

CE

CONRAD

Calendrier de l'Avent 2018 Raspberry Pi	3	13ème Jour	27
1er Jour	4	Aujourd'hui dans le calendrier de l'Avent	27
Aujourd'hui dans le calendrier de l'Avent	4	Résistance	27
Préparer le Raspberry Pi	4	Capteur de contact en pâte à modeler	27
Installation du système d'exploitation en bref	4	Le LED rouge clignote au hasard	28
Les principaux éléments en bref	5	Le programme	28
Platine enfichable	5	14ème Jour	30
LEDs	5	Aujourd'hui dans le calendrier de l'Avent	30
Câble de raccordement GPIO	5	Effets de lumière LED RVB	30
Un LED s'allume	6	Le programme	30
2ème Jour	7	15ème Jour	32
Aujourd'hui dans le calendrier de l'Avent	7	Aujourd'hui dans le calendrier de l'Avent	32
La première LED s'allume sur la crèche de Noël	7	Commander un chenillard avec un contact de la pâte à modeler	32
Réglages de base pour Scratch	7	Le programme	32
Le programme	8	16ème Jour	34
3ème Jour	9	Aujourd'hui dans le calendrier de l'Avent	34
Aujourd'hui dans le calendrier de l'Avent	9	Le Raspberry Pi émet des sons	34
Deux LEDs clignent en alternance	9	Le programme	34
Le programme	9	17ème Jour	36
4ème Jour	10	Aujourd'hui dans le calendrier de l'Avent	36
Aujourd'hui dans le calendrier de l'Avent	10	Chenillard étendu	36
Une étoile allume deux LEDs en alternance	10	Le programme	36
Le programme	10	18ème Jour	37
5ème Jour	13	Aujourd'hui dans le calendrier de l'Avent	37
Aujourd'hui dans le calendrier de l'Avent	13	Trois modèles différents de chenillard	37
Commuter les LEDs avec le bouton	13	Le programme	37
Le programme	13	19ème Jour	40
6ème Jour	15	Aujourd'hui dans le calendrier de l'Avent	40
Aujourd'hui dans le calendrier de l'Avent	15	Chenillard ou clignotant	40
Commuter les LEDs avec pupitre de commande Scratch	15	Le programme	40
Le programme	15	20ème Jour	42
7ème Jour	17	Aujourd'hui dans le calendrier de l'Avent	42
Aujourd'hui dans le calendrier de l'Avent	17	Commander des jeux de couleur rouge avec capteur de pâte à modeler	42
Chenillard avec 3 LEDs	17	Le programme	43
Le programme	17	21ème Jour	44
8ème Jour	19	Aujourd'hui dans le calendrier de l'Avent	44
Aujourd'hui dans le calendrier de l'Avent	19	Jeux de couleur sur l'écran	44
Câble de connexion	19	Le programme	45
Variation des LEDs	19	22ème Jour	46
Le programme	20	Aujourd'hui dans le calendrier de l'Avent	46
9ème Jour	21	Contrôler indépendamment les deux LEDs RVB	46
Aujourd'hui dans le calendrier de l'Avent	21	Le programme	46
Chenillard avec bouton	21	23ème Jour	48
Le programme	21	Aujourd'hui dans le calendrier de l'Avent	48
10ème Jour	22	Crèche de Noël lumineuse et colorée	48
Aujourd'hui dans le calendrier de l'Avent	22	Le programme	48
Commuter des LEDs avec deux boutons	22	24ème Jour	50
Le programme	22	Aujourd'hui dans le calendrier de l'Avent	50
11ème Jour	24	Chants de Noël sur le Raspberry Pi	50
Aujourd'hui dans le calendrier de l'Avent	24	Le programme	50
LEDs RVB	24	Aperçu	51
Jeux de couleur RVB	24		
Le programme	25		
12ème Jour	26		
Aujourd'hui dans le calendrier de l'Avent	26		
Mélange de couleur avec LEDs rouge	26		
Le programme	26		

Calendrier de l'Avent 2018 Raspberry Pi

Ce calendrier de l'Avent vous réserve chaque jour une expérience matérielle avec votre Raspberry Pi. Jour après jour, une crèche de Noël prend forme qui resplendira le 24 décembre. Les expériences sont programmées avec Scratch. Ce langage de programmation est installé sur le Raspberry Pi. Toutes les expériences fonctionnent avec le Raspberry Pi 3 et le Raspberry Pi 3 B+.

Commander un circuit électronique simple avec un PC normal ou même un ordinateur portable - même s'il ne s'agit que de quelques voyants LED - implique des efforts peu raisonnables pour le programmeur amateur. Le PC ne dispose tout simplement pas des interfaces nécessaires. En outre, le système d'exploitation Windows n'est pas très indiqué pour communiquer avec un système électronique.

Même s'il n'en donne pas l'impression à première vue, le Raspberry Pi est un ordinateur à part entière. Il est certes plus lent que les PC modernes habituels, mais le Raspberry Pi est aussi bien plus petit et surtout plus abordable qu'un PC.

1er Jour

Aujourd'hui dans le calendrier de l'Avent

- 1 platine enfichable
- 1 LED jaune avec résistance en série
- 2 câbles de raccordement GPIO (court)

Préparer le Raspberry Pi

Pour mettre en service le Raspberry Pi, vous avez besoin de :

- Clavier USB et souris
- Câble HDMI pour moniteur
- Câble de réseau
- Une carte MicroSD avec système d'exploitation avec OS Raspbian stretch
- Chargeur de téléphone portable micro-USB avec bloc d'alimentation (au moins 1.500 mA)

Le bloc d'alimentation doit être connecté en dernier, car avec lui, le Raspberry Pi s'allume automatiquement. Il n'y a aucun interrupteur marche / arrêt propre.

Installation du système d'exploitation en bref

Pour tous ceux qui n'ont pas encore leur Raspberry Pi opérationnel avec la version Raspian actuelle, voici l'installation du système en onze étapes :

Téléchargez NOOBS (au moins version 2.4.5, pour Raspberry Pi 3 B+ au moins 2.7.0) de www.raspberrypi.org/downloads sur votre PC et décompressez l'archive zip sur le disque dur.

Si la carte SD est déjà utilisée, reformatez avec le formateur SD du PC : www.sdcard.org/downloads/formatter_4. En même temps allumez **Format Size Adjustment** (la carte SD doit avoir au moins 8 Go).

Copiez toutes les données et sous-répertoires de NOOBS sur la carte SD.

Retirez la carte SD du PC, insérez-la dans le Raspberry Pi et le démarrage. En bas choisissez le **français** comme la langue d'installation. Ainsi, le clavier français est automatiquement sélectionné.

Cochez le système d'exploitation Raspberry présélectionné et cliquez sur **installer** en haut à gauche. Après confirmation d'un message de sécurité que la carte mémoire est écrasée, l'installation commence et dure quelques minutes.

Une fois l'installation terminée, le Raspberry Pi redémarre.

Dans le menu, démarrer l'outil **Raspberry Pi Configuration** dans **paramètres**.

Sous l'onglet **Localisation** dans le champ **déterminez le fuseau horaire**, choisissez entre les options **Europe** et **Berlin**. **L'environnement linguistique** et le **clavier** doivent être réglés automatiquement sur le français. Si tel n'est pas le cas, sélectionnez la localisation française et le clavier.

Pour utiliser la WLAN sur un Raspberry Pi 3 B, sélectionnez via le bouton **WiFi Pays** l'option **FR-France**. Le WLAN sur le Raspberry Pi 3 fonctionne encore indépendamment de ce paramètre.

Sur les **interfaces** de l'onglet réglez le commutateur **SSH** sur **activé**, si vous souhaitez transférer des données depuis le PC sur le réseau Raspberry Pi.

Cliquez sur **OK** et redémarrez le Raspberry Pi via la commande de menu **Shutdown**. Un éventuel avertissement en raison d'un mot de passe non sécurisé peut être ignoré.

Touche Num-Lock

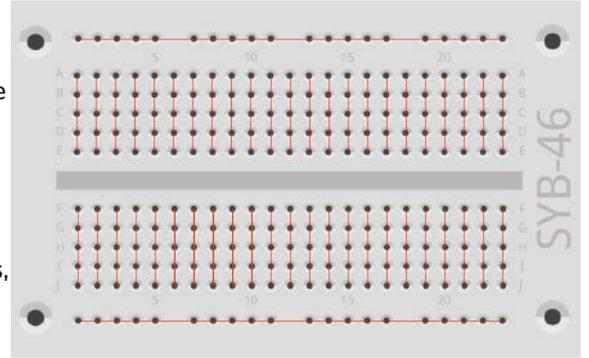
Comme avec presque tous les systèmes Linux, le clavier de chiffres est désactivé par défaut, même avec le lancement de Raspbian. Appuyez sur la touche Num-Lock du clavier pour l'allumer.

Les principaux éléments en bref

Platine enfichable

Pour le montage rapide des circuits électroniques, sans devoir souder, le calendrier de l'Avent contient une platine enfichable. Vous pouvez enficher directement les composants électroniques peuvent être insérés directement dans une trame perforée.

Toutes les lignes longitudinales externes de contacts (X et Y) sont liées dans cette maquette. Les rangées de ces contacts sont souvent utilisées comme pôle positif et négatif pour l'alimentation électrique des circuits. Dans les autres rangées de contacts, cinq contacts (A à E et F à J) sont respectivement liés transversalement, avec un espace au milieu de la platine. Il est ainsi possible de brancher des composants plus gros et de les câbler vers l'extérieur.



Les connexions sur la platine enfichable.

LEDs

Les LEDs (en français : diodes électroluminescentes) s'allument quand le sens du débit de courant les traverse. Les LEDs sont représentés dans les circuits avec le symbole d'un triangle en forme de flèche qui indique le sens de débit du pôle positif vers le pôle négatif ou la conduite de terre. Un LED laisse passer presque autant de courant que vous le souhaitez dans le sens du débit, mais il a en revanche une très faible résistance. Pour limiter le flux d'écoulement et empêcher ainsi que le LED ne grille, on monte habituellement une résistance en série de 220 Ohm entre les broches GPIO utilisées et l'anode du LED ou entre la cathode et la broche de masse. Cette résistance en série protège aussi la sortie GPIO du Raspberry Pi contre des courants élevés. Les LEDs dans le calendrier de l'Avent sont déjà dotés de la résistance série et peuvent donc être connectés directement aux broches GPIO.

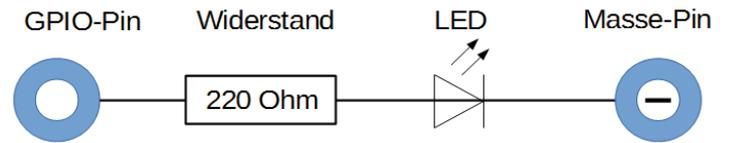


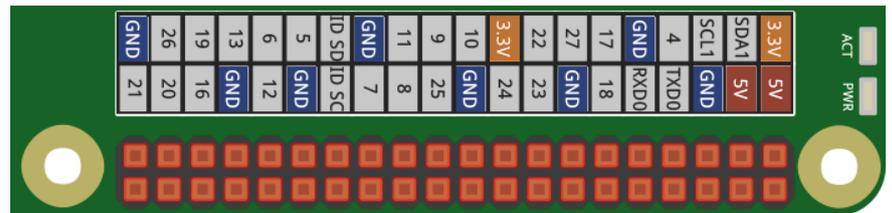
Schéma de commutation d'un LED avec résistance en série.

Dans quelle direction connecter le LED ?

Les deux fils de connexion d'un LED ont une longueur différente. Le plus long est le pôle positif, l'anode, le plus court est le pôle négatif, la cathode. Facile à retenir : Le signe Plus a un trait en plus du signe moins et le fils est donc optiquement un peu plus long. En outre la plupart des LEDs sont aplatis sur le côté négatif, comparable à un signe moins. Aussi facile à retenir : Cathode = court = bordure.

Câble de raccordement GPIO

Les câbles de connexion de couleur sont tous munis d'une fiche d'un côté, de l'autre côté d'une prise enfichable qui correspond à une broche GPIO du Raspberry Pi. Les LEDs sur la crèche de Noël sont branchés directement dans ces prises. Les fiches sont branchées sur la platine enfichable. Les broches GPIO programmables sur le Raspberry Pi ont des numéros, les broches de masse sont caractérisées par GND sur l'illustration.



Affectation des broches GPIO.

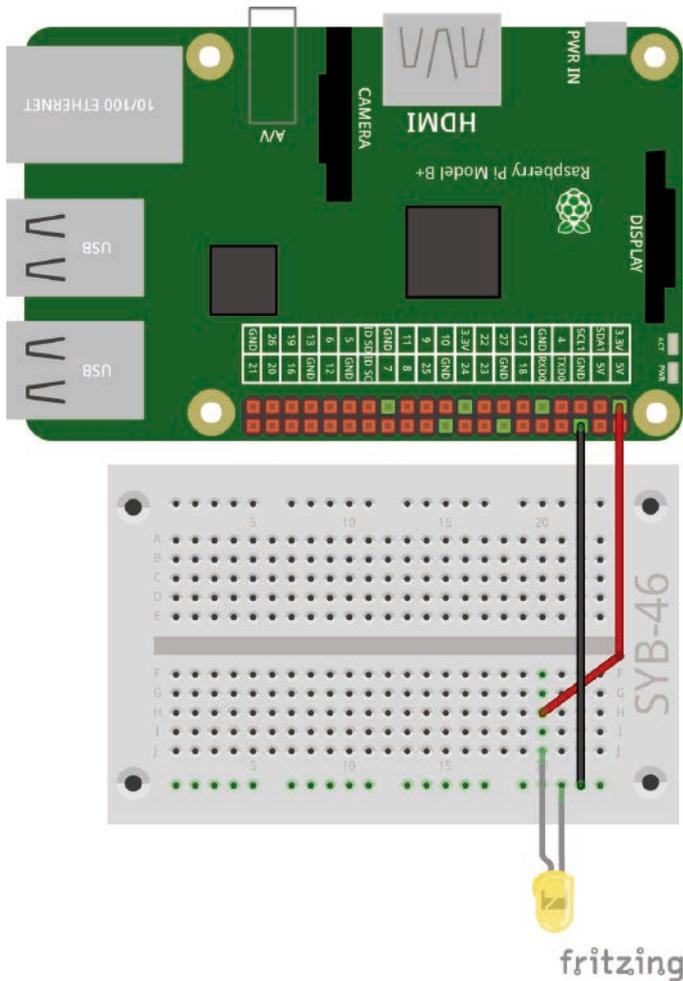
Mesures de précaution

En aucun cas, vous ne devez connecter les broches GPIO entre elles et attendre de voir ce qui se passe. Toutes les broches GPIO ne se laissent pas programmer librement. Certaines sont prévues pour l'alimentation électrique et d'autres buts.

Certaines broches GPIO sont directement connectées aux ports du processeur, un court-circuit peut détruire complètement le Raspberry Pi. Si vous raccordez deux broches ensemble via un interrupteur ou un LED, une résistance en série doit toujours être connectée entre les deux. Les LEDs avec une résistance intégrée sont une exception.

Pour les signaux logiques, utilisez toujours la broche 1 qui fournit + 3.3 V et qui peut supporter une charge allant jusqu'à 50 mA. La broche 6 est la conduite de masse pour les signaux logiques.

Les broches 2 et 4 fournissent 5 V à l'alimentation électrique du matériel externe. Vous trouverez ici autant de courant que l'alimentation USB du Raspberry Pi fournit. Cependant, ces broches doivent pas être connectées avec une entrée GPIO.



Le premier LED s'allume sur le Raspberry Pi.

Un LED s'allume

Aucun programme n'est requis pour la première expérience. Le Raspberry Pi est utilisé ici uniquement comme un bloc d'alimentation pour le LED RGB. L'expérience montre comment les LED RGB sont connectés. Assurez-vous que le LED RGB est correctement monté. Le côté plat se trouve à droite dans l'illustration.

La plupart des montages de circuits utilisent la barre de contact sur un côté de la platine enfichable comme contact de masse. Ici, les cathodes de toutes les LED sont insérées et connectées via un câble avec une broche GND sur le Raspberry Pi.

Composants : 1 platine enfichable SYB-46 ; 1 LED jaune avec résistance de série ; 2 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable)

Programmes à télécharger

Les programmes utilisés dans le calendrier de l'Avent sont disponibles ici pour téléchargement : <http://bit.ly/c-adventskalender-raspberry-pi-18>



Ouvrez la page Web directement avec le navigateur pré-installé sur le Raspberry Pi et téléchargez le fichier zip dans le répertoire d'origine / home / pi.



Démarrez le gestionnaire de fichier sur le Raspberry Pi. Celui-ci montre lors du démarrage le répertoire d'origine. Avec le bouton droit de la souris, cliquez sur le fichier zip téléchargé et choisissez le menu contextuel **Décompresser ici**.

L'archive de téléchargement pour le calendrier de l'Avent contient ce manuel au format PDF en couleur, afin que vous puissiez mieux voir les conduites individuelles sur les schémas de circuit.

2ème Jour

Aujourd'hui dans le calendrier de l'Avent

- 3 câbles de raccordement GPIO (long)

Le calendrier de l'Avent contient 3 câbles de raccordement GPIO de deux longueurs différentes. Les câbles courts servent la connexion entre le Raspberry Pi et la platine enfichable, les câbles longs raccordent les LEDs sur la crèche de Noël à la platine enfichable.

La première LED s'allume sur la crèche de Noël

Le LED est branché dans les trous indiqués sur la queue de l'étoile de Noël au-dessus de la crèche de Noël. Coupez-le le long des lignes marquées au dos du carton du calendrier de l'Avent.

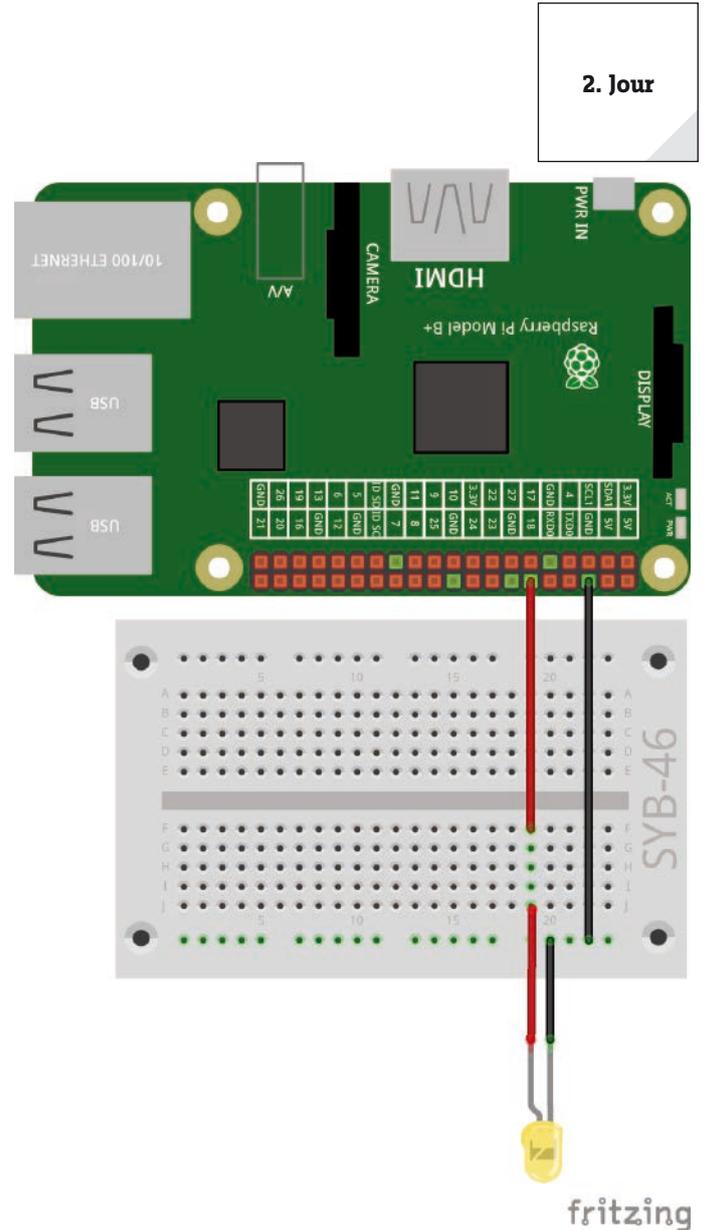
Les douilles des câbles de raccordement GPIO viennent de l'arrière sur les fils de connexion du LED. Branchez le LED aussi loin que possible dans les prises de câble. Vous devez néanmoins glisser, plier d'un angle faible les fils de connexion du LED quelques millimètres avant les extrémités.

Composants : 1 platine enfichable SYB-46 ; 1 LED jaune avec résistance en série ; 2 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 2 câbles de connexion GPIO longs (platine enfichable - LEDs)

Branchez le LED en haut sur l'étoile sur la crèche de Noël.

Cette fois, le LED ne s'allume pas en permanence, mais il s'allume par un programme dans le langage de programmation Scratch pendant une demi-seconde.

Scratch est préinstallé sur le Raspberry Pi dans le menu **Développement** et il est considéré comme l'un des langages de programmation les plus faciles à apprendre.



Le premier LED s'allume sur la crèche de Noël.

Le nouveau scratch 2

Depuis la première version de Raspbian, le langage de programmation Scratch est préinstallé dans la version 1.x. Pour les PC, il existe depuis quelques années la nouvelle version Scratch 2 avec beaucoup plus d'options. Il est notamment possible de créer ainsi certains blocs de fonctions propres. Scratch 2 fonctionne sur le PC en ligne dans votre navigateur. Toutefois il requiert plus de puissance de calcul que n'en peut proposer actuellement le Raspberry Pi. Depuis la version NOOBS 2.4.0, une version de scratch 2 est préinstallée dans le système d'exploitation Raspbian, qui s'exécute en mode hors connexion sans un navigateur et fonctionne facilement avec celui d'un Raspberry Pi 3. Le contrôle du matériel via l'interface GPIO est devenu beaucoup plus facile avec Scratch 2. Cependant, certaines fonctions importantes pour le contrôle GPIO ne sont pas encore supportées. C'est pourquoi nous continuons à utiliser le Scratch 1.4 qui a fait ses preuves pour tous les projets dans ce calendrier de l'Avent.

Réglages de base pour Scratch

Cliquez en haut à gauche à côté du logo Scratch sur le globe, puis choisissez **français**. La langue sélectionnée reste mémorisée, il n'est donc pas nécessaire de la sélectionner à nouveau chaque fois.

Scratch 1.4 fournit le support pour divers composants matériels au port GPIO qui doivent être activés une fois via l'élément de menu **Bearbeiten/Start GPIO server**.



Commuter Scratch 1.4 en allemand.

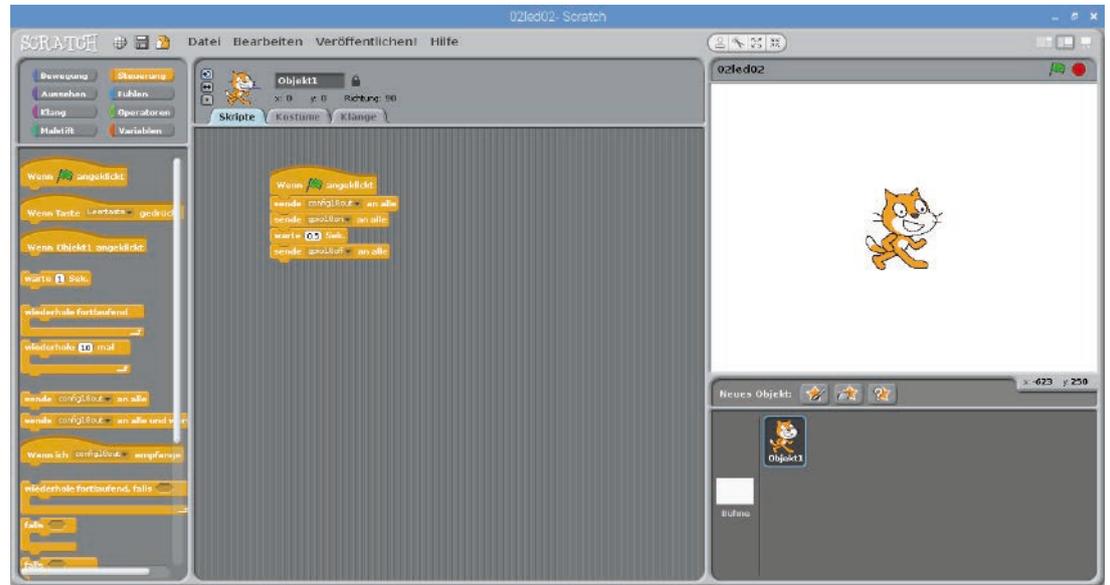


Démarrez le serveur GPIO Scratch 1.4.

Vous reconnaissez que le support GPIO est actif quand cet élément de menu change dans **Stop GPIO server**. Vérifiez cela dans chaque nouveau programme Scratch.

Le programme

Dans Scratch, il n'y a aucun besoin de taper un code de programmation. Les blocs sont accrochés les uns aux autres par simple glisser-déposer. La palette de blocs dans la partie gauche de la fenêtre Scratch contient les blocs disponibles ordonnés par thèmes.



