, la première DEL RVB est éteinte et les valeurs chromatiques configurées sont attribuées à la seconde. La commutation en alternance des DEL génère un clignotement en alternance.



Le programme 13rgb2 fait clignoter deux DEL RVB en alternance

Jour 14

Crabes rampants

L'expérience du quatorzième jour est un jeu rapide qui se joue sur l'écran à l'aide d'une manette de jeu à deux boutons-poussoirs. Durant ce jeu, un crabe court sur un parcours en rond et le joueur doit veiller à ne pas quitter la trajectoire prédéfinie.

Le crabe avance tout seul vers l'avant. Les touches permettent de tourner le crabe afin de modifier la direction dans laquelle il rampe.

Composants : 1 tableau de connexions, 2 boutons-poussoirs, 1 strap (fil de raccordement), 3 câbles de raccordement

Le programme

Un clic sur le drapeau vert place le crabe en position initiale et le compteur d'erreurs est remis à 0. Une boucle infinie démarre ensuite et fait avancer le crabe d'un petit pas à chaque passage.

Le programme contrôle ensuite si la pince rouge ou verte du crabe touche le fond blanc. En tel cas, le crabe a quitté sa trajectoire prédéfinie et un point d'erreur est attribué.

Deux autres requêtes contrôlent si l'une des deux touches est actionnée. Le cas échéant, le crabe est tourné de 5 degrés dans la direction correspondante à chaque pression sur une touche.

Jour 15

Lumière défilante

Les lumières défilantes sont toujours des effets populaires, pas seulement pour la publicité et les salles de fêtes. L'expérience du quinzième jour allume sept DEL sous forme de lumière défilante. Les trois DEL RVB ne sont respectivement raccordées qu'avec une seule couleur.

Composants : 1 tableau de connexions, 1 DEL rouge, 1 DEL jaune, 1 DEL verte, 1 DEL bleue, 3 DEL RVB, 7 résistances 220 $\times$, 3 straps (fils de raccordement), 8 câbles de raccordement

Le programme

Au cours d'une boucle infinie, les DEL sont successivement allumées pendant un court instant. La durée est enregistrée dans la variable z et peut être réglée de manière interactive au moyen d'un bouton de réglage sur l'écran.



Le programme 151auf1icht fait clignoter sept DEL en alternance sous forme de lumière défilante

Jour 16

Piloter la DEL RVB par capteur à contact

Trois capteurs à contact peuvent piloter les trois couleurs d'une DEL RVB indépendamment les unes des autres. Un contact simultané avec deux contacts génère une couleur mélangée. En cas de contact simultané avec les trois contacts, la DEL RVB vire au blanc.

Composants : 1 tableau de connexions, 1 DEL RVB, 3 résistances 220 $\times$, 3 résistances 20 M $\times$, 4 straps (capteurs à contact), 1 strap (fil de raccordement), 8 câbles de raccordement, 3 câbles à pinces crocodile, 3 contacts en pâte à modeler

Trois contacts en pâte à modeler sont raccordés aux trois straps à l'aide des câbles à pinces crocodile. Le strap complètement en haut sur la figure, le contact de terre, n'est requis que dans certains cas.

Le programme

Le programme repose sur un programme plus ancien avec lequel les valeurs RVB sont enregistrées dans des variables et écrites dans les variables associées aux broches GPIO par le biais de blocs **set...to**.



Le programme 16rgb2-sensor fait clignoter une DEL RVB

Un deuxième bloc de scripts, qui démarre également lorsque vous cliquez sur le drapeau vert, interroge trois capteurs en pâte à modeler et définit en conséquence les variables r, g et b.



Le deuxième bloc de scripts interroge les capteurs

Jour 17

Course automobile

Deux boutons-poussoirs permettent de piloter une voiture au cours d'une course simple. Tant que la voiture reste sur la route, la DEL verte est allumée sur le contrôleur de jeu. Si la voiture dévie de la route vers la gauche ou la droite, l'une des deux diodes rouges s'allume.