Le programme Scratch 02Led02 fait s'allumer le LED pendant une demi-seconde.

Vous pouvez assembler le programme vous-même sur l'écran, ou bien utiliser le programme 02Led02 pour le calendrier de l'Avent par téléchargement. Pour cela, sélectionnez Menu **Données/ouvrir** et cliquez sur la boîte de dialogue suivante sur le bouton de commande **pi** pour sélectionner le répertoire personnel où se trouvent les programmes téléchargés.

Cliquez dans Scratch en haut à gauche sur le symbole jaune **Commande** pour afficher les blocs de contrôle dans la gamme de blocs à gauche. Pour ce premier programme, nous n'avons besoin que des blocs jaunes.

Faites glisser les blocs dont vous avez besoin, simplement hors de la gamme de blocs dans la fenêtre de script au milieu de Scratch.



Le bloc **si cliqué (drapeau vert)** est utilisé pour démarrer un programme. Les éléments de script suivants sont exécutés si vous cliquez sur le drapeau vert en haut à droite dans Scratch. Le bloc est rond en haut, ne passe donc sous aucun autre bloc. Il doit toujours être placé en premier.

Les commandes GPIO sont envoyées via le bloc Scratch gratter bloc **Envoyer... à tous**. Le nom respectif des broches et les mots clés pertinents sont entrés dans la zone de texte. Pour cela, cliquez dans le champ de texte dans le bloc, sélectionnez **Nouveau/éditer...** et saisissez le texte.

Au début la broche GPIO broche 18 est définie avec **config18out** comme sortie. Chaque broche GPIO peut être soit sortie soit entrée.

Dans la prochaine étape, le LED connecté à la broche 18 GPIO est allumé via un autre bloc Scratch **Envoyer... à tous** avec le texte **gpio18on**.

Ensuite le programme attend une demi-seconde. Pour cela, Scratch fournit son propre bloc **attendez... Sec.** an. Scratch utilise, comme beaucoup de programmes américains, le point comme séparateur décimal, et non pas la virgule selon l'habitude en Allemagne. Une demi-seconde est donc saisie 0.5 et non 0,5.

Enfin, le LED connecté à la broche 18 GPIO est allumé via un autre bloc Scratch **Envoyer... à tous** avec le texte **gpio18off**.

Le programme démarre quand vous cliquez dans la fenêtre Scratch sur le drapeau vert en haut à droite.

3ème Jour

Aujourd'hui dans le calendrier de l'Avent

- 1 LED rouge avec résistance en série
- 1 câble de raccordement GPIO (court)

Deux LEDs clignotent en alternance

L'expérience du 3ème jour fait s'allumer deux LEDs alternativement rouge et jaune. Le tout est contrôlé via une boucle infinie dans Scratch.

Composants : 1 platine enfichable ; 1 LED rouge avec résistance en série ; 3 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable)

Dans cette expérience, brancher les LEDs directement sur la platine enfichable avec la cathode sur le conducteur de la terre.

Le programme

Au début, les deux broches GPIO 18 et 24 et avec `config18out` et `config24out` sont définis comme sorties.

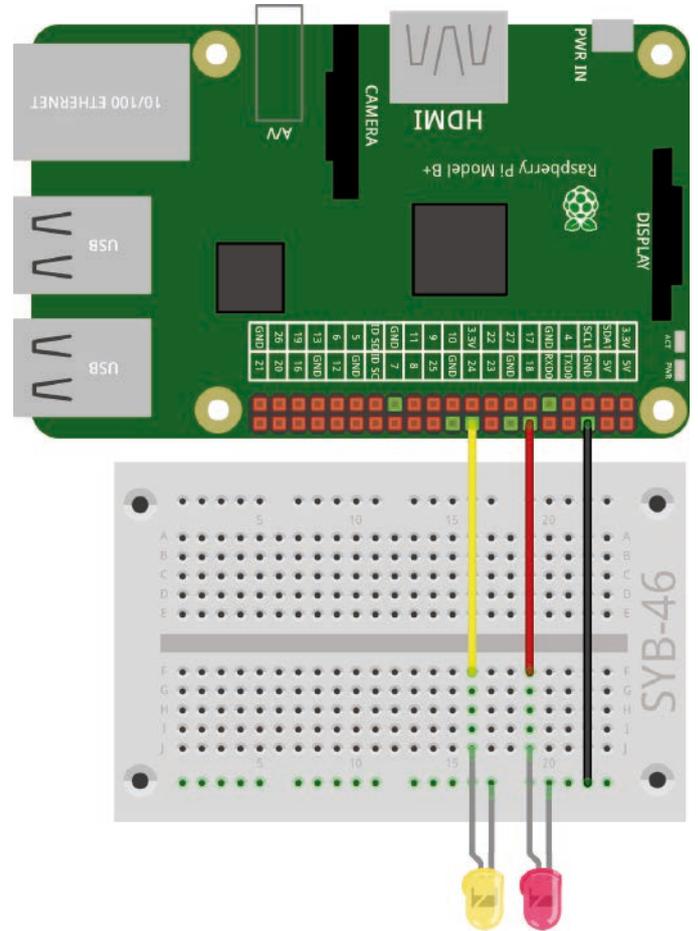


Une boucle **à répétition et sans interruption** veille à ce que les deux LEDs clignotent sans cesse, jusqu'à ce que l'utilisateur clique sur le symbole rouge en haut à droite dans Scratch.



Après que le LED rouge sur la broche 18 soit allumé et que le jaune sur la broche 24 soit éteint, le programme attend une demi-seconde. Ensuite, le LED jaune s'allume de la même manière et le rouge sur la broche 18 s'éteint. Après une demi-seconde supplémentaire le cycle recommence à nouveau.

3. Jour



fritzing

Deux LEDs clignotent sur le Raspberry Pi.



LE programme 03Led03 commande les deux LEDs.

4. Jour

4ème Jour

Aujourd'hui dans le calendrier de l'Avent

• 3 câbles de raccordement GPIO (long)

Une étoile allume deux LEDs en alternance

L'expérience du 4ème jour allume ou éteint deux LEDs. La commande utilise un objet dans Scratch pour le déplacer sur l'écran.

Composants : 1 platine enfichable SYB-46 ; 1 LED jaune avec résistance en série ; 1 LED rouge avec résistance en série ; 3 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 4 câbles de connexion GPIO longs (platine enfichable - LEDs)

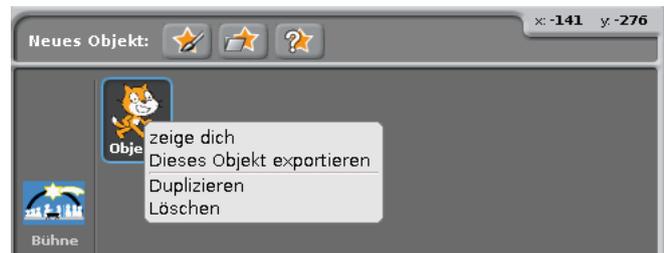
Le programme

Le programme utilise l'image de la crèche à l'arrière du calendrier de l'Avent comme fond d'écran.

Cliquez dans la fenêtre de l'objet (en bas à droite) sur la scène et allez dans la fenêtre de script sur l'**arrière-plan** de l'onglet.

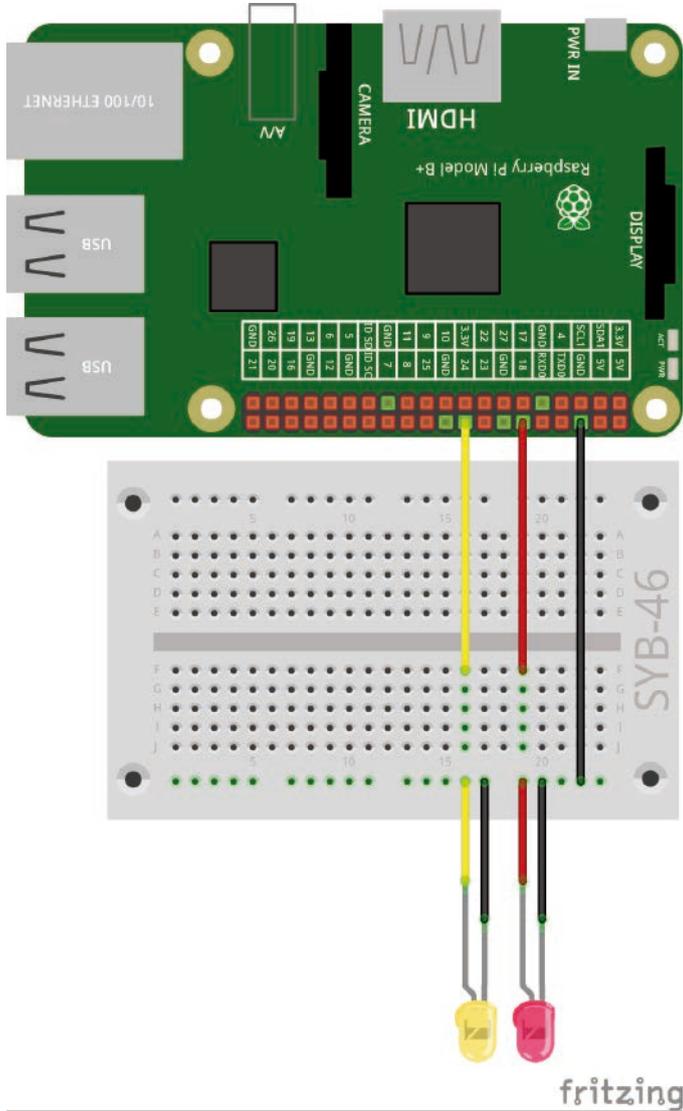
Cliquez sur **Importer** et importez l'image Crèche_bLeu.png par téléchargement. Ensuite, retirez le fond prédéterminé **Arrière-plan1** en un seul clic sur le symbole X.

Ensuite, supprimez le chat, qui n'est pas utilisé dans ce programme. Pour cela, cliquez avec le bouton droit de la souris sur le chat dans la fenêtre de l'objet, puis sélectionnez **Effacer** dans le menu de contexte.

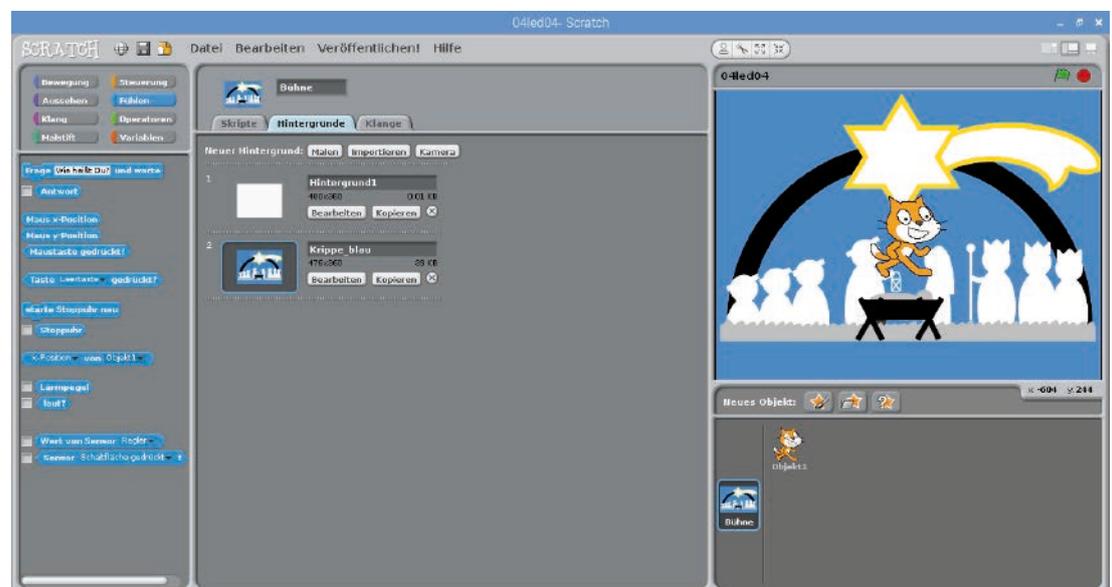


Supprimer le chat Scratch.

Dans le programme, un objet **étoile** doit commuter les LEDs. Scratch contient un programme de dessin simple, avec lequel vous pouvez peindre directement cet objet dans Scratch. Pour ce faire, cliquez le symbole **peindre nouvel objet** dans la fenêtre de l'objet.



Deux LEDs sont commutés via un programme Scratch.

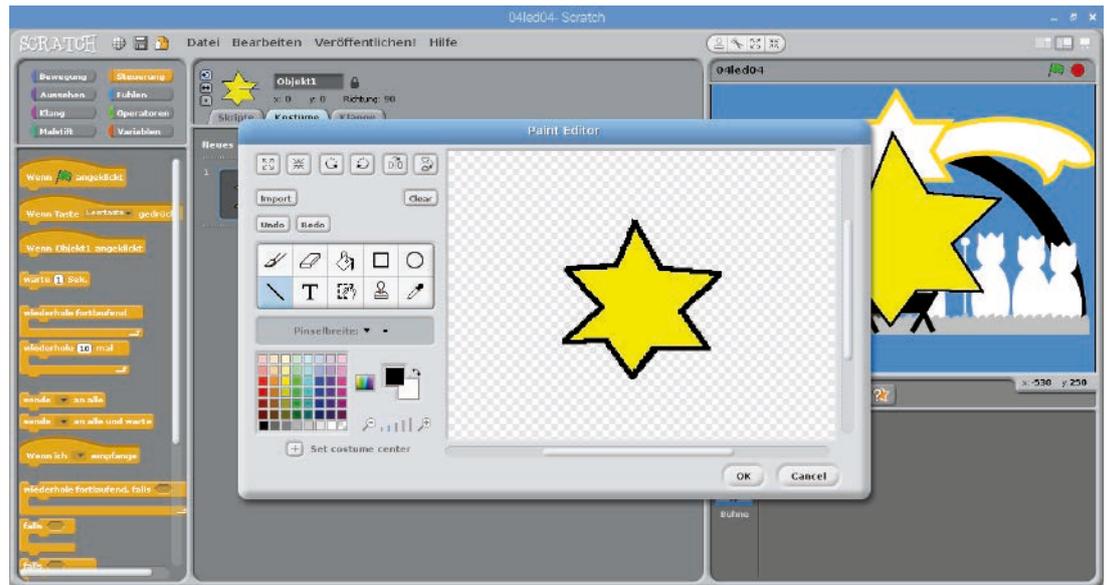


La nouvelle image de fond pour le programme.

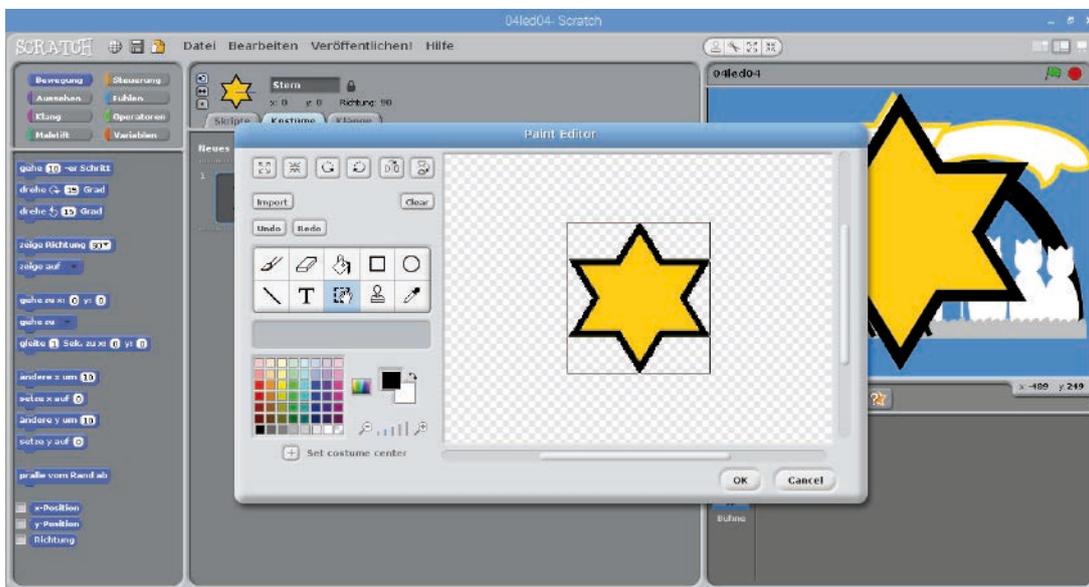
En alternative, vous pouvez télécharger via le symbole **charger nouvel objet des fichiers** le fichier `stern.png` comme nouvel objet.

Par défaut, l'étoile est beaucoup trop grande pour la scène. Commutez dans la fenêtre de script sur l'onglet **costumes** et cliquez sur **Modifier** pour ouvrir le programme de peinture.

Rétrécissez l'étoile via le symbole **Rétrécir** en haut à gauche sur la taille appropriée.

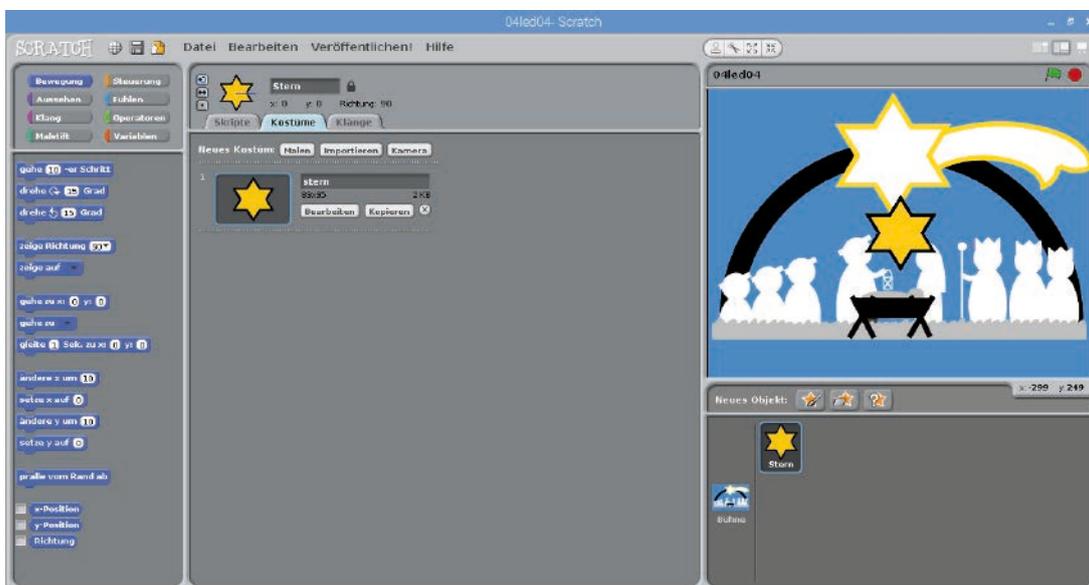


Le programme de peinture dans Scratch.



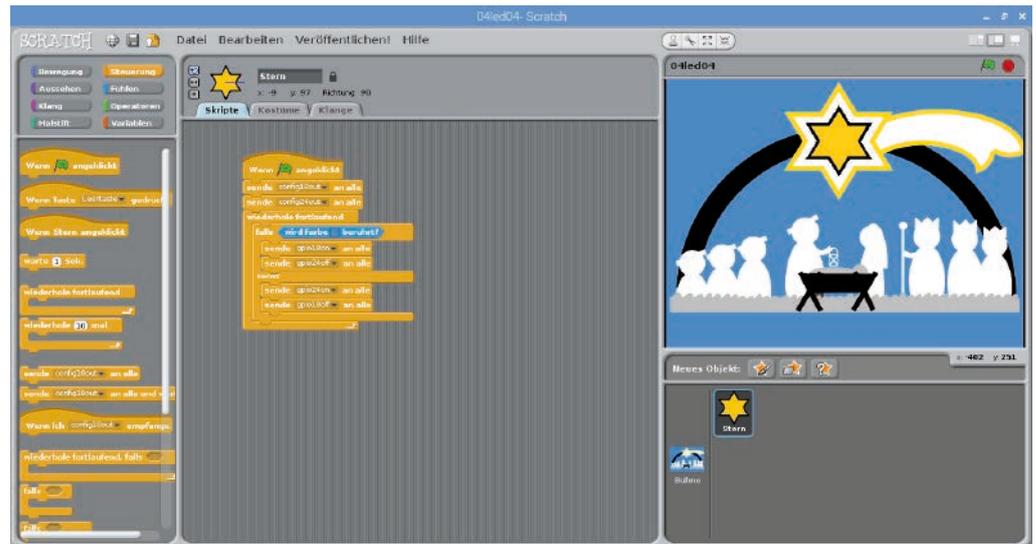
L'étoile importée dans le programme de peinture dans Scratch.

Renommez le nouvel objet dans le champ nom via la fenêtre de script en **Etoile**.



L'étoile réduite renommée.

Déplacez ensuite l'étoile sur la grande étoile de l'image d'arrière-plan.



Le programme 04Led04 commute deux LEDs avec une étoile.

Le programme 04Led04 démarre après que les broches GPIO aient été initialisées, une boucle infinie qui vérifie continuellement si l'étoile touche une zone bleue. Tant que ce n'est pas le cas, il se trouve sur l'étoile dans l'image de fond - la seule surface blanche qui soit assez grande.

Un bloc **Si ... sinon ...** allume un LED selon la position de l'étoile et éteint l'autre.

Tirez un bloc **touche ... une couleur** du bloc-palette **Sensation** dans le champ d'interrogation du bloc **Si ... sinon ...**

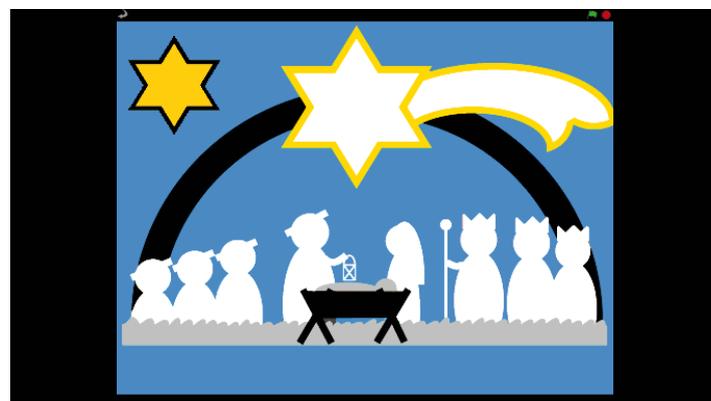
Appuyez sur le carré coloré dans ce bloc, une pipette apparaît. Cliquez avec celle-ci sur une surface bleue dans l'image de fond. Ainsi, cette couleur est sélectionnée lors de l'interrogation.

Ajoutez encore le bloc **Envoyer ... à tous** pour allumer les LEDs dans les deux parenthèses du bloc **Si ... sinon ...**, comme indiqué sur la figure.

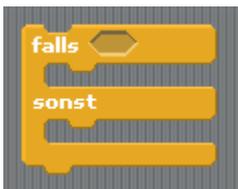
Démarrez le programme en cliquant sur le drapeau vert, le LED s'allume sur la broche 24.

Faites glisser l'étoile dans la direction de l'un des coins de l'écran, afin qu'elle touche le fond bleu. Le programme commute automatiquement sur les autres LEDs.

Le symbole de présentation en haut à droite dans le coin de la fenêtre Scratch commute en mode de présentation, dans lequel la scène Scratch apparaît en mode plein écran. Tous les éléments de fonctionnement disparaissent. Un clic sur la flèche en haut à gauche commute en retour.



Le programme 04Led04 dans le mode de présentation.



5ème Jour

Aujourd'hui dans le calendrier de l'Avent

- 1 bouton
- 1 câble de raccordement GPIO (court)

Commuter les LEDs avec le bouton

L'expérience du 5ème jour ne commute pas automatiquement les LEDs, mais uniquement lorsque l'utilisateur appuie sur un bouton.

Composants : 1 platine enfichable SYB-46 ; 1 LED jaune avec résistance en série ; 1 LED rouge avec résistance en série ; 4 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 1 câbles de connexion GPIO long (jusqu'à ce jour pas assez de câbles courts sont disponibles)

Le programme

Le programme fonctionne comme celui du 3è jour. Encore une fois, les deux LEDs s'allument et s'éteignent dans une boucle infinie. Contrairement au programme précédent, la commutation n'a pas lieu après un certain temps, mais seulement quand l'utilisateur a appuyé sur le bouton. Chaque pression de bouton commute les LEDs.



Le programme 05Led05commande les deux LEDs via un bouton.

Au début, une entrée est définie en plus des deux broches de sortie GPIO 21. Les entrées du Raspberry Pi traitent les signaux logiques numériques. Si une entrée avec + 3,3 V est connectée, elle reçoit le signal logique **élevé**, ce que Scratch évalue comme **1** ; Si l'entrée est connectée avec GND, elle reçoit le signal logique **Bas**, ce que Scratch évalue comme **0**.

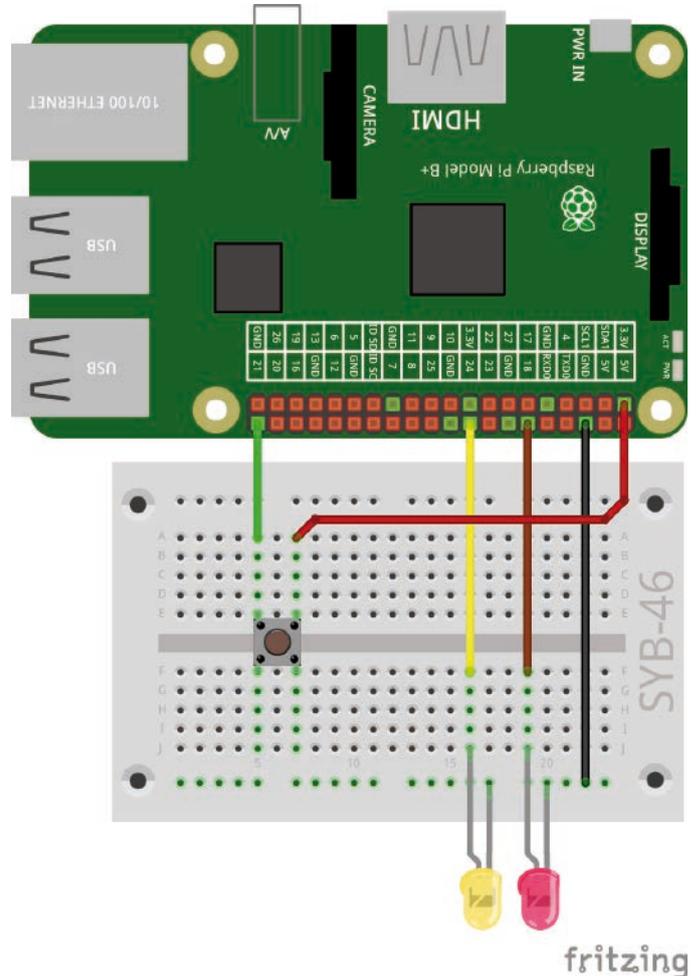
Attention

Ne jamais utiliser la broche + 5 V pin du Raspberry Pi pour les signaux logiques dans les circuits. 5V surchargerait les entrées GPIO et endommageraient le Raspberry Pi.

Une pression sur la touche relie dans notre circuit la broche GPIO 21 avec + 3,3 V. Si vous relâchez la touche, l'entrée reçoit un état indéfini, ce qui ne doit pas arriver dans l'électronique numérique. Dans de tels cas, toutes les broches GPIO disposent de soi-disant résistances déroulantes, qui tirent automatiquement vers le bas une entrée où aucun signal n'est présent.

Définir la broche GPIO pour le bouton avec **config21inpulldown** pour activer la résistance Pull-down intégrée à l'entrée.

5. Jour



Commuter les LEDs avec le bouton

Maintenant, cliquez sur la surface verte en haut à droite pour démarrer le programme encore inachevé. Ainsi toutes les broches GPIO sont définies. Seulement de cette façon, elles peuvent être choisies dans les listes. Le programme prend fin automatiquement.



Les broches GPIO sont initialisées.

Commence alors une boucle infinie, qui à chaque passage allume d'abord le LED sur la broche GPIO 18 et éteint celui de la broche 24.



Quand un LED est allumé, le programme n'attend plus un certain temps, mais avec un bloc **attend jusqu'à ...**, jusqu'à ce qu'un événement spécifique se produise, dans ce cas jusqu'à ce que la broche GPIO 21 prenne la valeur **1**, et qu'on a appuyé sur le bouton.

Pour l'interrogation elle-même, prévoir dans le bloc **attend jusqu'à ...** un champ allongé avec des extrémités pointues. Ici, il faut ajouter de la palette verte **Opérateurs**. Tirez le bloc avec le signe égal sur le champ avec l'espace réservé dans le bloc **attend jusqu'à ...**.



Cet opérateur est toujours juste si les deux valeurs à gauche et à droite du signe égal sont les mêmes.

Dans notre cas, la valeur de la broche GPIO 21 doit correspondre à la valeur **1**. Le chiffre 1 signifie **Élevé**. Donc, écrivez un **1** dans le champ de texte de droite dans le bloc vert.

Pour l'interrogation d'entrées GPIO, le bloc **Valeur de capteur ...** est utilisé dans la palette bleue **Sensation**. Dans le menu déroulant du bloc bleu choisissez le capteur **gpio21**. En plus de certains capteurs prédéfinis, toutes les broches GPIO qui sont définies comme entrée sont laissées au choix. C'est pourquoi le programme devait être lancé avant une fois brièvement.



Faites glisser alors le bloc bleu **Valeur de capteur ...** dans le champ de gauche du bloc vert dans le bloc **attend jusqu'à ...**.

Ensuite le programme attend avec **attend ...** Bloc **Sec.** pendant 0,2 seconde. Cela permet d'éviter que le bouton soit considéré immédiatement comme appuyé si le programme continue de s'exécuter. L'utilisateur dispose d'autant de temps pour relâcher le bouton.

Seulement ensuite les LEDs sont commutés et le programme attend à nouveau que l'utilisateur appuie sur le bouton.

Dupliquer les blocs

Lors de la construction d'un programme Scratch, vous ne devez pas créer à chaque fois des combinaisons de blocs similaires. Cliquez avec le bouton droit de la souris sur le premier bloc qui doit être dupliqué. Choisissez ensuite dans le menu **Dupliquer**. Tous les blocs inférieurs sont dupliqués automatiquement. Les blocs dupliqués peuvent être insérés à l'emplacement correspondant dans le programme.

6ème Jour

Aujourd'hui dans le calendrier de l'Avent

- 4 câbles de raccordement GPIO (long)

Commuter les LEDs avec pupitre de commande Scratch

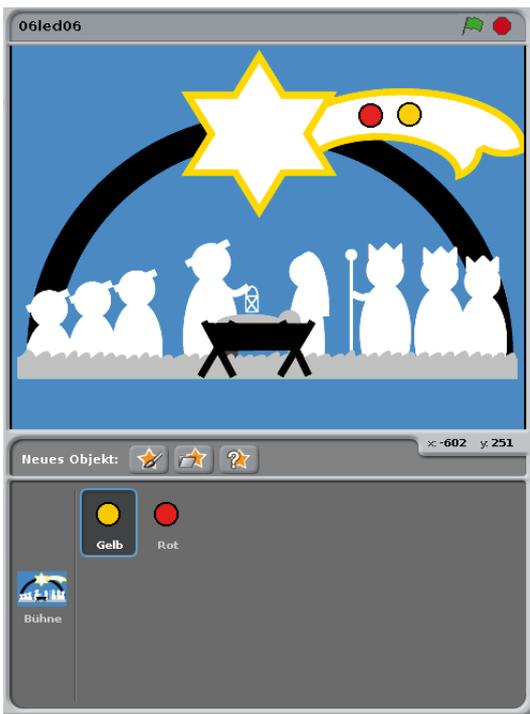
Dans le programme du 6ème jour, deux LEDs sont allumés et éteints via des clics dans Scratch. A cet égard, le pupitre de commande Scratch affiche l'état des LEDs.

Composants : 1 platine enfichable SYB-46 ; 1 LED jaune avec résistance en série ; 1 LED rouge avec résistance en série ; 3 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 4 câbles de connexion GPIO longs (platine enfichable - LEDs)

Le programme

Le programme 06led06 utilise la même image d'arrière-plan que les programmes des jours précédents.

Pour les deux LEDs sur la queue de l'étoile on créera deux objets circulaires avec les noms **jaune** et **rouge**. Supprimez le chat et vous dessinez l'un des objets. Vous pouvez également importer le fichier `kreis_gelb.png` par téléchargement. Pour créer le deuxième objet, dupliquez le premier plus tard avec tous les réglages et les blocs de script et ne personnalisez-le que légèrement.



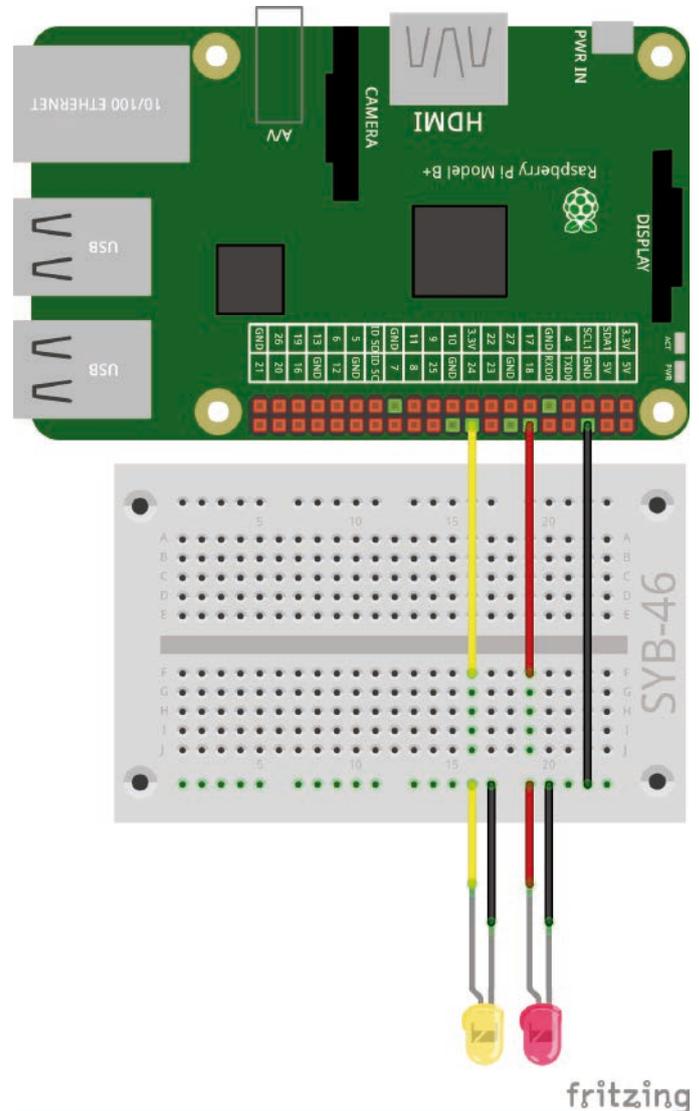
Les deux objets pour les LEDs.

Chaque objet dans Scratch peut changer son apparence via les soi-disant costumes. Sélectionnez le LED objet de LED et allez dans la fenêtre de script sous l'onglet **Costumes**.

Copiez le costume existant en cliquant sur **copier**. Renommez les deux costumes : **allumé** pour le LED allumé et **éteint** pour celui éteint.

Modifiez le costume **éteint** avec le programme de peinture. Remplissez la zone de couleur avec du gris pour représenter le LED éteint.

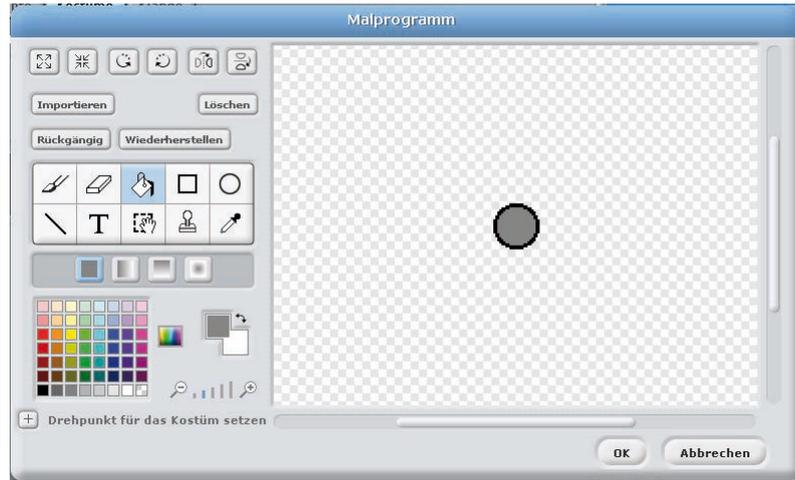
6. Jour



Trois LEDs sur la crèche de Noël.



Les deux costumes de l'objet jaune.



Costume pour un LED éteint dans le programme de peinture.



Le programme se compose de plusieurs parties - pour la scène et les objets. Tout d'abord, choisissez la scène dans la fenêtre de l'objet. Le bloc de script de la scène initialise les broches GPIO en cliquant sur le drapeau vert.

L'objet **jaune** obtient ses propres blocs de script. Lorsqu'on clique sur le drapeau vert, l'objet doit être éteint, donc apparaître en gris. Pour cela, accrochez au bloc **quand vert (drapeau vert) cliqué** un bloc **Enfile un costume...** de la gamme de bloc **Apparence**. Sélectionnez dans le champ de sélection de ce bloc le costume **éteint**.

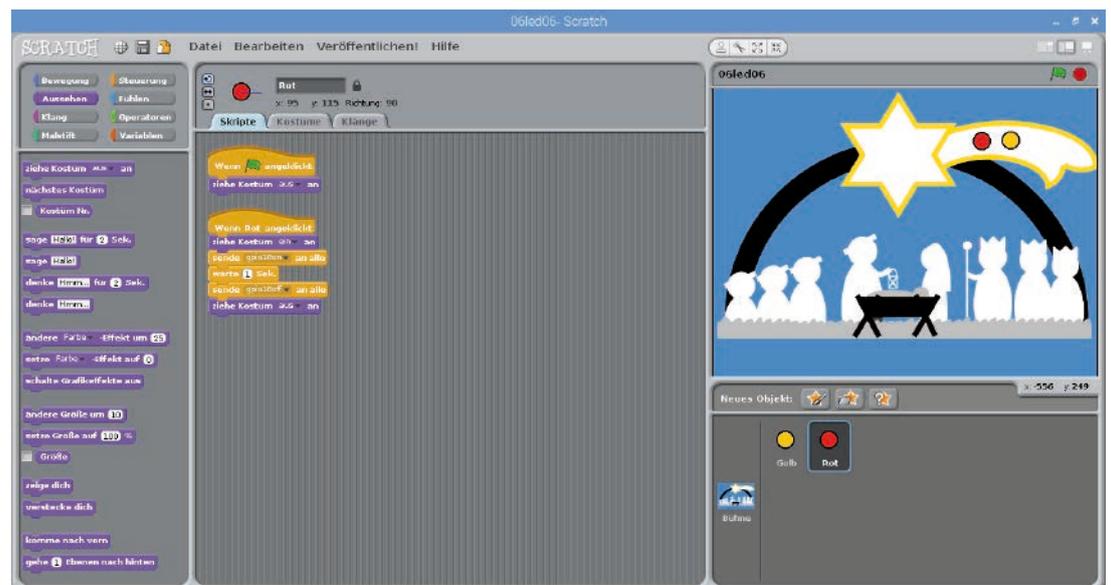


Un clic sur l'objet lui-même est cliqué, l'écran devrait s'allumer en couleur pendant une seconde. Dans le même temps, le LED correspondante doit s'allumer.

Pour cela utilisez le bloc **si jaune cliqué** dans l'onglet **Commande**.

Dans ce cas, le costume est d'abord commuté sur **allumé**, puis le LED s'allume sur la broche GPIO 24. Après une période d'attente d'une seconde, ce LED est à nouveau éteint et le costume revient sur **éteint**.

Tous les blocs de script sont terminés, vous dupliquez l'objet **jaune** avec un clic droit dans la fenêtre objet. Renommer le nouvel objet en **rouge**.



L'objet dupliqué pour le LED rouge.

Modifiez le costume **allumé** avec le programme de peinture. Remplissez ici le cercle avec de la couleur rouge. Puis changez encore la broche GPIO sur 18 utilisée dans le bloc de script.

Démarrez le programme en cliquant sur le drapeau vert. Les deux LEDs sont éteints, les objets apparaissent en gris. Cliquez sur l'un des objets, il apparaît en couleur, et la LED correspondante s'allume pendant une seconde.

7ème Jour

Aujourd'hui dans le calendrier de l'Avent

• 1 LED rouge avec résistance en série

7. Jour

Chenillard avec 3 LEDs

Dans le programme du 7ème jour permet d'allumer alternativement trois LED. L'effet chenillard apparait quand un LED s'allume pour une courte période et une fois qu'il est éteint, le LED à côté de lui s'allume immédiatement. Après que le dernier LED de la série soit éteint, la première s'enclenche à nouveau.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED rouge avec résistance en série ; 4 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 6 câbles de connexion GPIO longs (platine enfichable - LEDs)

Le programme

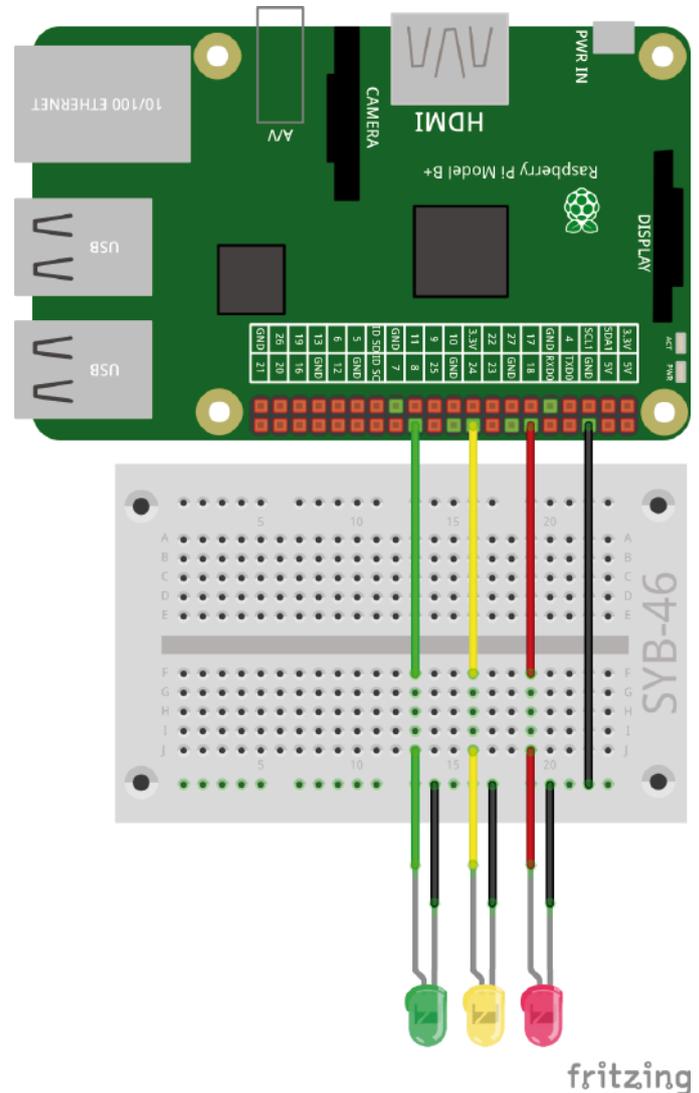
Le programme 071ed07 permet à trois LEDs de s'allumer comme un chenillard. La vitesse peut être réglée interactivement tandis que le programme est en cours d'exécution. Pour cela, le programme utilise une variable appelée **pause**, qui spécifie la pause entre deux actions de commutation.



Le programme 071ed07 permet à trois LEDs de s'allumer comme un chenillard.

Variables dans Scratch

Les variables ne sont pas de petits supports de données dans lequel programme retient un nombre ou toute autre chose. Si le programme se termine, cette mémoire variable être vidée automatiquement. Les variables sur Scratch doivent être appliquées sur le bloc de palette **Variables** d'abord avec un clic sur **nouvelle variable**, avant de pouvoir être utilisés. Ensuite, vous pouvez faire glisser de la variable nouvellement créée depuis le bloc palette dans un champ prévu à cet effet bloc dans le programme. En outre, divers blocs pour la lecture et la modification des variables sont disponibles sur le bloc palette.



Chenillard avec 3 LEDs



Le programme s'exécute entièrement sur la scène, il ne contient aucun objet. Après le démarrage, les trois broches GPIO, 18, 24 et 8 sont définies en premier comme sorties pour les LEDs et toutes sont éteintes.

Un bloc **Faire pause sur 5** du bloc palette **Variables** règle la pause sur 5, ce qui correspond à une demi-seconde dans le programme.

Puis commence une boucle infinie, éteint les LEDs l'un après l'autre. Comme dans Scratch, seuls les nombres entiers sont paramétrables de manière interactive comme valeurs de variables, le programme divise la valeur de la variable **pause** par 10. Pour cela, utilisez le bloc **.../...** du bloc palette **Opérateurs**. Faites glisser le symbole de la variable **pause** du bloc palette **Variables** dans le champ de l'opérateur et inscrivez 10 à droite.

Pour pouvoir modifier interactivement la valeur de la variable dans le programme, cochez à gauche à côté du symbole de variable sur le bloc palette **Variables**. Alors apparaît la variable sur la scène et montre toujours la valeur actuelle.

Cliquez avec le bouton droit de la souris sur la variable sur la scène et choisissez dans le menu contextuel **Variateur**. S'affiche alors un variateur qui permet de régler la valeur de variable.

Cliquez alors aussi sur définir **Définir la plage de variation** et fixez la valeur maximale sur **10**, ce qui équivaut à une pause d'une seconde pause dans ultérieures divisions par 10.



Démarrez le programme. Les LEDs clignotent en séquence automatiquement. Vous pouvez maintenant régler la vitesse à l'aide du curseur pour les variables **pause** sur la scène. Les variables sur la scène sont aussi affichées en mode plein écran.



8ème Jour

Aujourd'hui dans le calendrier de l'Avent

• Câble de connexion

8. Jour

Câble de connexion

Aujourd'hui, le fil de branchement est inclus dans le calendrier de l'Avent. Vous pouvez ainsi établir de brèves liaisons avec lesquelles les rangées de contacts sur la platine enfichable peuvent être connectées. Selon l'expérience, coupez le fil à la longueur appropriée avec une petite pince coupante. Afin de pouvoir mieux enficher les fils dans la platine, il est conseillé de les couper de manière légèrement transversale de sorte qu'une espèce d'angle se forme.

Variation des LEDs

Les LEDs peuvent avoir deux états différents : allumé et éteint. Il en va de même pour les sorties numériques des ports GPIO définis. Par conséquent, il serait théoriquement impossible de varier l'intensité d'un LED.



Liens : Rapport cyclique 50 % - à droite : Rapport cyclique 20 %.

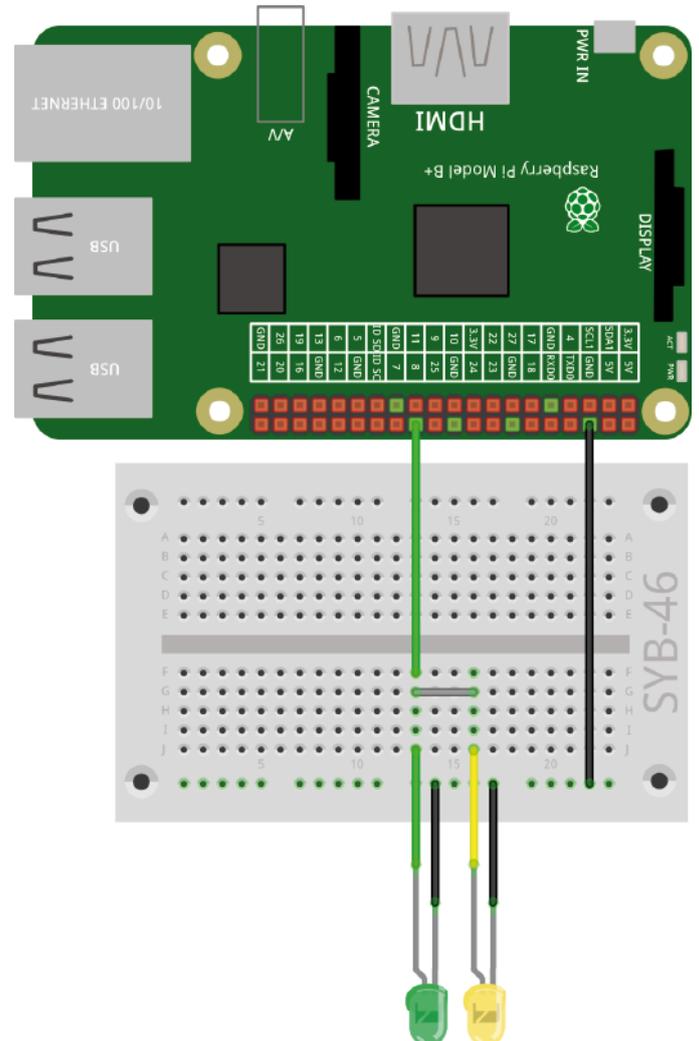
Cependant, une astuce permet de régler la luminosité d'un LED sur un port numérique GPIO. Si vous laissez un LED clignoter rapidement, l'œil humain ne perçoit plus le clignotement. La technique appelée modulation de largeur d'impulsion (PWM) produit un signal pulsé qui s'allume et s'éteint à intervalles très courts. La tension du signal reste toujours la même, mais le rapport entre le niveau **Faux** (0 V) et le niveau **Vrai** (+ 3,3 V) est modifié. Le rapport cyclique donne le ratio de la longueur d'un l'état allumé avec la durée totale d'un cycle de commutation. Plus le rapport cyclique est petit, plus le temps d'allumage d'un LED dans un cycle de commutation est court. En conséquence, le LED semble plus sombre qu'un LED allumé en permanence.