Composants : 1 tableau de connexions, 2 DEL rouges, 1 DEL verte, 3 résistance 220 $\times$, 2 boutons-poussoirs, 2 straps (fils de raccordement), 6 câbles de raccordement

Le programme

Chaque fois que la voiture atteint le bord supérieur de l'estrade, un nouveau circuit s'affiche et la voiture redémarre sur le bord inférieur de l'estrade.

La voiture se pilote au moyen de deux blocs de scripts. Le premier bloc se compose de deux boucles infinies imbriquées l'une dans l'autre. La boucle extérieure place la voiture sur le bord inférieur de l'estrade, attend jusqu'à ce qu'elle soit arrivée en haut et envoie alors un message **background** auquel le décor réagit et change.

La boucle intérieure déplace la voiture tant qu'elle ne touche pas le bord et contrôle à chaque déplacement si une roue noire de la voiture touche le pré vert clair à gauche de la route. Le cas échéant, la DEL rouge gauche s'allume et la DEL verte s'éteint. Si la roue ne touche pas ce pré, une autre interrogation contrôle si la roue touche le pré vert foncé à droite de la route. En tel cas, la DEL rouge droite s'allume et la DEL verte s'éteint également. Si cela n'est pas non plus le cas, cela signifie que la voiture se trouve sur la route et la DEL verte est allumée.

L'estrade comporte encore deux scripts simples. Au début, lorsque vous cliquez sur le drapeau vert, la première image de fond est activée et, à chaque fois que le message **background** est reçu, le prochain décor s'affiche.

Jour 18

Effets lumineux

Les broches GPIO peuvent également piloter plus d'une DEL, chaque DEL devrait alors toutefois être équipée de sa propre résistance série. L'expérience du dix-huitième jour montre un effet lumineux sur six DEL, qui ne nécessitent que trois broches GPIO.

Composants : 1 tableau de connexions, 2 DEL rouges, 2 DEL jaunes, 1 DEL verte, 1 DEL bleue, 6 résistances 220 $\times$, 3 straps (fils de raccordement), 4 câbles de raccordement

Le programme

Les broches GPIO sont successivement activées au cours d'une boucle infinie. Contrairement à la lumière défilante, elles restent activées et ne sont éteintes qu'à la fin simultanément. La durée entre les cycles de commutation est enregistrée dans la variable *z* et peut être réglée de manière interactive au moyen d'un bouton de réglage sur l'écran.



Le programme 18lichteffekt
pilote l'effet lumineux sur
trois broches GPIO

Jour 19

Match de tennis

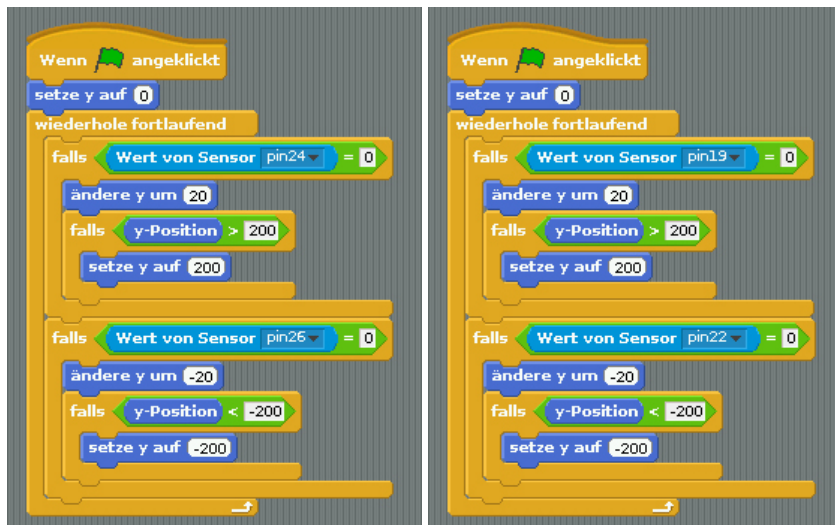
L'expérience du dix-neuvième jour est un grand classique de jeu qui remonte à la dernière ère des ordinateurs personnels. Deux joueurs tentent de frapper une balle sur le terrain. Un point est attribué à chaque fois qu'un joueur touche la ligne de fond de l'adversaire. Le programme repose sur le jeu Pong que nous avons examiné il y a quelques jours.