Pourquoi la fréquence idéale de PWM est 50 Hertz

L'œil humain ne perçoit pas le changement de lumière plus rapide que 20 hertz. Étant donné que le réseau de courant alternatif en Europe utilise une fréquence de 50 Hertz, de nombreux corps lumineux clignotent avec cette fréquence, ce qui n'est pas perçu par l'œil humain. Si un LED Clignotait avec plus de 20 Hertz, mais moins de 50 Hertz, il y aurait des interférences avec les autres sources de lumière, de sorte que l'effet de variation ne semblerait plus uniforme.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 2 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 4 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 1 cavalier

L'effet PWM fonctionne avec une intensité différents avec les LEDs de couleurs diverses. Dans le circuit, les deux LEDs sont connectés sur la même broche GPIO pour montrer clairement cet effet. Pour cela, deux rangées de la platine enfichable sont reliées avec un cavalier court en fil de connexion.



Le programme

Le programme 08Led08 fait varier les deux LEDs à l'aide d'une valeur réglable interactivement.



Le programme 08Led08 fait varier les deux LEDs.

Tout d'abord, le programme définit la broche GPIO 8 comme sortie avec fonction PWM. Avec une broche définie de telle sorte, Scratch propose des fonctions spéciales pour la commande d'un signal PWM.

Après cela, la variable définie préalablement **pwm** est réglée sur la valeur **0**. Dans cet état, la broche PWM est complètement éteinte. Ensuite, une boucle infinie interroge la variable **pwm** réglable interactivement et définit la broche GPIO 8 sur la nouvelle valeur.

Réglez la plage de variation sur **Min.:0** et **Max:500**. Dans cette plage, Scratch attend les valeurs pour PWM.



Pour envoyer la valeur paramétrée à la broche via **Envoyer... à tous**, nous utilisons le bloc **Raccorder** qui relie deux textes arbitraires entre eux de sorte qu'ils sont accrochés consécutivement. Les numéros sont traités comme du texte, ils ne sont donc pas additionnés, mais aussi simplement placés consécutivement.

Écrivez dans le premier champ du bloc la séquence de signes `gpio8pwm` et faites glisser la variable **pwm** dans le deuxième champ. De cette façon, vous réglez une broche PWM sur une valeur spécifique. Puis faites glisser le bloc **Raccorder ...** dans le champ de texte du bloc **Envoyer... à tous**.

Les LEDs s'allument avec la luminosité réglée, puis la boucle infinie redémarre et interroge la nouvelle valeur de la variable **pwm**.

9ème Jour

9. Jour

Aujourd'hui dans le calendrier de l'Avent

- 4 câbles de raccordement GPIO (courts)

Chenillard avec bouton

Dans le programme du 9ème jour contient à nouveau un chenillard mais cette fois-ci avec des capacités spéciales. Les LEDs s'allument uniquement lorsque vous appuyez sur un bouton et la luminosité de chaque LED peut être réglée individuellement via le PWM.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED rouge avec résistance en série ; 1 bouton ; 6 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 6 câbles de connexion GPIO longs (platine enfichable - LEDs)

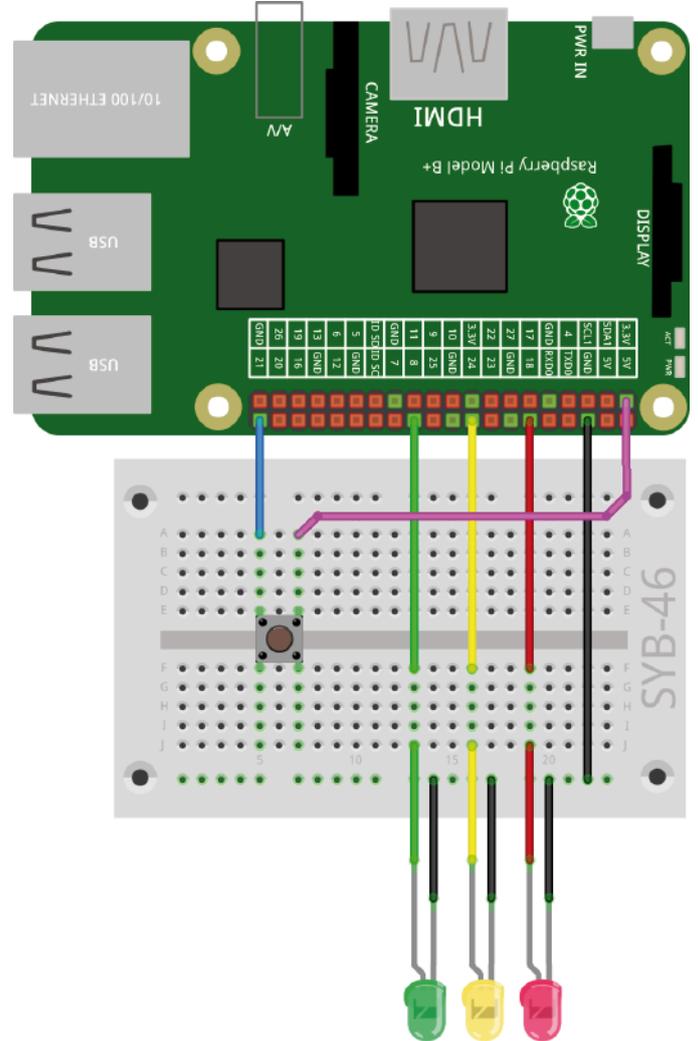
Le programme

Le programme 09Led09 initialise au début les broches GPIO, 8, 24 et 18 comme sorties PWM et la broche 21 comme entrée avec résistance pull-down.

Une boucle infinie interroge constamment si le signal 1 est présent sur la broche 21, c'est-à-dire si le bouton est appuyé.

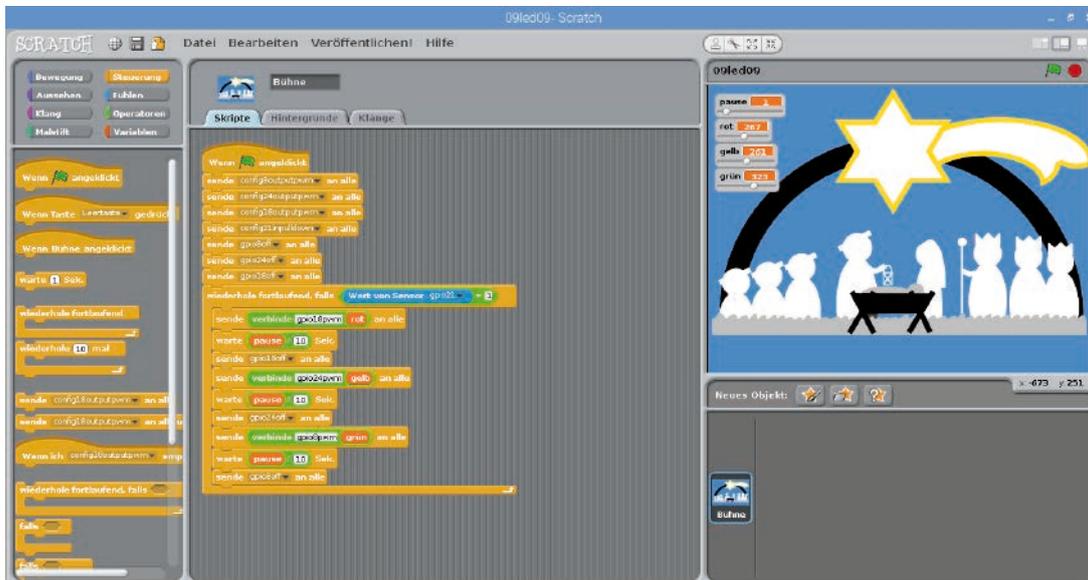
Les valeurs PWM des trois LEDs sont sauvegardées dans les variables **rouge**, **jaune** et **vert** qui peuvent toutes être réglées via un variateur dans la plage **0...500**. La pause est à nouveau réglable dans la plage **0...10** et elle est calculée en dixièmes de seconde comme pour le jour 7.

Démarrez le programme en cliquant sur le drapeau vert. Définissez les valeurs PWM et maintenez le bouton enfoncé pour voir le chenillard. Les valeurs peuvent être modifiées même si le bouton n'est pas enfoncé parce que le programme continue également de s'exécuter. Dès que vous appuyez de nouveau sur le bouton, le chenillard démarre avec les paramètres PWM actuels.



fritzing

Trois LED et un bouton.

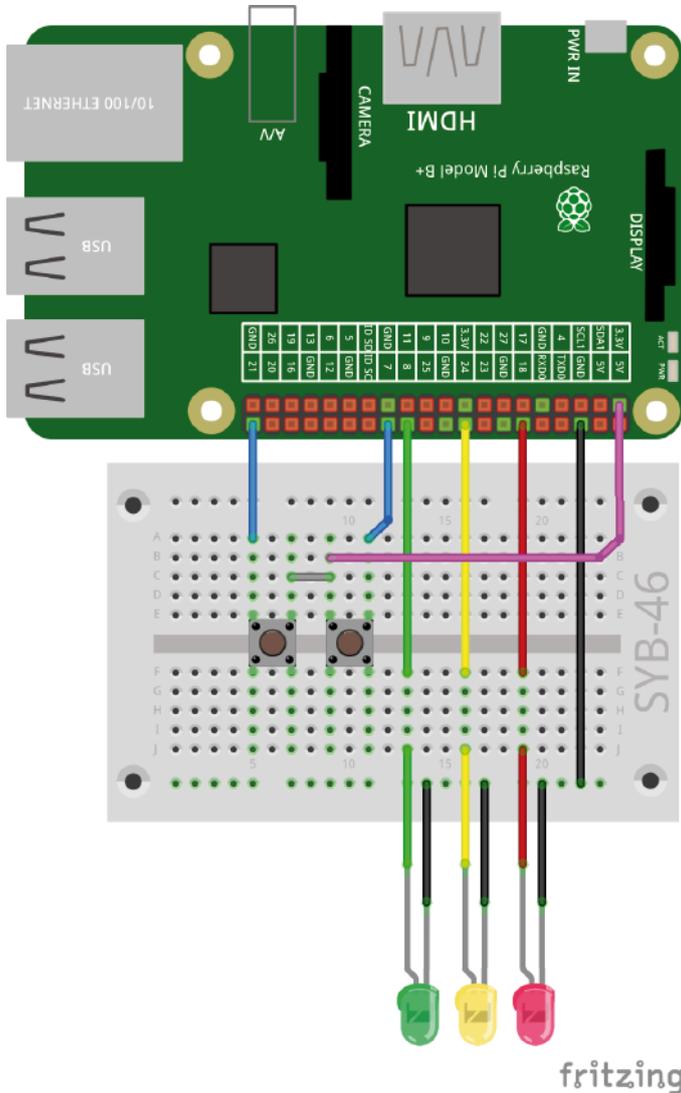


Les programmes 09Led09 commande un chenillard avec les LEDs PWM.

10ème Jour

Aujourd'hui dans le calendrier de l'Avent

- 1 bouton
- 1 câble de raccordement GPIO (court)



Trois LEDs et deux boutons.

Commuter des LEDs avec deux boutons

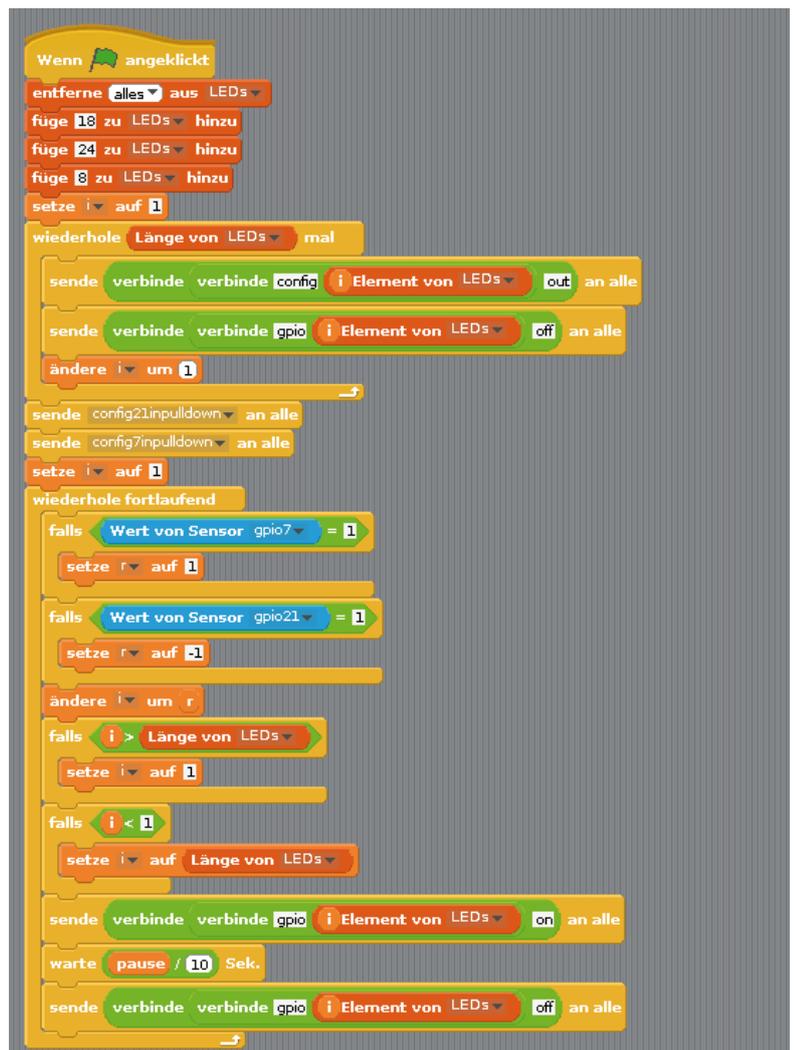
La direction du chenillard peut être sélectionnée par l'intermédiaire de deux boutons.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED rouge avec résistance en série ; 2 boutons ; 7 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 6 câbles de connexion GPIO longs (platine enfichable - LEDs) ; un cavalier

Le cavalier est utilisé pour connecter deux rangées des contacts de la platine enfichable via laquelle les deux boutons sur la broche + 3, 3 V sont connectés au Raspberry Pi.

Le programme

Le programme `10led10` permet d'allumer trois LEDs comme un chenillard pour lequel la direction est commuté via deux boutons. Pour ce faire, le programme utilise une liste plutôt que des variables individuelles pour les trois broches GPIO.



Le programme `10led10` permet à trois LEDs de s'allumer comme un chenillard.

Créer une liste avec le nom **LEDs** pour ce programme. Pour ce faire, cliquez sur le bloc palette **Variables** sur la **nouvelle liste**. Sur ces listes, plusieurs valeurs peuvent être stockées et adressées via la position dans la liste.

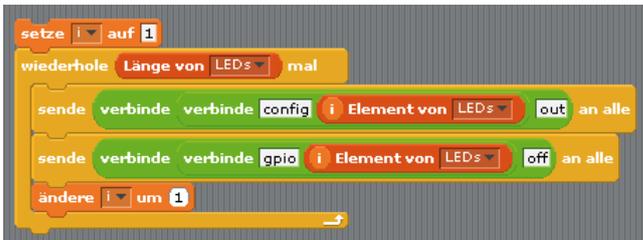
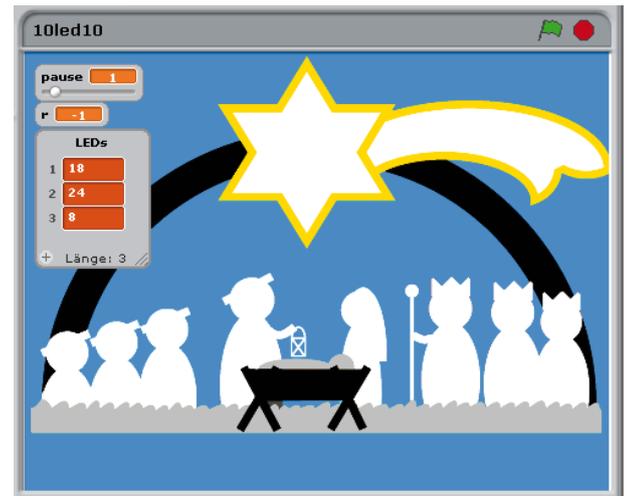
En outre, trois variables simples sont nécessaires :

- **i** - un compteur simple qui est utilisé à divers points dans le programme.
- **r** - spécifie la direction du chenillard.
- **pause** - spécifie, dans les programmes précédents, la durée d'allumage d'un LED.

La variable **pause** est paramétrable de manière interactive sur la scène Scratch, **r** s'affiche, mais ne peut pas être réglée. Le compteur **i** n'est pas intéressant pour l'utilisateur et n'est pas affiché. La liste peut être vue sur la scène. Mais comme les valeurs de changent jamais, vous pouvez également les désactiver.

La première étape supprime tous les éléments de la liste, s'il y a encore des valeurs d'une exécution de programme précédente. Ensuite, les numéros des broches GPIO utilisées pour les trois LEDs sont enregistrés dans les trois premières positions de liste.

Une boucle est parcourue trois fois et elle adresse les trois broches GPIO pour les LEDs et les éteint. L'utilisation du bloc **Longueur de ...** au lieu du chiffre 3, le programme est très facile à adapter pour l'allongement des listes.



La variable **i** vaut pour un nombre compris dans la plage **1...3**. Ainsi le bloc **Element i de LEDs** spécifie le nombre des broches GPIO respectiv. Des combinaisons de blocs **Raccorder ...** permettent d'initialiser les broches GPIO, par exemple **config18out** et **gpio18off**.

Après que les deux broches GPIO aient été initialisées pour les boutons comme entrées, une boucle infinie démarre, qui interroge tout d'abord les deux boutons. Selon le bouton qui est enfoncé, la direction du chenillard est réglée dans la variable **r** sur **1** ou **-1**.

Selon la direction, le compteur **i** qui désigne le LED successif qui doit s'allumer augmente ou diminue.

Ce compteur ne doit pas fonctionner au-delà des valeurs limites énoncées de 3 et 1. Si la variable **i** après l'augmentation est supérieure à la longueur de la liste, elle est réinitialisée à 1. Après le dernier LED; le premier s'allume à nouveau. Selon le même schéma, **i** est réglé sur le numéro di dernier LED, s'il est passé en dessous de 1.

Une fois que le programme a défini quel LED doit s'allumer, celui-ci est allumé selon une combinaison de blocs **raccorder ...** et à nouveau éteint après une pause. Puis la boucle recommence depuis le départ.



11. Jour

11ème Jour

Aujourd'hui dans le calendrier de l'Avent
 - 1 LED rouge avec résistance en série



Broches de connexion d'un LED RVB.

LEDs RVB

Un LED normal s'allume seulement dans une seule couleur. Les LEDs RVB utilisés dans le calendrier de l'Avent peuvent briller au choix de plusieurs couleurs. Pour les LEDs RVB, trois LEDs de couleurs différentes sont en principe montés dans un boîtier transparent. Chacun de ces trois LEDs a une anode propre, via laquelle ils sont liés à une broche GPIO. La cathode qui est reliée à la ligne de masse n'est présente qu'une seule fois. Par conséquent, un LED RVB a quatre fils de connexion.

Les fils de raccordement du LED RVB sont de longueur variable pour les rendre clairement identifiables. Contrairement aux LED normaux, la cathode a ici le fil le plus long.

Les LEDs RVB fonctionnent comme trois LEDs individuels et nécessitent donc trois résistances. Celles-ci sont intégrées dans les LEDs rouges contenus dans le calendrier de l'Avent.

Jeux de couleur RVB

L'expérience du 11ème jour permet d'allumer un LED rouge automatiquement dans différentes couleurs successives.

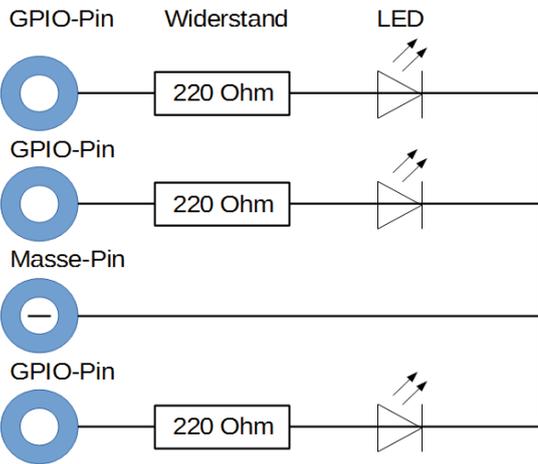
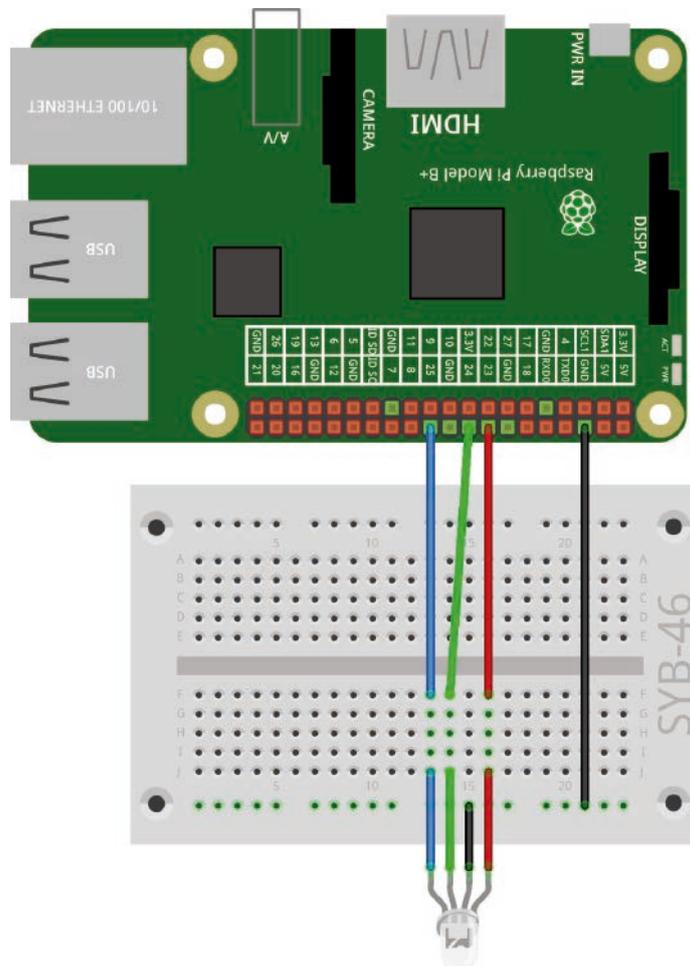


Schéma de circuit interne d'un LED RVB avec trois résistances.

Composants : 1 platine enfichable SYB-46 ; 1 LED RVB avec résistance en série ; 4 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 4 câbles de connexion GPIO longs (platine enfichable - LEDs)



fritzing

LED RVB aux broches GPIO 23, 24, 25.

Le programme

Le programme 11rgbLed01 allume puis éteint successivement différents composants de couleur parmi les trois du LED RVB. Comme parfois deux couleurs sont allumées, le LED RVB s'allume alternativement dans trois couleurs de base et dans trois couleurs mélangées.

Ici aussi variable **pause** qui commande la durée entre l'allumage des couleurs individuelles et ainsi la vitesse de clignotement du LED RVB, s'affiche sur la scène Scratch avec un contrôleur.

Après que les trois sorties GPIO utilisées aient été initialisées, différentes broches sont allumées et éteintes tour à tour. L'attente entre les processus de commutation peut être modifiée interactivement par l'utilisateur via le variateur sur la scène Scratch.

L'utilisation de numéros GPIO dans le programme n'est pas très claire et aussi sujettes à l'erreur, lorsqu'on transforme le programme. Une deuxième version de programme 11rgbLed02 fonctionne exactement de la même, mais utilise des noms de variables pour les broches GPIO. En conséquence, le programme est certes plus long mais plus clair.



Le programme 11rgbLed02 utilise des variables pour les numéros de broches GPIO.



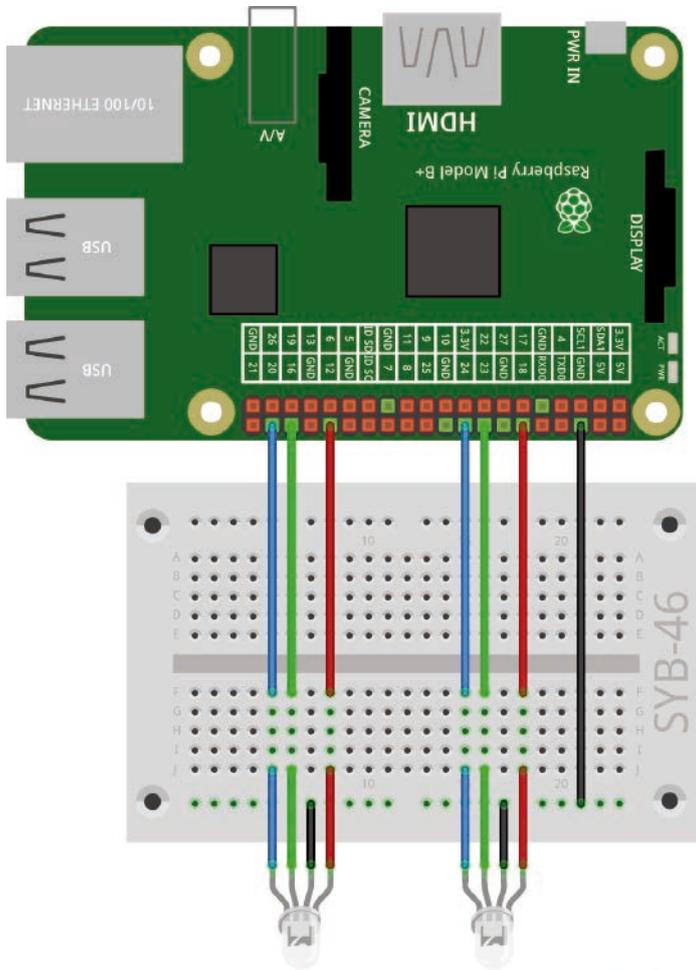
Le programme 11rgbLed01 commande un LED RVB.

12. Jour

12ème Jour

Aujourd'hui dans le calendrier de l'Avent

• 1 LED rouge avec résistance en série



fritzing

Deux LEDs RVB sur la crèche de Noël.

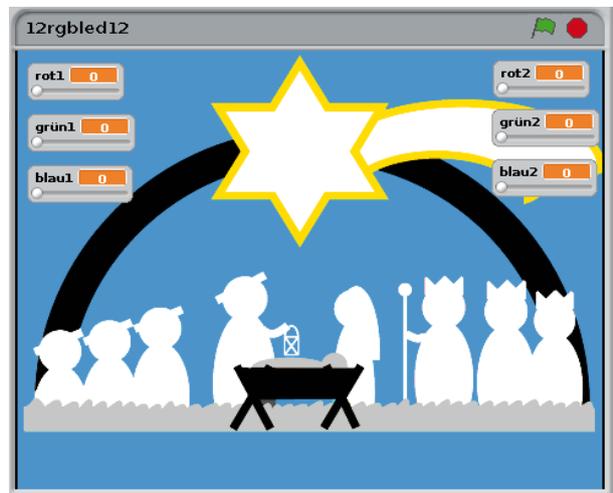
Mélange de couleur avec LEDs rouge

L'expérience du 12ème jour mélange diverses couleurs sur deux LEDs RVB via PWM.

Composants : 1 platine enfichable SYB-46 ; 2 LED RVB avec résistances en série ; 7 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 8 câbles de connexion GPIO longs (platine enfichable - LEDs)

Le programme

Il est possible de varier chaque composant de couleur d'un LED RVB via PWM. Par conséquent, tous les mélanges de couleurs possibles. Le programme 12rgbled12 comprend six variables qui peuvent être réglées via le variateur et les trois composants de couleur commandent des deux LEDs RVB indépendamment.



Six variables réglables mélangent les couleurs sur les LEDs RVB.

En cliquant sur le drapeau vert, les numéros de broches sont enregistrés pour les deux LEDs rouges dans les variables **r1**, **g1**, **b1** et **r2**, **g2**, **b2** et ces broches sont ensuite mises en place comme sorties PWM.

Après l'initialisation, une boucle infinie interroge constamment les valeurs réglées des variables **rouge1**, **vert1**, **bleu1** et **rouge2**, **vert2**, **bleu2** et définit les signaux PWM pour les LEDs RVB respectifs.



Les programmes 12rgbled12 mélange des couleurs vives sur deux LEDs RVB via PWM.

13ème Jour

13. Jour

Aujourd'hui dans le calendrier de l'Avent

- 1 x pâte à modeler

- 1 résistance 20 MOhm (rouge noir bleu)

Résistance

Les résistances sont utilisées pour limiter le courant des composants électroniques sensibles, mais sont aussi utilisées comme résistances pour LEDs. L'unité de mesure de résistance est l'ohm. 1.000 ohm correspondent à un kilohm, en abrégé kOhm. 1.000 kOhm correspondent à un mégohm, en abrégé MOhm. Pour l'unité Ohm ohms on utilise aussi souvent le symbole oméga Ω .

Les bagues de couleurs sur les résistances fournissent la valeur de la résistance. Avec un peu de pratique, les anneaux sont beaucoup plus faciles à reconnaître que les petits chiffres que l'on se trouve encore sur des résistances très anciennes.

La plupart des résistances ont quatre de ces anneaux de couleur. Les premiers deux anneaux de couleur désignent les numéros, le troisième un multiplicateur et le quatrième la tolérance. Cet anneau de tolérance est le plus souvent en or ou en argent - couleurs qui n'existent pas sur le premier anneau. En conséquence, le sens de lecture est toujours unique. La valeur de tolérance joue elle-même presque aucun rôle dans l'électronique numérique. Le tableau montre l'importance des anneaux colorés sur les résistances.

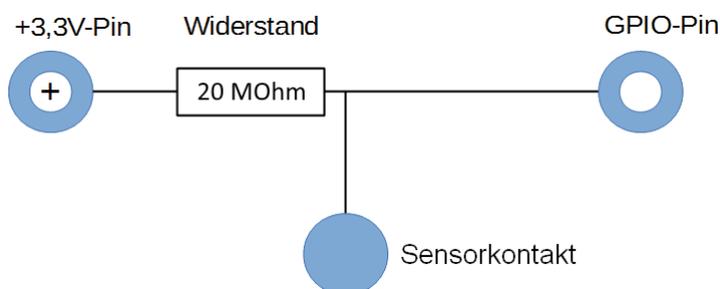
Le sens dans lequel une résistance est montée n'a pas d'importance. Pour les LEDs, cependant, le sens de montage joue un rôle important.

Capteur de contact en pâte à modeler

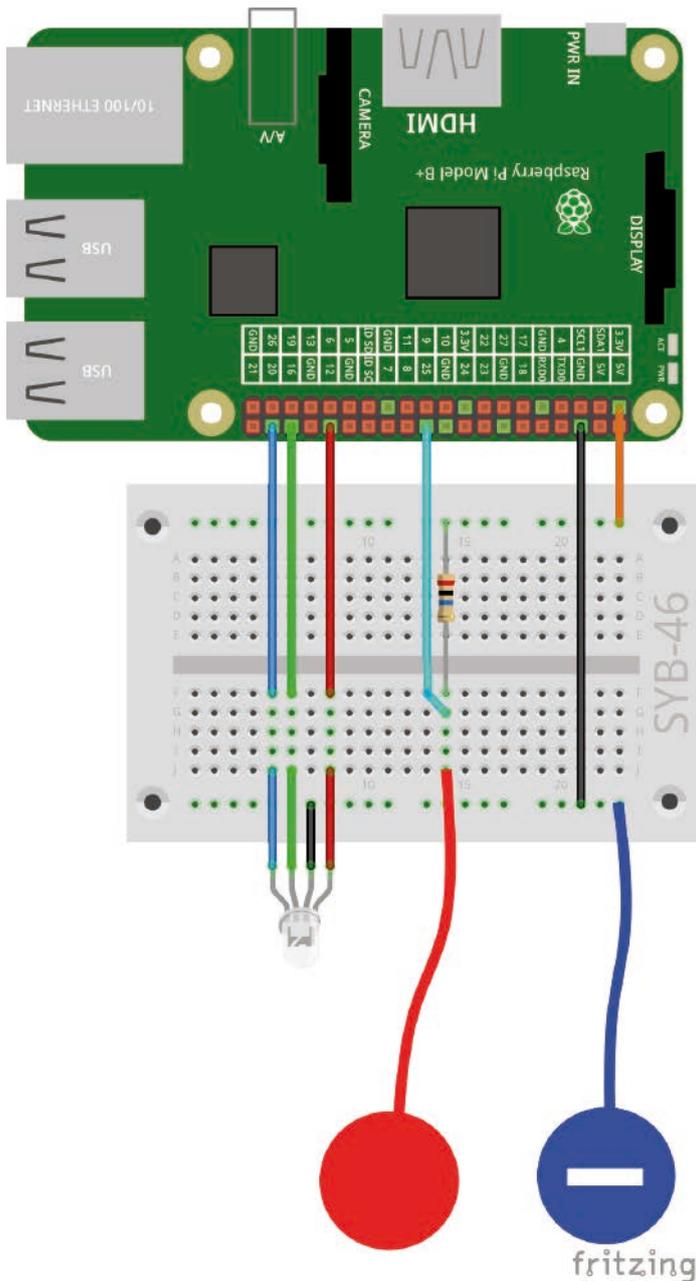
Les feux de circulation, ouvre-portes, interrupteurs de lumière et distributeurs automatiques sont aujourd'hui souvent contrôlés avec les capteurs qu'il suffit de toucher. Les boutons que l'on doit appuyer sont de plus en plus rares. L'expérience du 13ème jour commande un LED RVB via un capteur de contact.

La broche GPIO connectée comme entrée est reliée à une résistance de très haute impédance (20 MOhm) avec + 3,3 V, de sorte qu'un signal faible, mais clairement défini comme **Elevé** est produit. Une personne qui ne flotte pas librement dans l'air est toujours reliée à la terre et fournit un niveau **faible** à travers la peau conductrice d'électricité. Si cette personne touche un capteur de contact, au faible signal **Elevé** se superpose clairement le niveau **Bas** plus puissant de la main et tire la broche GPIO à **bas** niveau. Pour cela, la résistance pull-down à l'entrée GPIO doit être désactivée.

Couleur	Valeur de la résistance en ohm			
	1er Bague (Dizaine)	2ème Bague (Un)	3ème Bague (Multiplicateur)	4ème Bague (Toleranz)
Argent			10 ⁻² = 0,01	±10 %
Or			10 ⁻¹ = 0,1	±5 %
Noir		0	100 = 1	
Marron	1	1	10 ¹ = 10	±1 %
Rouge	2	2	10 ² = 100	±2 %
Orange	3	3	10 ³ = 1.000	
Jaune	4	4	10 ⁴ = 10.000	
Vert	5	5	10 ⁵ = 100.000	±0,5 %
Bleu	6	6	10 ⁶ = 1.000.000	±0,25 %
Violet	7	7	10 ⁷ = 10.000.000	±0,1 %
Gris	8	8	10 ⁸ = 100.000.000	±0,05 %
Blanc	9	9	10 ⁹ = 1.000.000.000	



Le niveau de la résistance entre la main et la masse dépend en réalité de beaucoup de choses, y compris des chaussures et du sol. La meilleure connexion à la masse est la marche pieds nus dans l'herbe humide, mais cela fonctionne bien aussi sur les sols en pierre. Les sols en bois isolent plus fortement, les revêtements de sol synthétiques sont souvent même positivement chargés. Dans le cas où le contact du capteur ne fonctionne pas, le circuit contient un second contact en pâte à modeler qui raccorde à la borne de terre de la platine enfichable Touchez en même temps celle-ci et le capteur. Dans ce cas la liaison à la masse est bien établie.



La pâte à modeler commande un LED RVB.

La pâte à modeler conduit l'électricité aussi bien que la peau humaine. On peut facilement lui donner la forme souhaitée et un contact de pâte à modeler est plus facile à toucher qu'un simple morceau de fil. La surface avec laquelle la main touche le contact est beaucoup plus importante. Ainsi, il est plus difficile d'avoir un « mauvais contact ». Insérez un morceau de fil dénudé dans un morceau de pâte. Branchez l'autre extrémité de fil sur la platine enfichable.

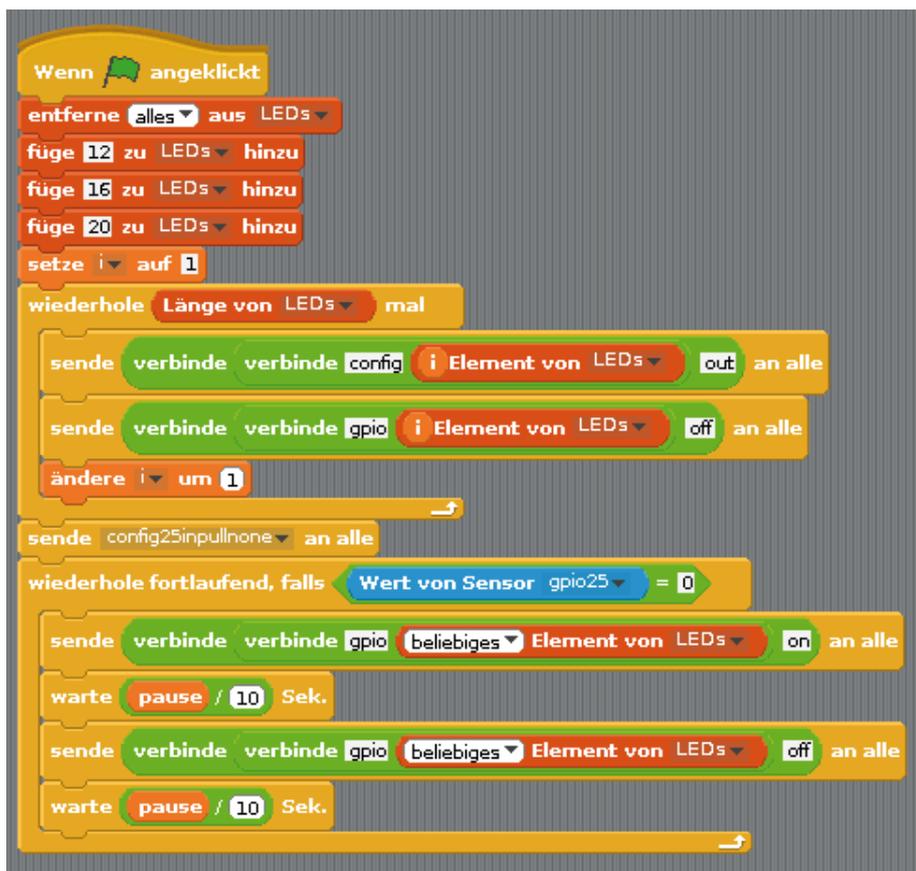
Le LED rouge clignote au hasard

Dans l'expérience du 13ème jour, un LED RVB clignote dans des couleurs aléatoires tant que l'on touche le contact du capteur.

Composants : 1 platine enfichable SYB-46 ; 1 LED RVB avec résistance en série ; 1 résistance 20 MOhm ; 6 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 4 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 2 contacts de pâte à modeler

Le programme

Le programme `13rgbled13` utilise une liste pour les trois broches GPIO du LED RVB. Une boucle initialise les broches comme des entrées et les éteint toutes. Le bloc **Envoi config25inpullnone** initialise la broche GPIO 25 pour le contact de capteur comme entrée sans résistance pull-down.



Le programme 13rgbLed13 commande un LED RVB Servo via un contact de capteur.

Le LED RVB doit clignoter uniquement lorsque l'on touche le contact du capteur. C'est ce que fait une boucle du type **Répéter en continu si ...** Si l'on touche le contact du capteur, il est relié à la terre et le signal à l'entrée GPIO passe à 0. La condition de la boucle utilise le bloc **Valeur du capteur...** Démarrez le programme encore inachevé avec un clic sur le drapeau vert, pour initialiser l'entrée GPIO, afin qu'il apparaisse dans la liste de sélection des capteurs.

À chaque cycle de boucle, un composant de couleur choisi au hasard pour allumer le LED rouge. Pur cela, on utilise un bloc **Élément arbitraire de LEDs**.

Comment sont formés les nombres aléatoires ?

On pense souvent que rien ne peut survenir au hasard dans un programme – alors, comment un programme peut-être en mesure de générer des nombres aléatoires ? Si on divise un grand nombre premier avec une valeur quelle qu'elle soit, à partir de l'éniesième chiffre décimal, on obtient des nombres qui sont difficilement prévisibles. Ils changent aussi sans aucune régularité lorsque vous augmentez régulièrement le diviseur. Ce résultat est certes aléatoire en apparence, mais il est possible de le reproduire par un programme identique ou en rappelant plusieurs fois le même programme à tout moment. Mais si on prend un nombre composé de certains de ces chiffres et qu'on le divise à son tour par un nombre, qui résulte de la seconde actuelle du temps ou du contenu de n'importe quel emplacement de mémoire de l'ordinateur, on obtient un résultat qu'on ne peut pas reproduire et qu'on peut donc désigner comme un nombre aléatoire.

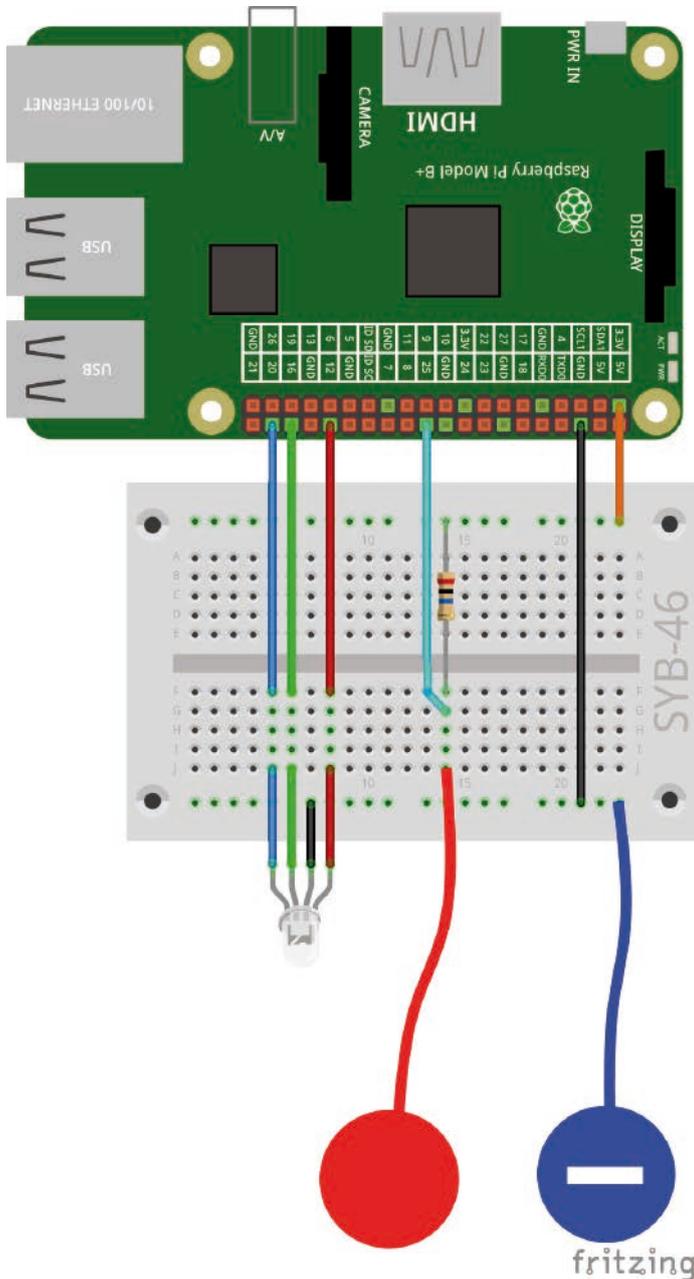
Après une pause à nouveau réglable via un variateur, un composant coloré du LED RVB sélectionné au hasard est éteint. Il peut arriver que, par l'intermédiaire de l'algorithme utilisé, le LED RVB s'éteigne complètement brièvement à court terme ou bien montre deux fois la même couleur.

14. Jour

14ème Jour

Aujourd'hui dans le calendrier de l'Avent

• 1 x pâte à modeler



La pâte à modeler commande un LED RVB.

Avec la deuxième pâte à modeler, vous pouvez construire les deux contacts de capteurs sur les broches GPIO et la broche de terre en différentes couleurs.

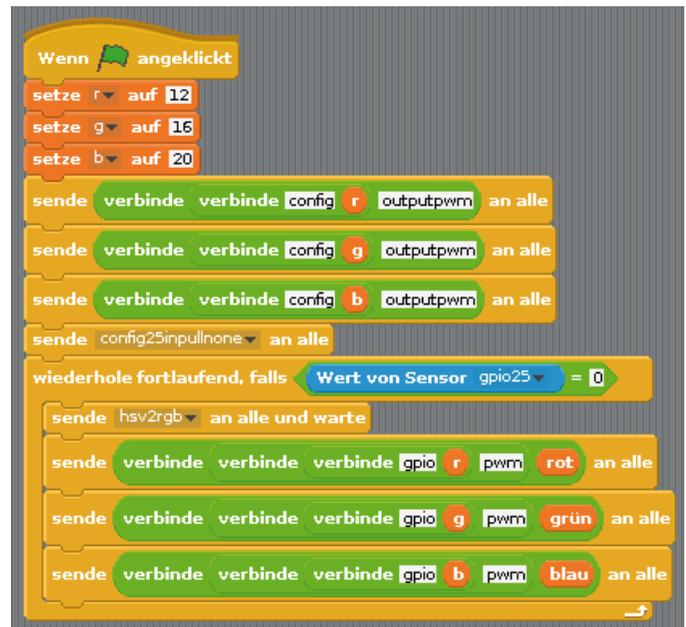
Effets de lumière LED RVB

Le programme du 14ème jour permet d'allumer le LED RVB dans l'une des couleurs de l'arc-en-ciel en touchant le contact du capteur. La couleur est sélectionnée par une commande unique au lieu de trois valeurs rouges. Si vous relâchez le capteur la dernière couleur allumée reste.

Composants : 1 platine enfichable SYB-46 ; 1 LED RVB avec résistance en série ; 1 résistance 20 MOhm ; 6 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 4 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 2 contacts de pâte à modeler

Le programme

Le programme 14rgbLed14 se compose de deux blocs de programmes indépendants. Le bloc qui démarre en cliquant sur le drapeau vert correspond largement aux programmes déjà connus. Les trois broches des LEDs RVB sont initialisées comme sorties PWM, la broche 25 comme entrée pour le capteur de contact sans résistance pull-down.



Le programme 14rgbLed14 permet d'allumer le LED rouge avec les couleurs vives de l'arc-en-ciel.

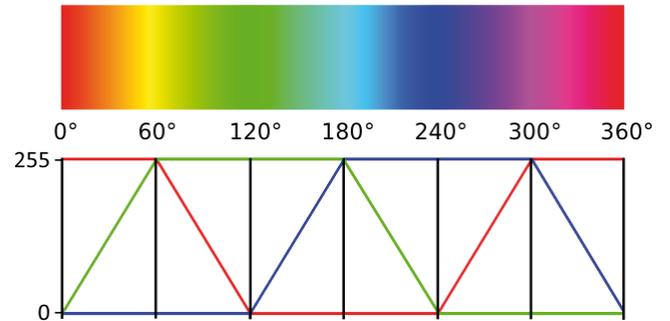
Une boucle infinie vérifie si l'on touche le contact du capteur. Si c'est le cas, un bloc **Envoyer ... à tous et attendre** le message **hsv2rgb**. De cette façon il est possible de construire des sous-programmes dans Scratch que l'on peut appeler à partir du programme principal. Le programme principal attend jusqu'à ce que le sous-programme soit terminé et définit ensuite les broches GPIO du LED RVB sur les valeurs PWM calculées.

Système HSV - couleur rouge

Le système de couleur RVB qui était utilisé dans tous les programmes jusqu'à présent, décrit les couleurs comme les trois composants rouge, vert et bleu qui sont mélangés entre eux. Pour les personnes, il est relativement difficile d'imaginer une couleur mélangée. En revanche, le système de couleur HSV décrit les couleurs au-delà des valeurs H = teinte (couleur), saturation (saturation) S = et V = valeur (valeur de luminosité).

Un simple changement de la valeur de H permet de décrire toutes les couleurs du spectre des couleurs avec une pleine intensité, si vous réglez les deux autres valeurs au maximum.

Le message **hsv2rgb** démarre un bloc de programme qui calcule les trois composants de couleurs R, G et B à partir de la valeur de H. La valeur de H peut prendre des valeurs situées entre 0 et 360 selon le nombre de degrés sur une roue chromatique. Les valeurs de S = saturation (saturation) et V = valeur (valeur de luminosité) sont automatiquement définies à la valeur maximale.