Composants : 1 tableau de connexions, 4 résistances 20 M $\times$, 5 straps (capteurs à contact), 6 câbles de raccordement, 4 câble à pinces crocodile, 4 contacts en pâte à modeler

Le programme

Comme avec le jeu Pong, le script pour la balle la laisse rebondir contre les bords. Lorsque la balle touche l'une des deux raquettes, la direction de vol est aussi légèrement modifiée de manière aléatoire.

Deux variables enregistrent les points des joueurs. Chaque joueur reçoit un point lorsque la balle touche la ligne de fond de l'adversaire. Les deux raquettes comportent des blocs de programmes, qui interrogent respectivement deux capteurs à contact dans des boucles infinies et qui déplacent la raquette correspondante vers le haut ou le bas en cas de contact.



Jour 20

Jeu de dés à DEL

Tout le monde connaît et a déjà joué à des jeux de dés typiques avec 1 à 6 points. Un dé à commande électronique qui allume les points à chaque pression sur une touche est bien plus cool - les 1 à 6 DEL ne sont toutefois pas simplement allumées en série, elles doivent respecter la disposition des points sur le dé. Le nombre de points sur le dé reste allumé jusqu'à ce que le dé soit encore une fois jeté.

Composants : 1 tableau de connexions, 2 DEL rouge, 2 DEL jaune, 2 DEL verte, 1 DEL bleue, 7 résistances 220 $\times$, 1 bouton-poussoir, 6 straps (fils de raccordement), 6 câbles de raccordement

Le programme

Une boucle infinie attend d'abord que le bouton-poussoir sur la broche 26 soit enfoncé. Les quatre broches GPIO employées sont maintenant désactivées et le résultat actuellement affiché pour le dé est effacé.

Un nombre aléatoire compris entre 1 et 6 est généré puis enregistré dans la variable *w*. Après avoir jeté le dé, six blocs **if** suivent pour chaque valeur possible. Lorsque la valeur définie est obtenue, chacun de ces blocs allume la combinaison correspondante des DEL.

Indépendamment du résultat du dé, le programme attend toujours encore une demi-seconde afin d'éviter que deux différentes actions du dé ne soient déclenchées rapidement l'une après l'autre en cas d'actionnement multiple de la touche.



Le programme 20wuerfe1 pilote le dé à DEL

Jour 21

Pupitre de commande lumineux RVB

L'expérience du vingt-et-unième jour propose différents effets lumineux avec trois DEL RVB. Les couleurs et effets peuvent être activés de manière interactive sur l'écran.

Composants : 1 tableau de connexions, 3 DEL RVB, 9 résistances 220 Ω , 2 strap (fil de raccordement), 10 câbles de raccordement

Le programme

Le programme emploie cinq variables qui peuvent être configurées à l'aide des boutons de réglage sur l'estrade. Les variables r , g et b permettent à l'utilisateur d'activer et de désactiver les trois composantes chromatiques. La variable z définit le temps d'attente entre le clignotement de deux DEL consécutives. La variable e définit le temps à la fin d'une boucle jusqu'à ce que la première DEL se rallume.

Au cours d'une boucle infinie, les trois broches GPIO d'une DEL RVB sont successivement activées avec la valeur chromatique configurée et les trois broches GPIO de la DEL RVB précédemment allumée sont désactivées.