Le graphique montre comment la valeur de H d'une couleur HSV est calculée en valeurs rouge.

```

Wenn ich hsv2rgb empfangt
  setze x auf 1 - Betrag von h / 60 mod 2 - 1 * 255
  falls h < 360
    setze rot auf 255
    setze grün auf 0
    setze blau auf x
  falls h < 300
    setze rot auf x
    setze grün auf 0
    setze blau auf 255
  falls h < 240
    setze rot auf 0
    setze grün auf x
    setze blau auf 255
  falls h < 180
    setze rot auf 0
    setze grün auf 255
    setze blau auf x
  falls h < 120
    setze rot auf x
    setze grün auf 255
    setze blau auf 0
  falls h < 60
    setze rot auf 255
    setze grün auf x
    setze blau auf 0
  
```

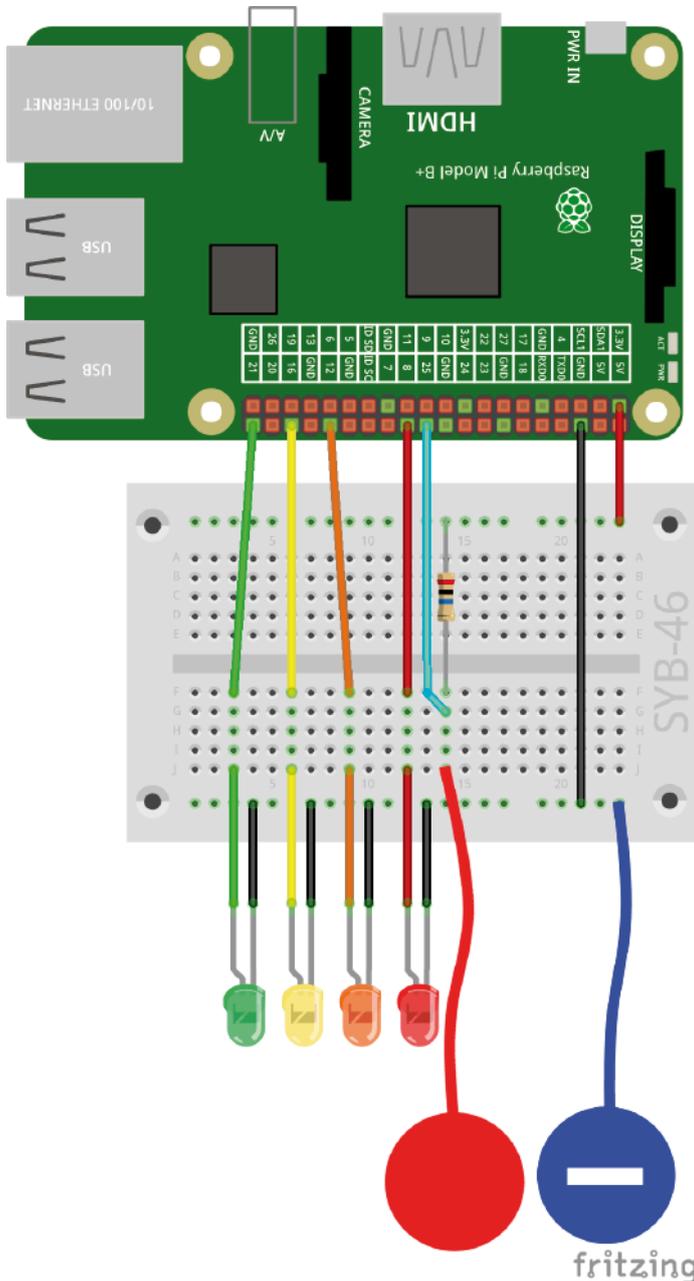
Sous-programme pour calculer HSV en valeurs rouges.

Le bloc **Si je reçois ...** démarre un sous-programme si un message déterminé a été reçu. Ce message peut être envoyé par un bloc **Envoyer ... à tous et** ou **Envoyer ... à tous et attends**.

15ème Jour

Aujourd'hui dans le calendrier de l'Avent

• 1 LED orange avec résistance en série



La pâte à modeler commande un chenillard.

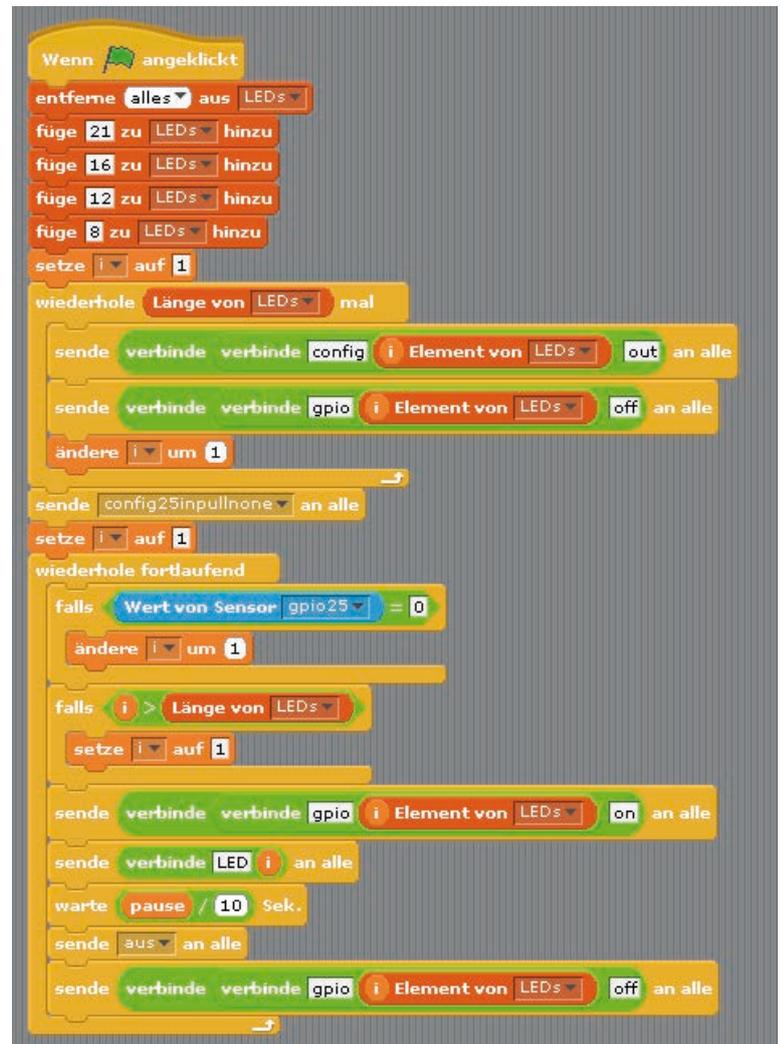
Commander un chenillard avec un contact de la pâte à modeler

L'expérience du 15ème jour est à nouveau un chenillard qui est contrôlé cette fois via un contact de capteur contact en pâte à modeler. En plus des vrais symboles de LEDs sur la scène de la crèche de Noël, des symboles colorés s'allument sur la scène Scratch. Tant que touche sur le contact de capteur, les LEDs clignotent successivement sur le chenillard. Si on lâche un contact de capteur, le dernier LED allumé continue de s'allumer.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED orange avec résistance en série ; 1 LED rouge avec résistance en série ; 1 résistance de 20 MOhm ; 7 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 8 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 2 contacts en pâte à modeler

Le programme

Le programme 15led15 se compose de plusieurs blocs indépendants. Le bloc de programme de la scène démarre en cliquant sur le drapeau vert, crée une liste avec les broches GPIO des quatre LEDs, les initialise comme sorties et la broche 25 broches comme entrée sans résistance pull-down.



Le bloc de programme de la scène contient la boucle principale du programme.

Une boucle infinie demande d'abord l'on touche le contact du capteur. Si c'est le cas, le compteur **i** augmente à 1, pour allumer le LED successif de la liste.

Si **i** est plus grand que la longueur de la liste, **i** est réglé à nouveau sur 1 et le chenillard recommence par le premier LED.

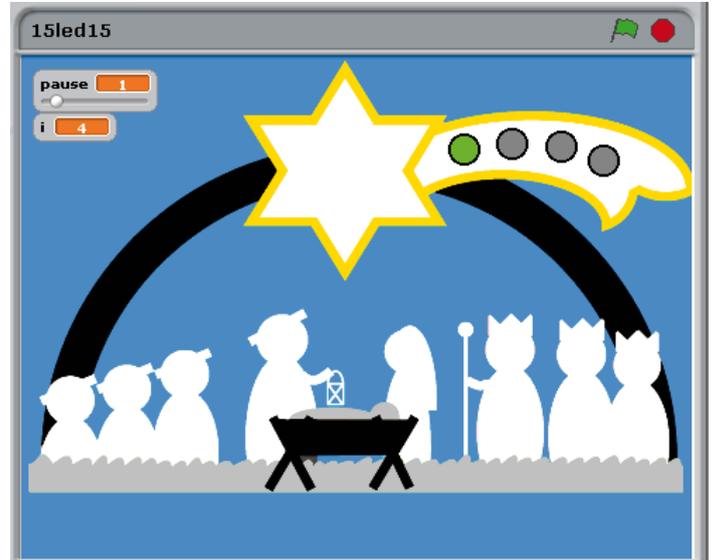
Maintenant la broche GPIO correspondant à la variable **i** dans la liste s'allume. Ensuite, un bloc **Raccorder...** construit un message à partir du mot **LED** et du numéro actuel issu de la variable **i** : **LED1**, **LED2**, **LED3** ou **LED4**. Ce message est envoyé pour allumer le symbole du LED sur la scène Scratch.

Après une pause réglable, le message **éteint** est envoyé pour éteindre tous les symboles LED sur la scène Scratch. En revanche pour l'allumage, aucun numéro de LED ne doit être spécifié. Le programme éteint simplement tous les symboles du LED, indépendamment du fait si elles étaient déjà allumées ou non.

Ensuite, le LED allumé est éteint via la broche GPIO avant le début du prochain cycle de la boucle.

Les quatre symboles LED sur la queue de l'étoile sont quatre objets Scratch. Chargez l'image `kreis_gruen.png` comme un nouvel objet dans la scène. Une fois que l'objet est construit, les autres objets sont dupliés.

Sélectionnez le nouvel objet dans la liste d'objets et copiez un second costume dans l'onglet **Costumes**. Remplissez le circuit dans le programme de peinture avec de la couleur grise.



Les quatre LEDs sur la scène Scratch.



Les costumes du symbole LED vert.



Les blocs de programme du symbole LED vert.

Renommez l'objet en **vert** et les deux costumes avec **allumé** et **éteint**. Ensuite, appliquez deux blocs de programme pour cet objet dans l'onglet **Scripts**.

Si l'objet reçoit le message **LED1**, le costume **allumé** est changé. Le symbole s'allume en vert.

Si l'objet reçoit le message **éteint**, le costume **éteint** est changé. Le symbole apparaît en gris.

Dupliquez l'objet avec en cliquant trois fois avec le bouton droit sur la palette de l'objet. Renommez le nouvel objet **jaune**, **orange** et **rouge** et déplacez-les dans les positions appropriées sur la queue de l'étoile.

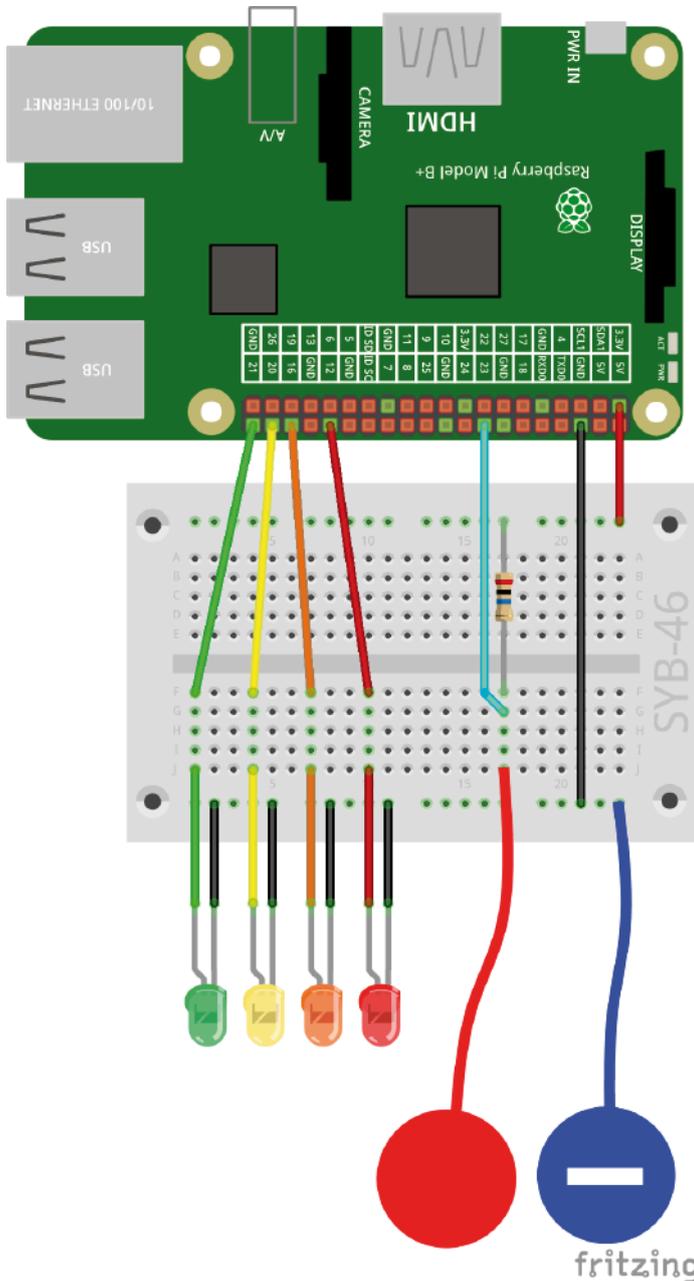
Cliquez pour chaque objet dans le costume **allumé** sur le bouton **Modifier** et remplissez le cercle de jaune, orange ou rouge. Puis sur l'onglet **Scripts**, modifiez le message dans le premier bloc sur **LED2**, **LED3** et **LED4**.

Démarrez le programme en cliquant sur le drapeau vert et touchez le contact de capteur. Le chenillard commence.

16ème Jour

Aujourd'hui dans le calendrier de l'Avent

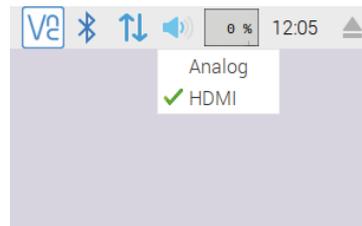
- 4 câbles de raccordement GPIO (long)



Le contact en pâte à modeler commande quatre LEDs.

Le Raspberry Pi émet des sons

Le Raspberry Pi peut reproduire via un moniteur HDMI, un haut-parleur ou casque externe de la musique sur la prise jack analogique 3,5 mm. Pour les moniteurs d'ordinateurs avec connexion DVI, connectés à la sortie HDMI, un haut-parleur d'ordinateur doit être connecté à la sortie analogue car le signal audio n'est pas transmis via le câble DVI. Cliquez avec le bouton droit de la souris en haut à droite sur le symbole du haut-parleur pour sélectionner la sortie audio.



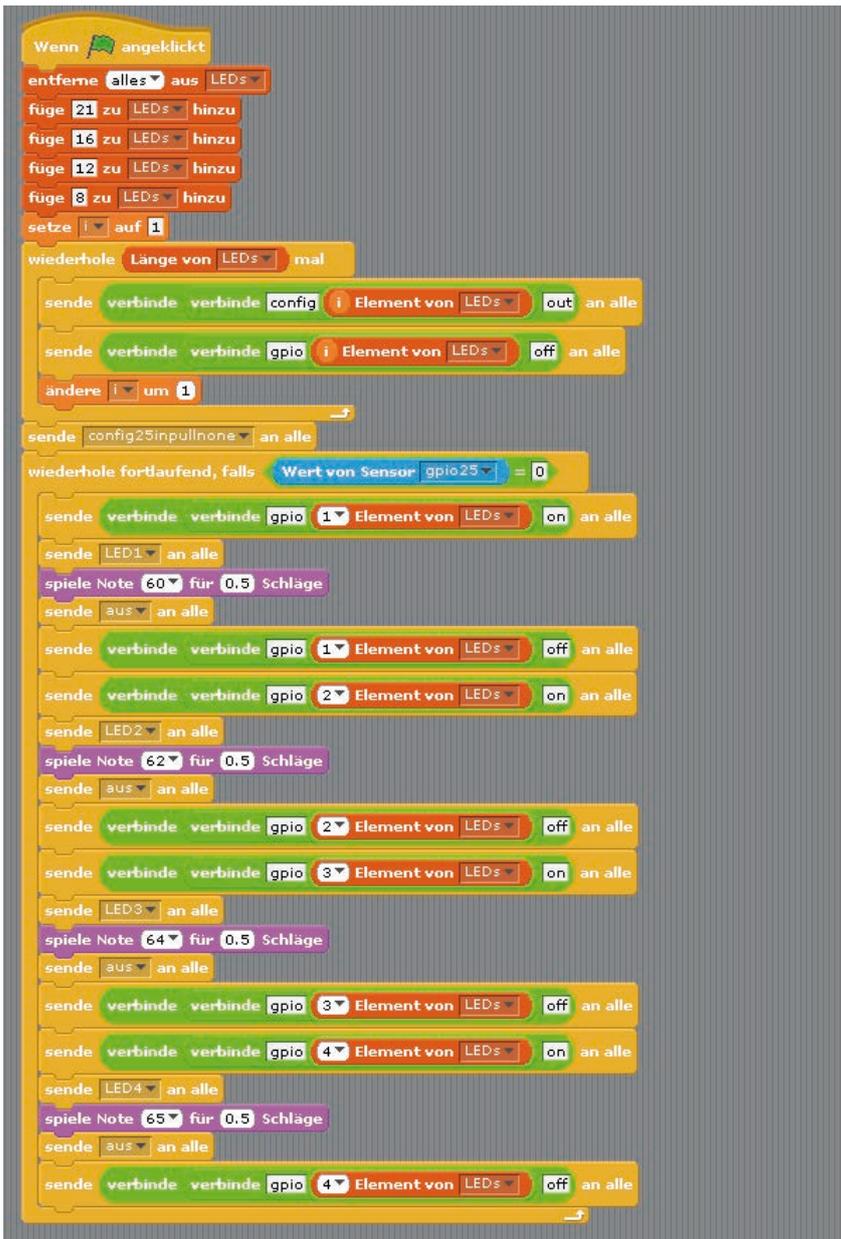
Sélectionner la sortie audio.

Chaque fois que vous touchez le contact de pâte à modeler, le Raspberry Pi joue une séquence sonore et permet d'allumer les LEDs en succession.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED orange avec résistance en série ; 1 LED rouge avec résistance en série ; 1 résistance de 20 MOhm ; 7 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 8 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 2 contacts en pâte à modeler

Le programme

Le programme 16led16 fonctionne de manière semblable au programme de la journée précédente et permet de faire clignoter quatre LEDs l'un après l'autre clignotantes à courts intervalles lorsque l'on touche le contact de la pâte. Pour chaque LED un signal sonore est émis.



Le programme 16Led16 permet de faire clignoter quatre LEDs à intervalles courts et de jouer des sons.

Encore une fois, les numéros des LEDs sont enregistrés dans une liste. Dans la boucle principale du programme se trouvent pour chaque LED les propres blocs de programme afin de pouvoir déterminer dans le bloc **Joue une note ... pour ... battement** le son de chaque LED interactivement via un clavier. Le programme utilise les notes C, D, E et F.

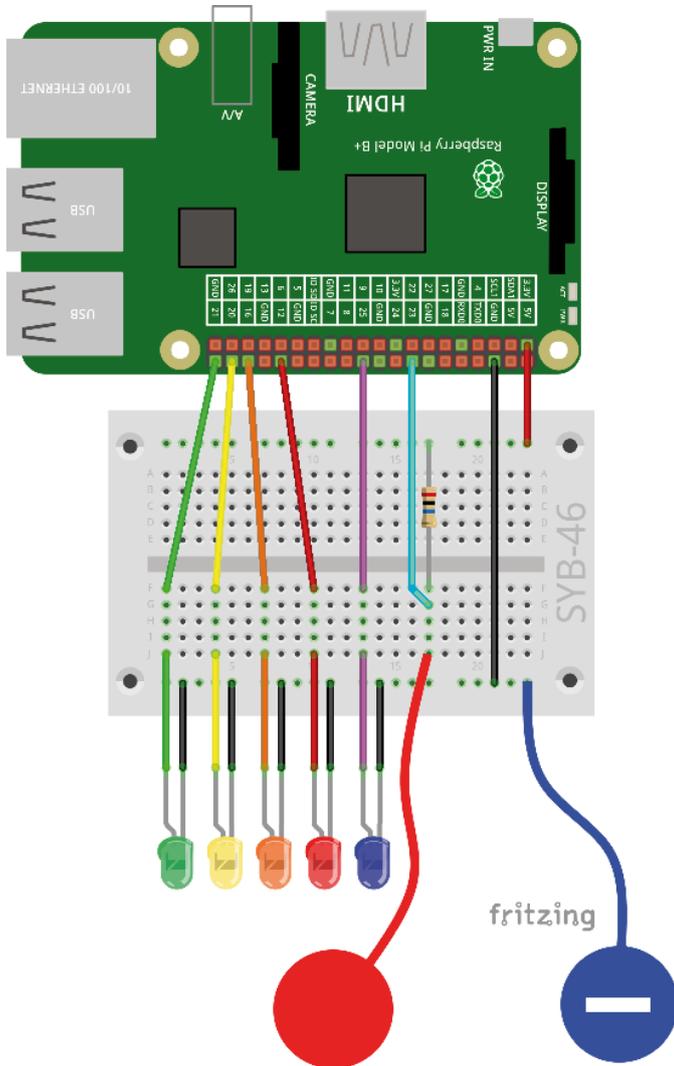


Les quatre notes utilisées.

17ème Jour

Aujourd'hui dans le calendrier de l'Avent

- 1 LED lilas avec résistance en série



Chenillard avec cinq LEDs

note ... pour ... Le programme utilise les notes **C(60)**, **D(62)**, **E(64)**, **F(65)**, **G(67)**.

L'initialisation des broches GPIO a lieu comme dans le précédent programme. Dans chaque cycle, quand le contact de capteur est touché, la boucle principale du programme règle le compteur **i** sur **1**. Une boucle interne s'exécute cinq fois et dans chaque cycle, allume l'un des cinq LEDs et le symbole correspondant sur la scène. Ensuite, la note enregistrée sous le numéro approprié retentit dans la liste **Son**. Ce son définit en même temps la durée d'allumage du LED. Puis les symboles de LED et les LEDs sont éteintes, le compteur de la boucle est incrémenté de 1 et le cycle suivant démarre avec le prochain LED.

Chenillard étendu

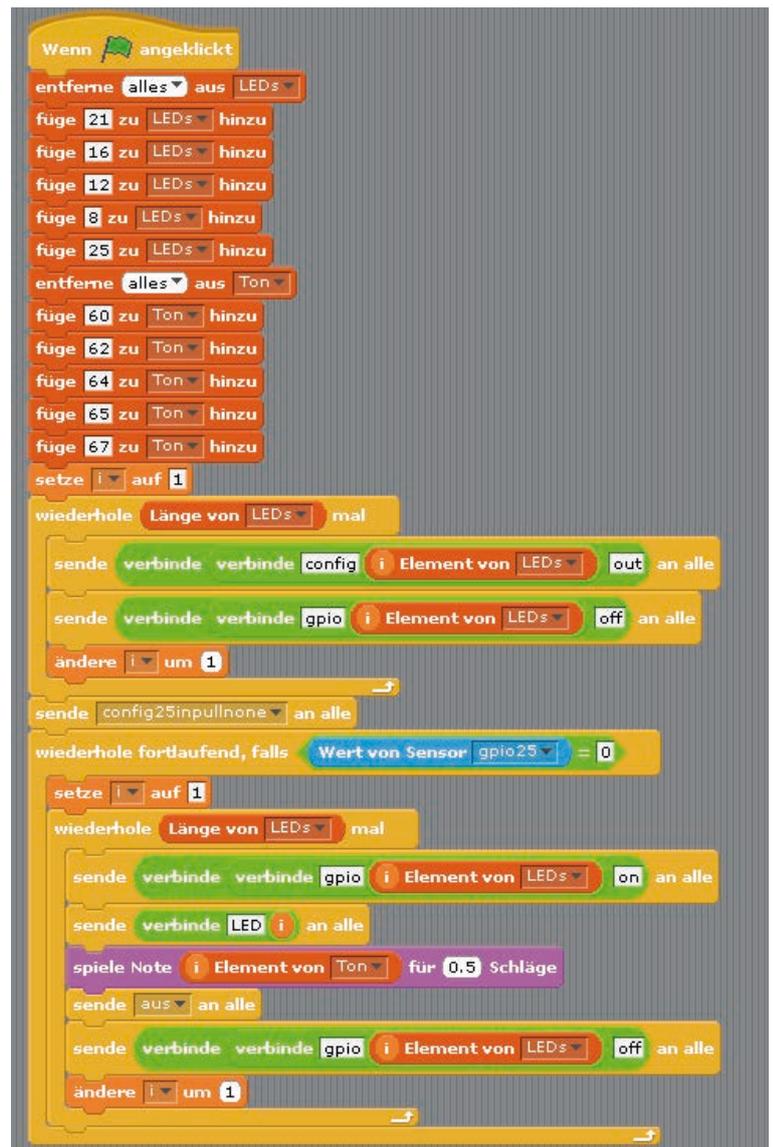
L'expérience du 17ème étend le chenillard à un cinquième LED Aussi ici, une séquence sonore est jouée.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED orange avec résistance en série ; 1 LED rouge avec résistance en série ; 1 LED lilas avec résistance en série ; 1 résistance de 20 MOhm ; 8 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 10 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 2 contacts en pâte à modeler

Le programme

Dans le programme 17led17 les notes sont aussi jouées via une liste, ce qui raccourcit considérablement la boucle principale du programme.

Au début, en plus d'une liste **LEDs** une liste **Sonest** créée avec des notes qui sont jouées pour chaque LED. Pour cette méthode, vous ne disposez d'aucun clavier. Au mieux, essayez les notes avec un **coup Joue la**



Chenillard de cinq LEDs avec une séquence sonore.

18ème Jour

18. Jour

Aujourd'hui dans le calendrier de l'Avent

- 3 câbles de raccordement GPIO (courts)

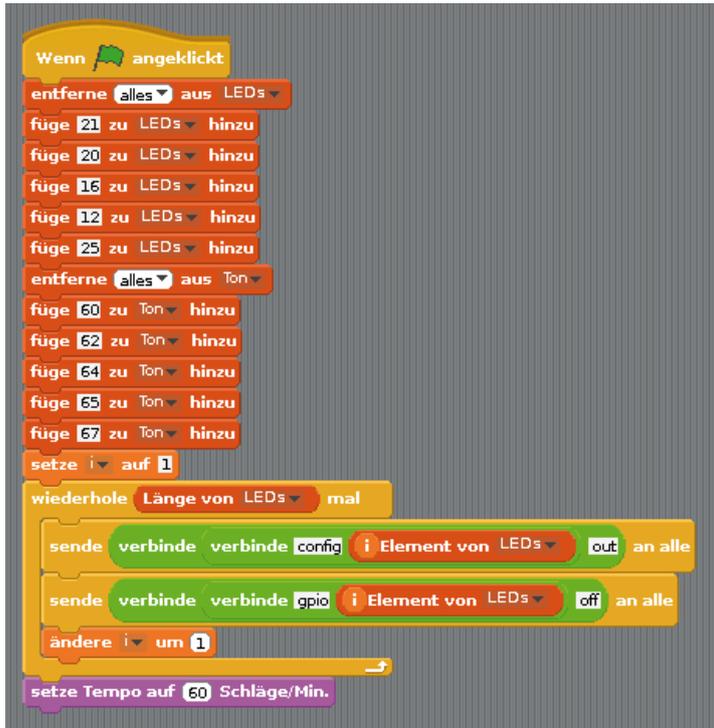
Trois modèles différents de chenillard

Le programme du 18ème jour montre sur les cinq LEDs à côté du chenillard simple deux autres modèles de lumière. Le programme 18Led18 comprend pour cela trois boutons sur la scène Scratch avec lesquels vous pouvez activer ces modèles.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED orange avec résistance en série ; 1 LED rouge avec résistance en série ; 1 LED lilas avec résistance en série ; 6 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 10 câbles de connexion GPIO longs (platine enfichable - LEDs)

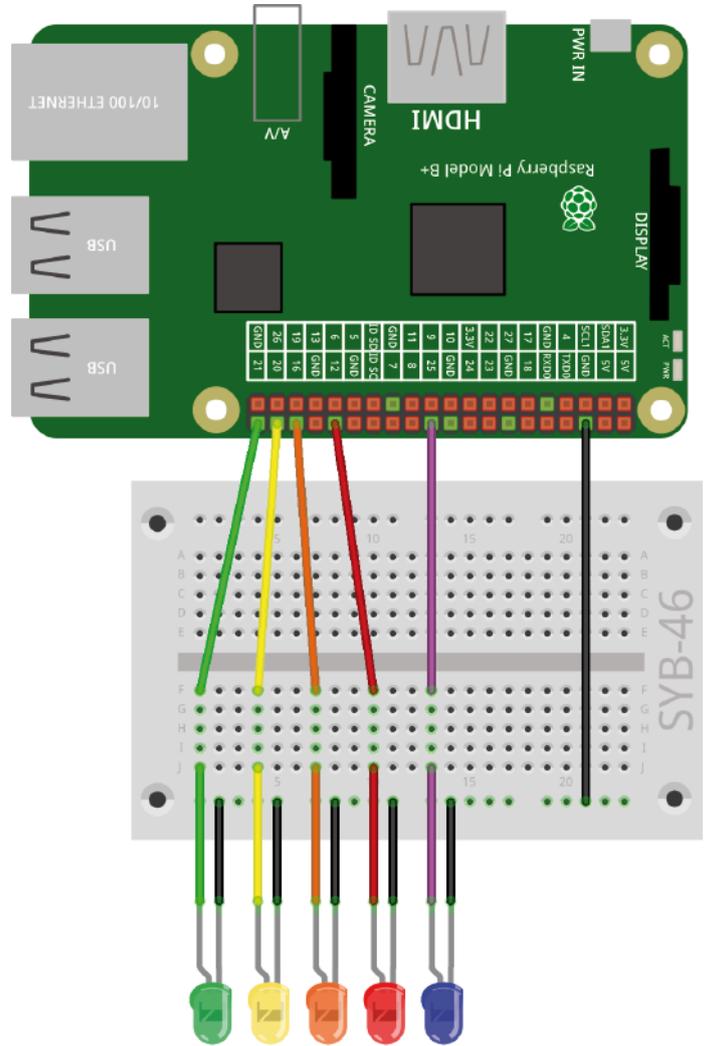
Le programme

Le programme 18Led18 initialise sur la scène uniquement les deux listes pour LEDs et les notes ainsi que les broches GPIO. À la fin une vitesse est définie pour la musique. De la musique est jouée à un rythme qui est spécifié dans ce que l'on appelle des battements, tandis que toutes les autres indications de temps dans Scratch sont mesurées en secondes. Le bloc **règle le rythme sur ... battements/min.** définit la vitesse pour la reproduction de la musique.

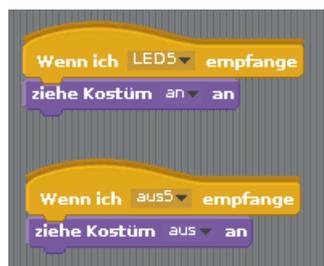


Les blocs de programme de la scène.

Le programme contient, comme les programmes précédents, cinq objets pour les cinq LEDs. Afin que ceux-ci puissent être commandés dans différents modèles, bien que pas tous sont toujours éteints à la fois, chaque objet réagit à son propre signal de commutation, sur la figure par ex. éteint5.

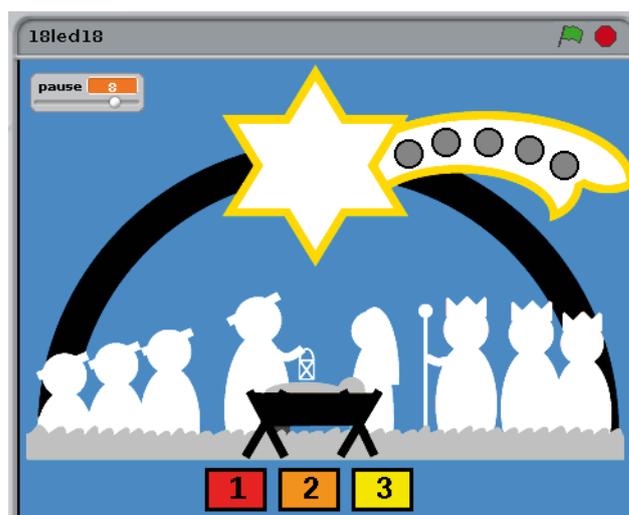


Le chenillard s'allume dans divers modèles.



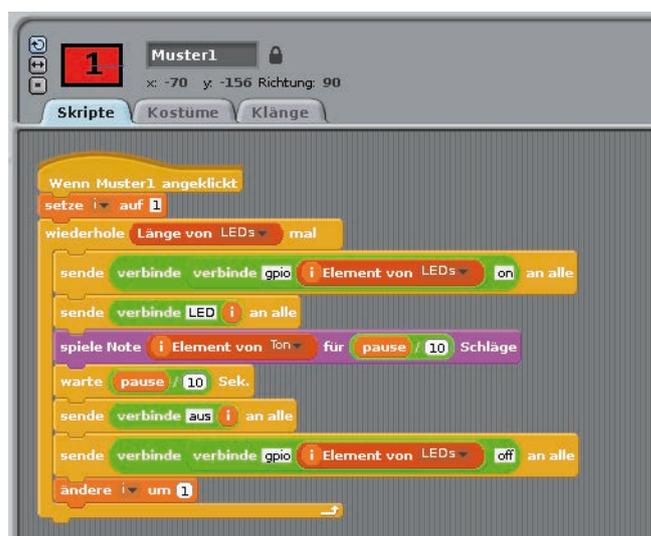
Blocs de programme d'objet lilas.

En plus des objets, pour les LEDs, il y a encore trois boutons dans le programme. Ce sont les propres objets qui jouent lorsque vous cliquez sur un des trois modèles prédéfinis de LED. Le temps entre les processus de clignotement individuels est à nouveau réglé via variable **pause** réglable interactivement.



Les trois boutons commutent le modèle de clignotement des LEDs.

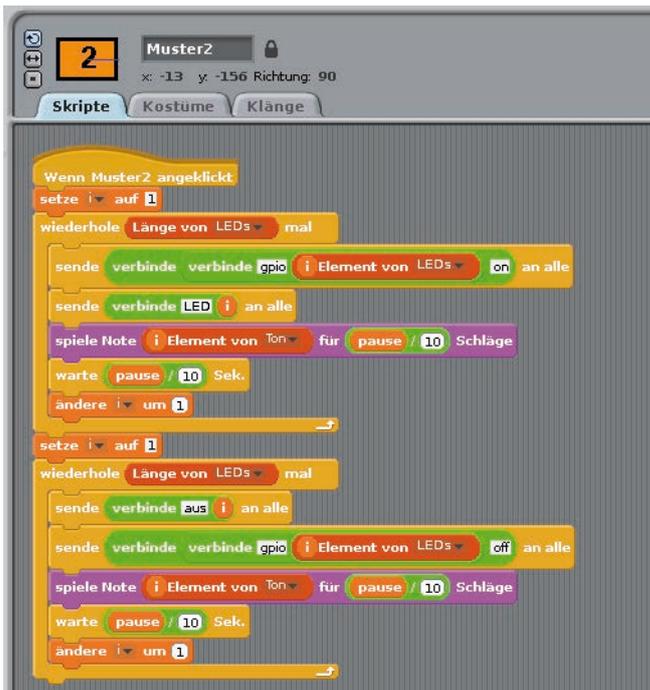
Le bouton avec le numéro 1 démarre en cliquant sur la séquence de chenillard. Un LED après l'autre est allumé et éteint après la pause définie.



Les blocs de programme pour le bouton 1.

Après avoir cliqué sur le bouton, une boucle s'exécute sur tous les éléments de la liste **LEDs**. Tout d'abord, un LED est allumé via sa broche GPIO. Un bloc **Envoyer ... à tous** envoie un message avec le numéro de LED pour allumer l'objet LED correspondant sur la scène.

Ensuite la note correspondante au LED est jouée issue de la liste **Son**. Le programme attend que les sons et le clignotement soient synchronisés. Ensuite, un bloc **Envoyer ... à tous** envoie un message avec le numéro de LED pour éteindre à nouveau l'objet LED. Dans ce modèle, tous les objets LED peuvent être facilement éteints. Mais comme cela ne fonctionne pas avec les autres modèles, le message pour éteindre contient cette fois aussi le numéro de LED. À la fin, le « vraie » LED est éteint via sa broche GPIO et le compteur de boucle augmente de 1.

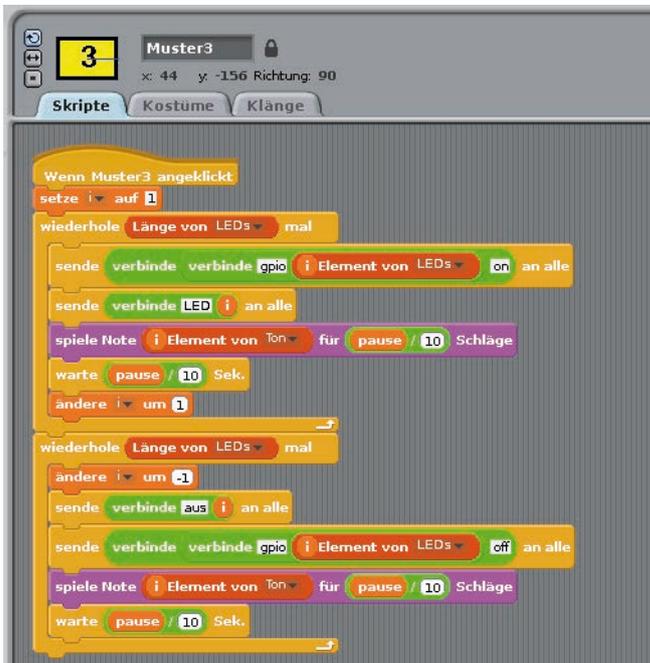


Les blocs de programme pour le bouton 2.

Le bouton avec le numéro 2 commence lorsque vous cliquez sur un LED après l'autre et les laisse allumés. Si tous les cinq LEDs sont allumés, ils s'éteignent à nouveau dans le même ordre.

En cliquant sur le bouton avec le numéro 2, deux boucles sont traitées l'une après l'autre. La première boucle allume un LED via sa broche GPIO, envoie le message pour allumer l'objet LED sur la scène, reproduit le son et attend.

Après que cette boucle se soit exécutée cinq fois, le compteur de boucle est réinitialisé à 1. Ensuite, les objets LED et les LED sont éteints à nouveau l'un après l'autre et les notes sont rejouées une fois.



Les blocs de programme pour le bouton 3.

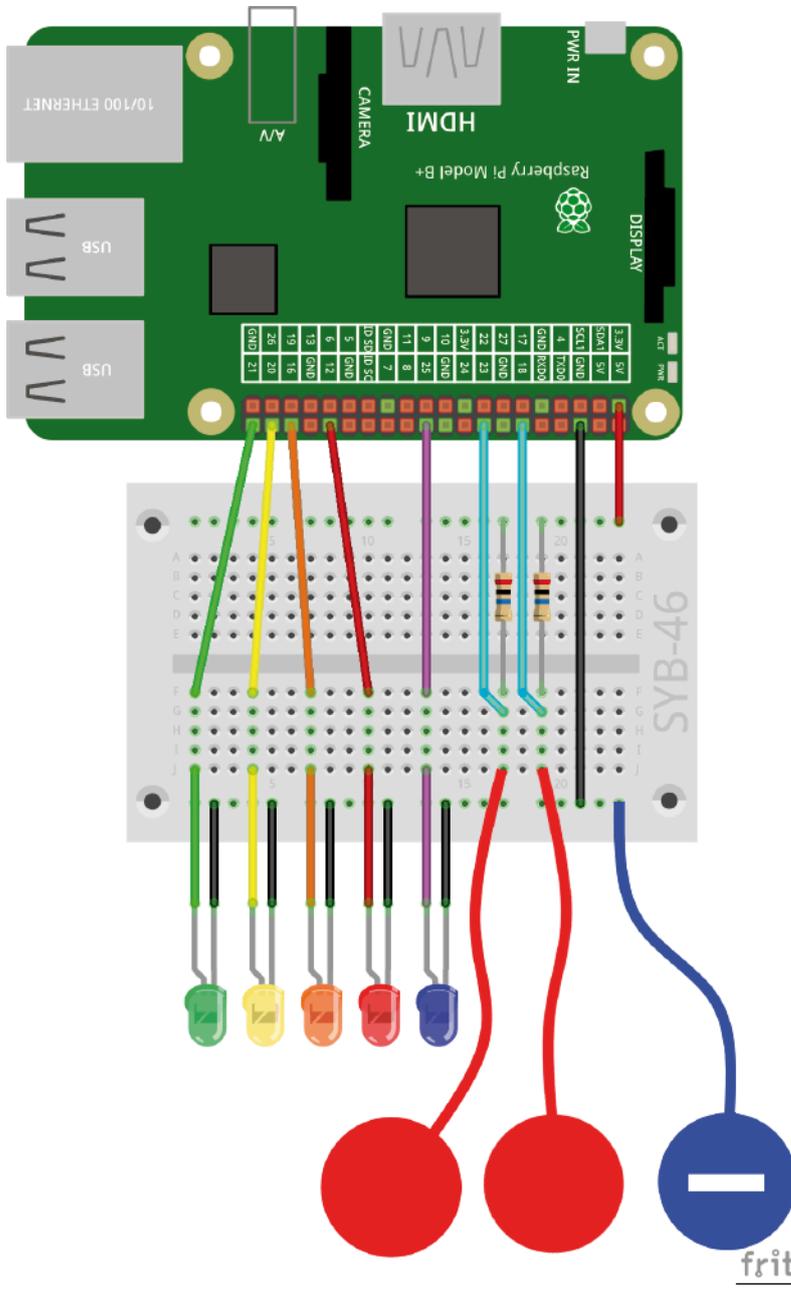
Le bouton avec le numéro 3 commence lorsque vous cliquez sur un LED après l'autre et les laisse allumés. Si tous les cinq LEDs sont allumés, ils sont éteints à nouveau dans l'ordre inverse.

La première boucle est la même que celle pour le bouton avec le numéro 2. Après les cinq cycles, le compteur de boucle reste sur 5 ensuite à chaque cycle de la deuxième boucle, il est réduit de 1. Ceci éteint les LEDs dans l'ordre inverse.

19ème Jour

Aujourd'hui dans le calendrier de l'Avent

- 1 résistance 20 MOhm (rouge noir bleu)



Deux contacts de pâte à modeler contrôlent les LEDs.

Chenillard ou clignotant

L'expérience du 19ème jour contient deux contacts de pâte à modeler et un contact de masse. Si on touche le contact de pâte à modeler sur la broche GPIO 23, un chenillard s'exécute selon le modèle choisi. Si on touche le contact de pâte à modeler sur la broche GPIO 18, toutes les LEDs s'allument brièvement.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED orange avec résistance en série ; 1 LED rouge avec résistance en série ; 1 LED lilas avec résistance en série ; 2 résistances de 20 MOhm ; 9 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 10 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 3 contacts en pâte à modeler

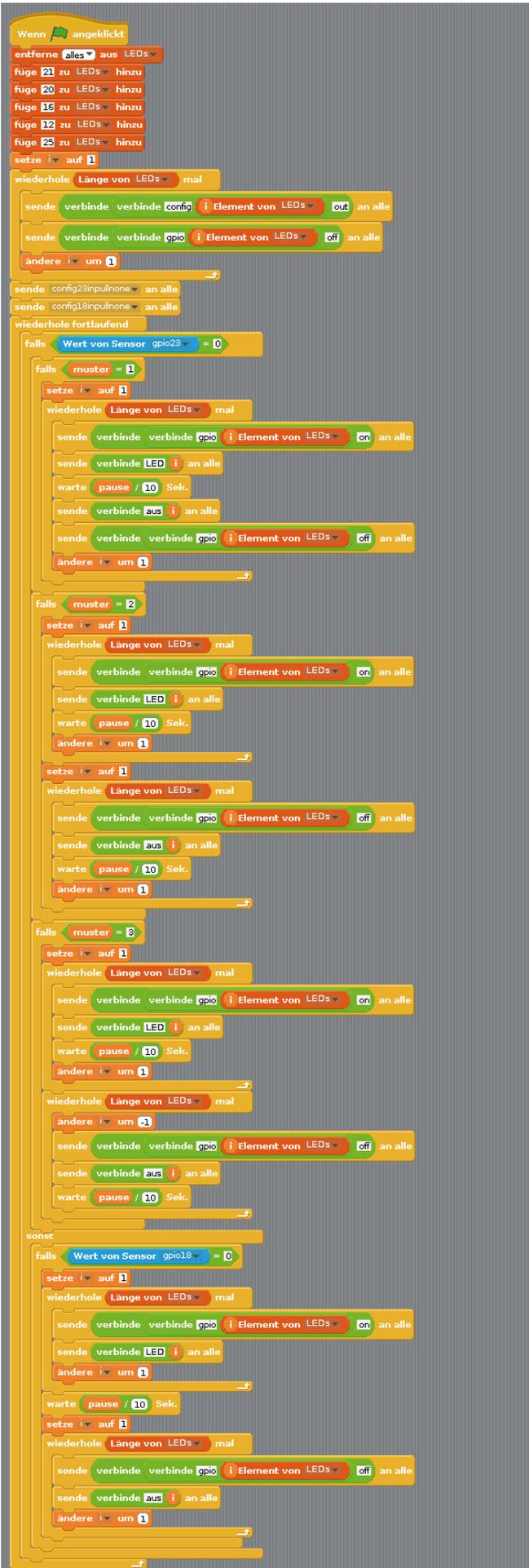
Le programme

Comme le chenillard dans le programme 19led19, contrairement au précédent programme, n'est pas déclenché directement en cliquant sur un bouton, mais attend que l'utilisateur touche un contact de pâte à modeler, il suit une logique de programme différente. En cliquant sur un bouton, il envoie un message avec le nom du modèle.

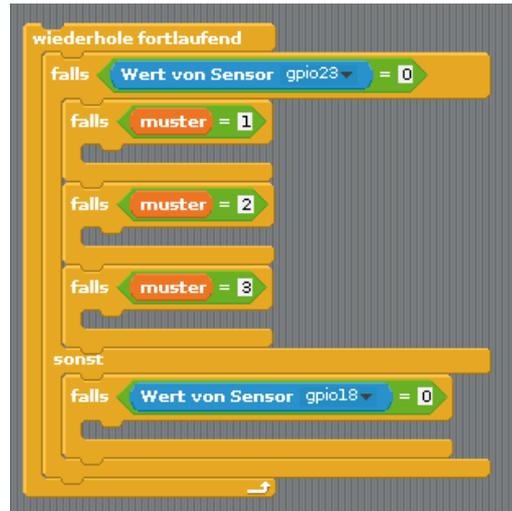


Bloc de programme de l'un des trois boutons.

Le programme principal sur la scène attend qu'un contact de pâte à modeler soit touché et évalue ensuite le modèle sélectionné. Pour le deuxième contact de pâte à modeler, une autre broche GPIO est initialisée comme entrée sans résistance pull-down.



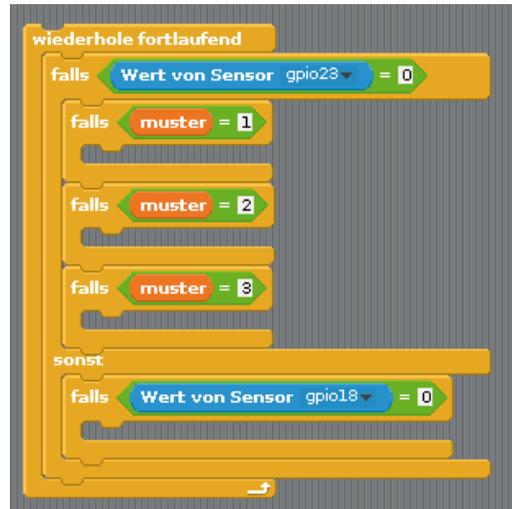
Le programme principal sur la scène.



Aperçu du montage de la boucle principale.

Si l'on touche le contact de pâte à modeler sur la broche GPIO 23, un modèle de chenillard est joué en fonction du dernier bouton cliqué. Comme dans les programmes précédents, les objets LED s'allument aussi sur la scène.

Dans le cas **sinon** - si ce contact de pâte à modeler n'est pas touché - une autre interrogation vérifie que le contact de pâte à modeler sur la broche GPIO 18 soit touché. Dans ce cas, tous les LEDs sont allumés puis à nouveau éteints après une courte pause.



Si on touche le contact de pâte à modeler sur broche GPIO 18, toutes les LEDs sont allumés et éteints à nouveau après une courte pause.

Comme l'espace devient restreint sur la platine enfichable, celle-ci et les photos des jours suivants montrent le montage de circuit en deux parties. Le câble du LED RVB, qui est inséré dans l'emplacement sur l'étoile au-dessus de la crèche, masqueraient les autres câbles dans le graphique.

Le programme

Le programme `20rgbLed20` initialise au début trois broches GPIO en plus comme sorties PWM pour le LED RVB.

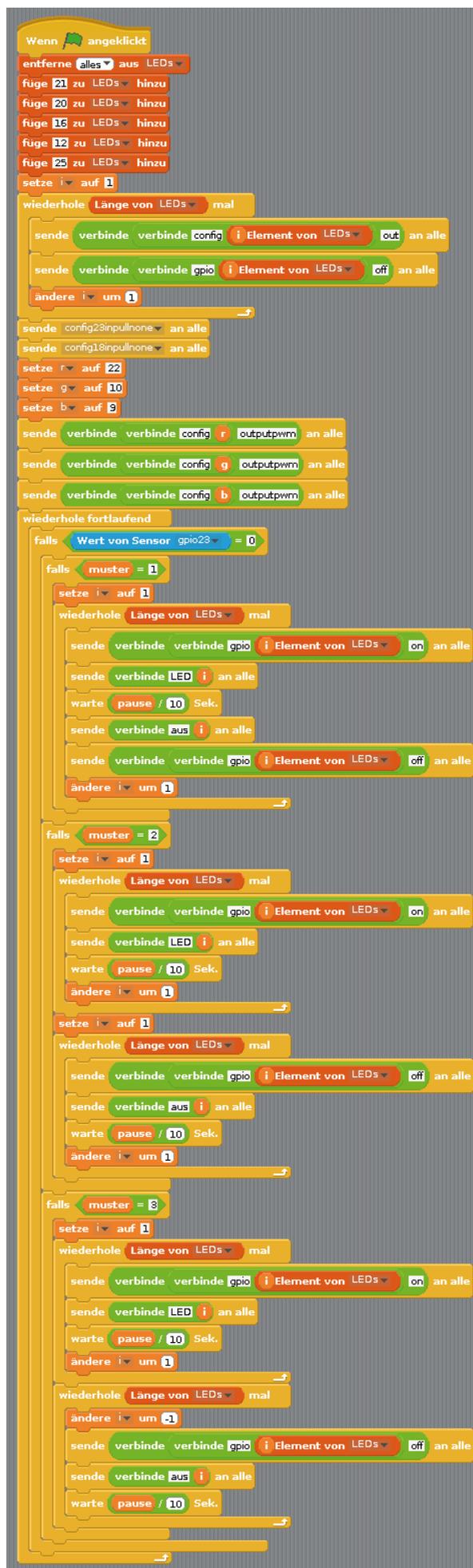
En outre, le sous-programme déjà connu pour calculer les valeurs de H en couleurs RVB est également sur la scène. Si vous assemblez le programme Scratch lui-même, vous pouvez importer ce bloc de programme d'un programme précédent.