Le programme 21rgb3 pilote l'effet lumineux sur trois DEL RVB

Jour 22

Le calendrier de l'Avent contient aujourd'hui le manuel pour le pack d'expérimentation Conrad « Comprendre et appliquer Raspberry Pi » sous forme de livre numérique. Ce livre contient différentes expériences avec le Raspberry Pi, qui sont réalisées avec le langage de programmation Python.

La plupart des composants employés sont déjà fournis avec le calendrier de l'Avent, les autres composants sont des composants courants qui se trouvent dans les boîtes de bricolage des amateurs d'électronique ou peuvent être commandés sur le site

www.conrad.de.

Dé à effets spéciaux

Un dé électronique qui affiche simplement un chiffre n'est pas très réaliste. L'expérience du vingt-deuxième jour affiche brièvement quatre autres chiffres avant d'afficher le résultat final.

Composants : 1 tableau de connexions, 2 DEL rouges, 2 DEL jaunes, 2 DEL vertes, 1 DEL bleue, 7 résistances 220 $\times$, 1 bouton-poussoir, 6 straps (fils de raccordement), 6 câbles de raccordement

Le programme

Le programme se compose de deux blocs. Après avoir cliqué sur le drapeau vert, une boucle infinie dans le programme principal attend que la touche soit enfoncée. Un message **roll** est ensuite envoyé cinq fois de suite et exécute à chaque fois le bloc de programme pour l'affichage d'un résultat aléatoire du dé. L'intervalle entre le nombre de points affichés augmente à chaque fois. Cela ressemble ainsi à un vrai dé qui roule lentement avant de s'immobiliser.

Jour 23

Méodies de Noël

L'expérience du vingt-troisième jour un petit piano simple, qui permet de jouer cinq notes à l'aide des contacts en pâte à modeler. Le sixième contact, le câble de terre, est uniquement employé en cas de besoin.

Composants : 1 tableau de connexions, 5 résistances 20 M $\times$, 6 straps (capteurs à contact), 7 câbles de raccordement, 5 câble à pinces crocodile, 5 contacts en pâte à modeler

Le Raspberry Pi peut bien sûr jouer des notes quelconques. Afin que les cinq notes qui peuvent être jouées à l'aide des cinq contacts en pâte à modeler disponibles permettent de jouer une mélodie harmonieuse, nous employons le système pentatonique, une gamme musicale qui ne compte que cinq notes, - il s'agit ici d'ailleurs de la gamme musicale la plus ancienne au monde, qui était déjà employée 3 000 ans avant J.-C. par les joueurs de flûte en os.

Comme le cerveau humain peut facilement se souvenir de cinq notes, la gamme pentatonique avec les notes Do, Ré, Mi, Fa, Sol est fréquemment employée dans les chansons enfantines et les jingles publicitaires. Quelques exemples :

Backe, backe, Kuchen (G-G-A-A-G-E)

Laterne, Laterne, Sonne, Mond und Sterne (A-G-E, A-G-E, G-G-A-A-G-E)

Old Mac Donald had a farm, hea hea ho ! (C-C-C-G-A-A-G, E-E-D-D-C)

Le programme

Une boucle infinie interroge successivement tous les cinq capteurs à contact. En cas de contact avec un capteur, la tonalité correspondante retentit. Le Raspberry Pi peut jouer de la musique par le biais d'un moniteur HDMI, d'un haut-parleur externe ou d'écouteurs branchés sur la prise jack 3,5 mm.

Si aucun son n'est audible...

... ouvrir `sudo raspi-config` dans une fenêtre LXTerminal puis sélectionner manuellement la bonne sortie audio sous **Advanced Options/Audio**.

Jour 24

Décoration de l'arbre de Noël

L'expérience du vingt-quatrième jour est une décoration clignotante pour l'arbre de Noël.

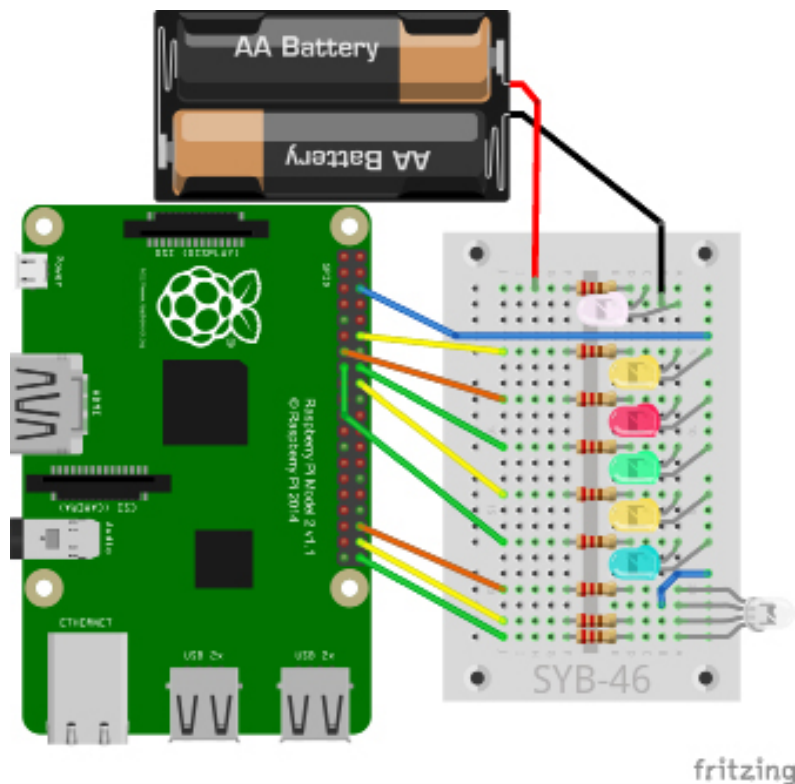
Composants : 1 tableau de connexions, 1 DEL rouge, 2 DEL jaunes, 1 DEL verte, 1 DEL bleue, 1 DEL RVB, 1 DEL clignotante, 9 résistances 220 Ω , 1 strap (fil de raccordement), 10 câbles de raccordement

Le programme

Le clignotement des cinq DEL et de la DEL RVB est piloté de manière entièrement aléatoire par le programme. La DEL clignotante qui se cache derrière la porte du jour du calendrier de l'Avent clignote même lorsqu'aucun programme n'est exécuté. Le programme utilise une liste dans laquelle les numéros des broches GPIO des DEL employées sont enregistrés. Dans Scratch, les variables à liste peuvent avoir une longueur quelconque et contenir des chiffres ou des chaînes de caractères. Après avoir cliqué sur le drapeau vert, tous les éléments de liste sont d'abord effacés car la liste contiendrait déjà des valeurs dans la mesure où le programme aurait déjà été exécuté. Les numéros des broches des cinq DEL ainsi que les trois couleurs sont ensuite enregistrés les uns après les autres dans la liste. Une boucle infinie active ensuite une broche GPIO sélectionnée de manière aléatoire dans la liste ainsi qu'une autre broche GPIO sélectionnée de manière aléatoire. Le programme ne contient aucune temporisation. Comme toutes les broches sont sélectionnées de manière aléatoire, les DEL clignotent pendant différentes durées. La DEL RVB s'allume dans les couleurs mélangées de manière aléatoire.

DEL clignotante sans Raspberry Pi

Une alimentation électrique supplémentaire de 3 V (par ex. deux piles 1,5 V) permet de faire clignoter la DEL clignotante sans que le Raspberry Pi ne soit en marche.



Décoration de l'arbre de Noël mit avec DEL clignotantes et alimentation électrique externe

Dans ce circuit, la DEL clignotante est raccordée à sa propre alimentation électrique, indépendamment du Raspberry Pi. Elle peut toujours clignoter, même lorsque le Raspberry Pi n'est pas en marche. Comme le programme n'influence pas la DEL clignotante, il n'est pas nécessaire de le modifier.