Les blocs de programme pour le LED RVB sont attribués à l'objet **Etoile**, car celui-ci doit également changer lorsqu'on touche le contact de pâte à modeler sur la broche GPIO 18.

Une boucle augmente progressivement la valeur de H de 0 à 360. À la fin de la boucle recommence la boucle infinie et redémarre la boucle interne à nouveau depuis 0, ce qui entraîne un dégradé de couleurs continu.

Seulement lorsque l'on touche le contact de pâte à modeler sur la broche GPIO 18, la valeur actuelle de H à ce moment-là est convertie en RVB. L'étoile reçoit est le costume lumineux, qui représente un LED allumé, et la valeur calculée de RVB est représentée par l'intermédiaire de signaux PWM sur le LED RGB.

Tant que le contact de pâte à modeler n'est pas touché, l'objet **Etoile** utilise le costume gris et le LED RVB restera éteint.



Les blocs de programme sur la scène.

21ème Jour

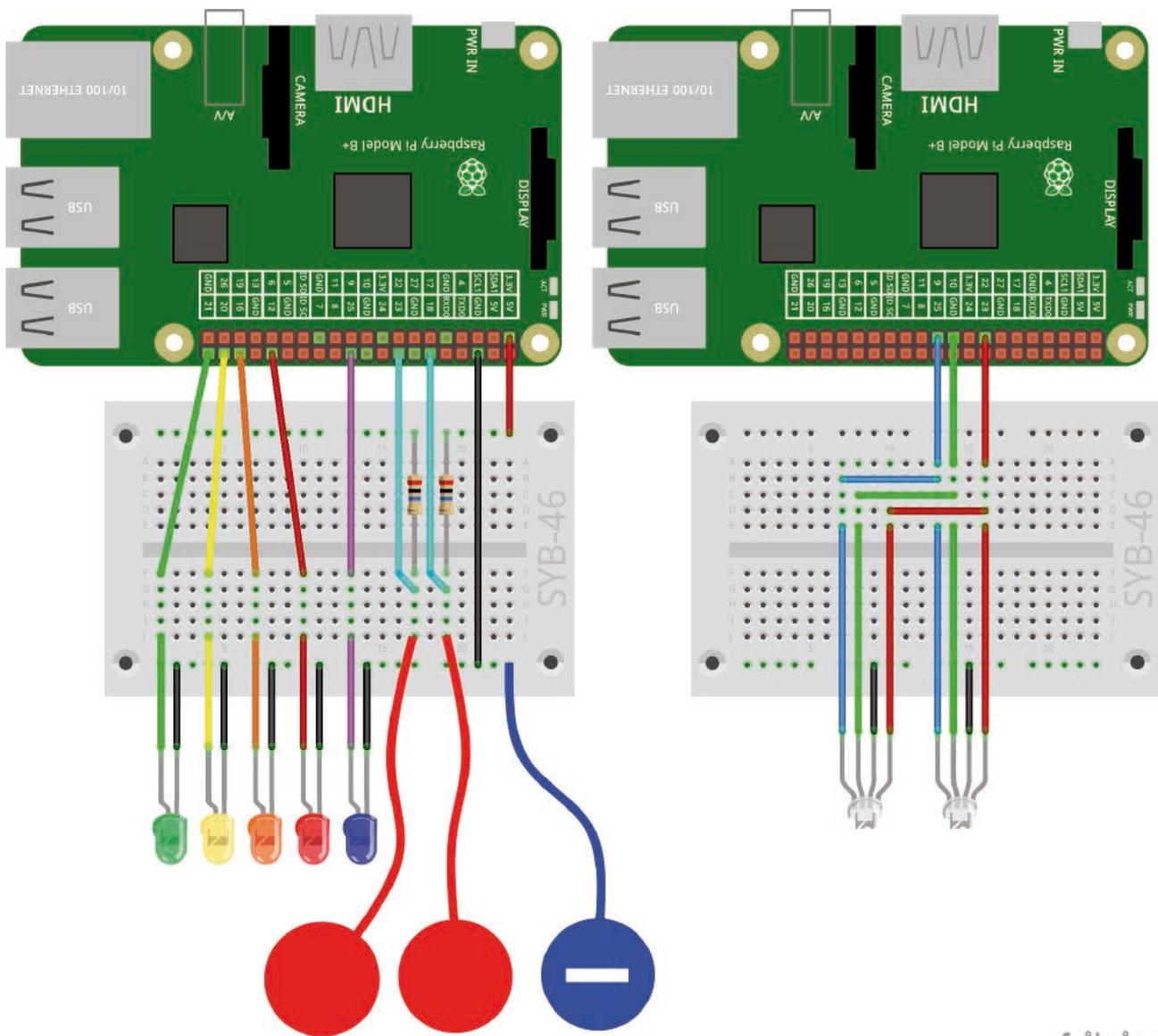
Aujourd'hui dans le calendrier de l'Avent

• 4 câbles de raccordement GPIO (long)

Jeux de couleur sur l'écran

L'expérience du 21ème jour est semblable à celle de la veille. Toutefois, deux LEDs RVB montrent le même dégradé de couleur et un point coloré sur l'étoile montre les mêmes couleurs que les LEDs RVB. Le contrôle des modèles de chenillard fonctionne comme dans les programmes précédents.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED orange avec résistance en série ; 1 LED rouge avec résistance en série ; 1 LED lilas avec résistance en série ; 2 LEDs RVB avec résistance en série ; 2 résistances de 20 MOhm ; 12 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 18 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 3 contacts en pâte à modeler



fritzing

Deux contacts de pâte à modeler contrôlent le chenillard et deux LEDs RVB connectés en parallèle.

Les deux LEDs RVB sont connectés par l'intermédiaire de cavaliers sur les mêmes broches GPIO et montrent donc toujours la même couleur.

Le programme

Le changement de couleur sur l'étoile, en raison du grand nombre de couleurs différentes, n'est pas représenté via des costumes, mais à travers le stylo dans Scratch, avec lequel les objets dans leur position actuelle peuvent laisser des traces sur la scène.

Les blocs de programme de l'objet **Etoile** modifient encore une fois les couleurs de façon cyclique et interrogent le contact de pâte à modeler sur la broche GPIO 18.

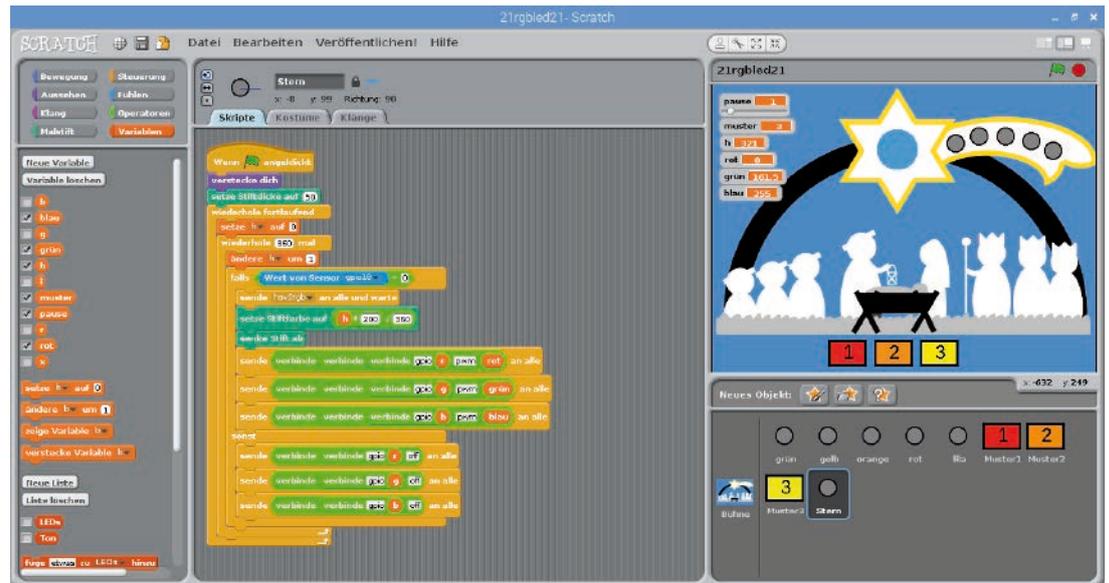
Tout d'abord, l'objet **Etoile** est caché, afin de ne pas recouvrir ses propres traces de peinture. Un objet qui est caché avec le bloc **cache toi** du bloc palette **Apparence** ne se voit plus sur la scène mais peut encore se déplacer, exécuter des blocs de programme et aussi des traces de peinture.

Un bloc **Règle l'épaisseur de stylo sur 50** règle l'épaisseur du stylo de peinture de sorte qu'il laisse un point fort.

Chaque fois que l'on touche le contact de pâte à modeler sur la broche GPIO 18, un point de couleur est peint en plus de la lumière du LED RVB. Celui-ci est situé exactement dans la position de l'objet **Etoile** masqué.

Pour ce faire, le nouveau stylo de couleur est calculé en premier. Scratch utilise 200 couleurs de stylos différents à travers le spectre de couleurs, semblable aux valeurs H dans le système HSV, mais pas dans la plage **0...360**, mais seulement dans l'intervalle **0...200**. Pour la conversion, la valeur H actuelle est multipliée par 200 et divisée par 360.

Puis le stylo est abaissé. Semblable à un traceur à plume, l'objet laisse des traces à chaque mouvement avec un stylo abaissé et aucune trace avec un stylo relevé. Même sans mouvement, l'abaissement du stylo crée à chaque fois un point dans la couleur actuelle du stylo.



Les blocs de programme pour l'étoile.

22ème Jour

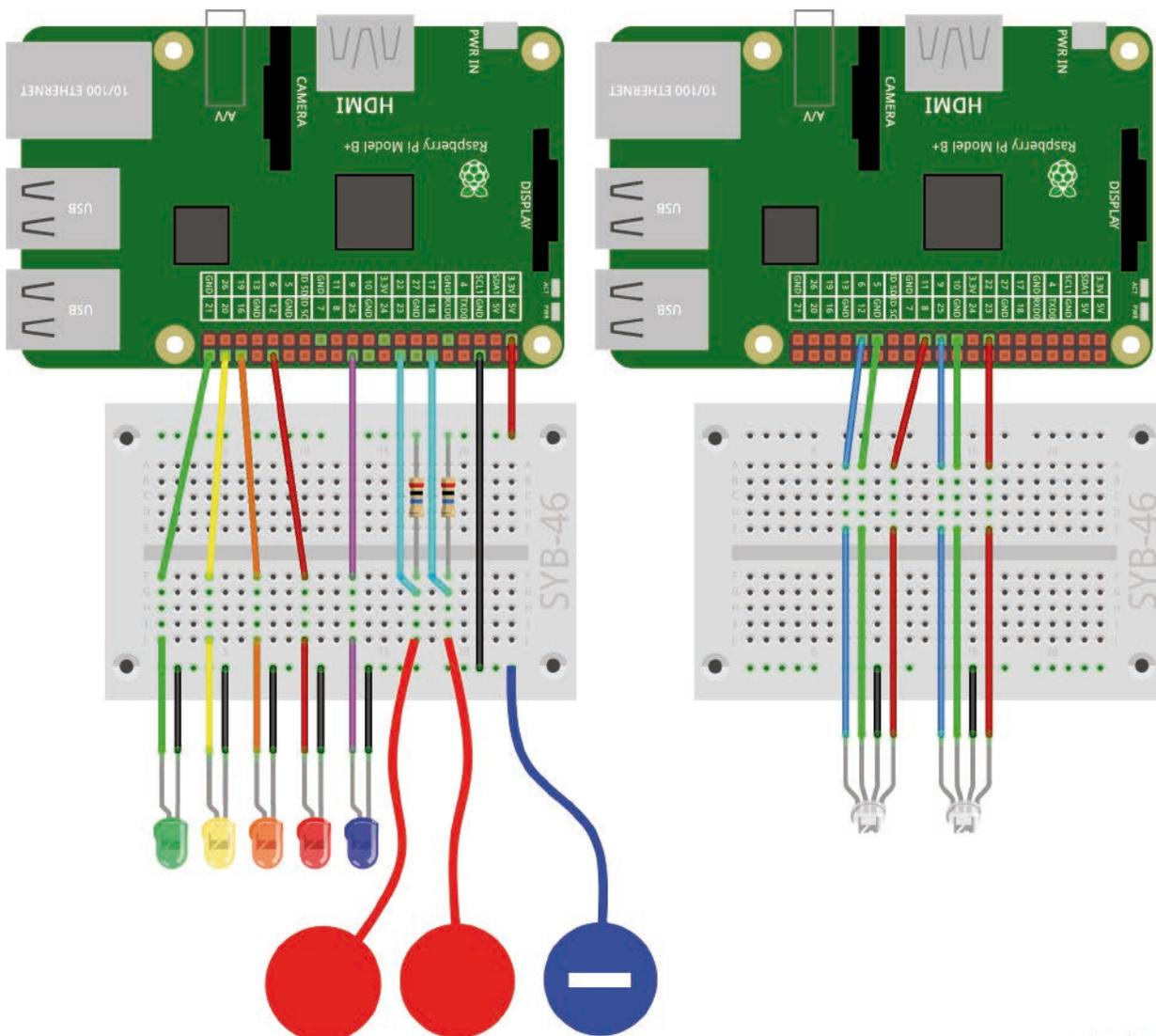
Aujourd'hui dans le calendrier de l'Avent

• 6 câbles de raccordement GPIO (courts)

Contrôler indépendamment les deux LEDs RVB

L'expérience du 22ème jour commande les deux LEDs RVB indépendamment l'un de l'autre avec les deux contacts de pâte à modeler. Cela nécessite trois broches GPIO supplémentaires. Les modèles de chenillards sont contrôlés exclusivement via les boutons. Dans ce programme, deux autres boutons ont été ajoutés. Le modèle de 4 fait clignoter brièvement tous les LED, le bouton **éteint** éteint tous les LEDs.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED orange avec résistance en série ; 1 LED rouge avec résistance en série ; 1 LED lilas avec résistance en série ; 2 LEDs RVB avec résistance en série ; 2 résistances de 20 MOhm ; 15 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 18 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 3 contacts en pâte à modeler



Deux contacts de pâte à modeler contrôlent le chenillard et deux LEDs RVB commutables indépendamment.

fritzing

Le programme

Le programme principal initialise sur la scène, comme déjà connu, une liste des cinq LEDs pour le chenillard, ainsi que les trois sorties PWM **r**, **g** et **b** pour le premier LED RVB. En plus s'ajoutent les trois autres sorties PWM **r2**, **g2** et **b2** pour le deuxième LED RVB.

Une boucle infinie crée à l'aide des variables **modèle** qui sont déterminées par les boutons, l'un des cinq modèles de clignotement sur les cinq LEDs. Les modèles 0 et 4 sont nouveaux :

Le modèle de 0 éteint tous les cinq LEDs sans attente supplémentaire.

Le modèle de 4 laisse tous les LEDs clignoter presque en même temps. Comme les broches GPIO sont allumées l'une après l'autre et qu'en même temps d'autres blocs Scratch sont exécutés, de courts délais entre le clignotement des LEDs individuels sont visibles.

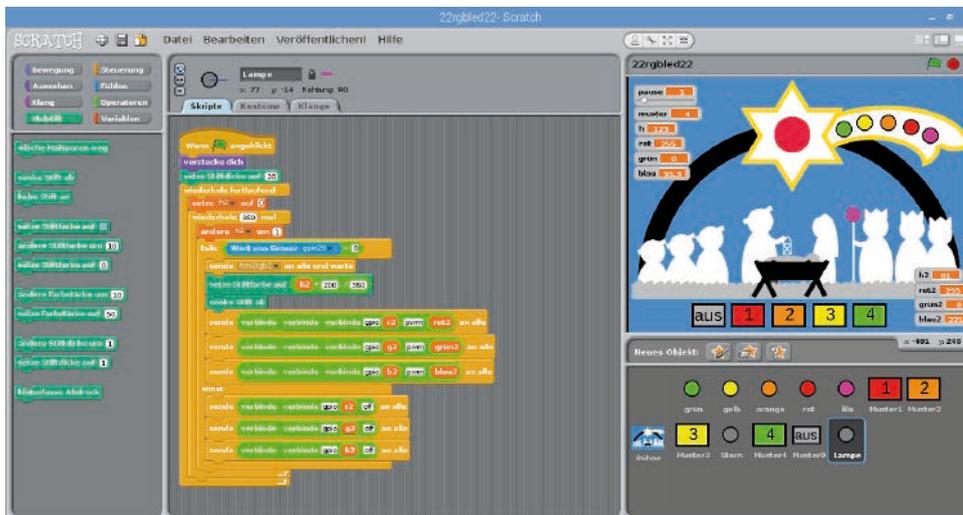
En outre, deux blocs de programme se trouvent sur la scène qui sont exécutés lorsque le signal **hsv2rgb** ou **hsv2rgb2** est reçu. La méthode est déjà connue. Le second bloc du programme pourrait être directement attribué à l'objet LED correspondant, mais il est facile à créer à travers la duplication et l'échange de variables et se trouve donc également sur la scène.



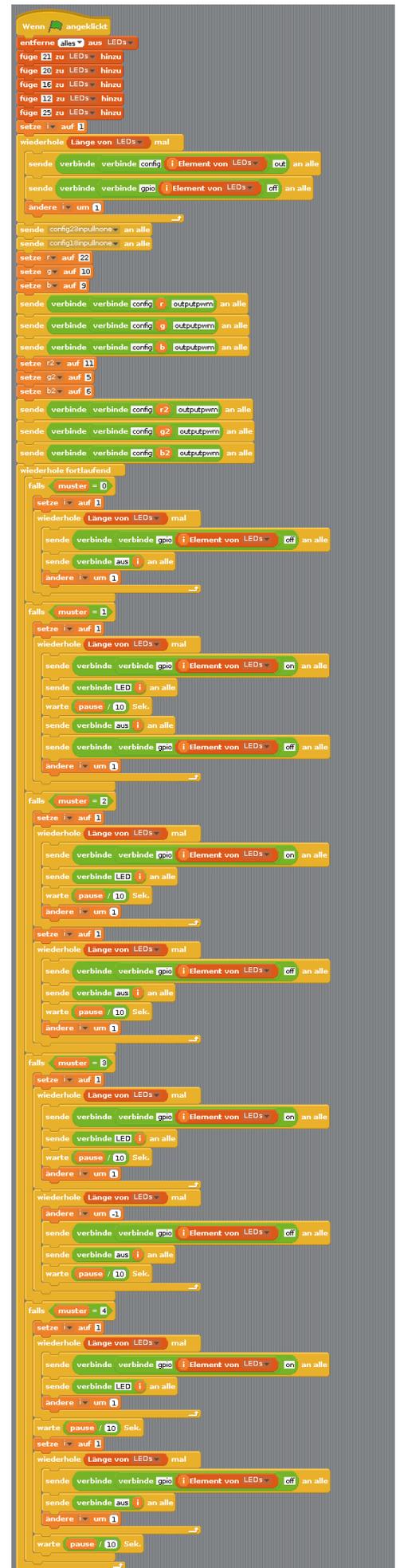
Ces blocs déterminent les couleurs RVB pour les deux LEDs RVB à partir de valeurs H.

Le nouvel objet **Lampe** sur la baguette du premier roi montre la couleur du deuxième LED RVB. Selon le principe bien connu, la valeur chromatique **h2** est augmentée graduellement de 0 à 360 étapes. Si l'on touche le contact de pâte à modeler sur la broche GPIO 23, le message **hsv2rgb2** est ainsi les valeurs de couleur RVB **rouge2**, **vert2** et **bleu2** sont calculées. Enfin, une couleur d'un stylo est déterminée, un point coloré est tracé et les broches GPIO **r2**, **g2** et **b2** sont placées sur les valeurs PWM.

Tant que le contact de pâte à modeler n'est pas touché, le LED RVB reste éteint. L'objet **Lampe** montre la dernière couleur définie.



Les blocs de programme pour le nouvel objet **Lampe**.



Le programme principal sur la scène.

23ème Jour

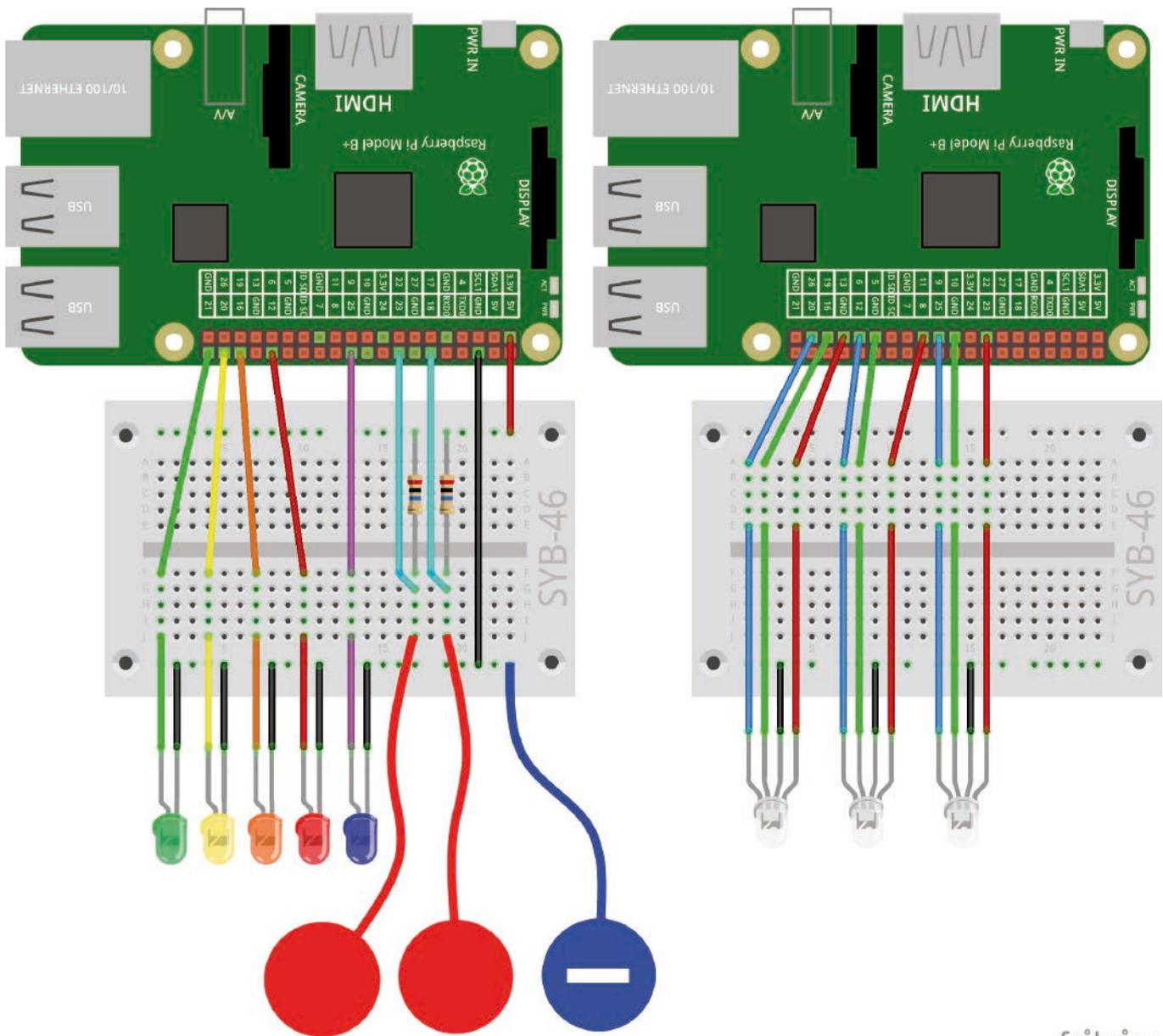
Aujourd'hui dans le calendrier de l'Avent

• 1 LED RVB avec résistance en série

Crèche de Noël lumineuse et colorée

Un troisième LED RVB dans la lanterne que tient Joseph sur la crèche de Noël, s'allume dans différentes couleurs primaires, selon laquelle des deux pâtes à modeler est touchée.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED orange avec résistance en série ; 1 LED rouge avec résistance en série ; 1 LED lilas avec résistance en série ; 3 LEDs RVB avec résistance en série ; 2 résistances de 20 MOhm ; 18 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 22 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 3 contacts en pâte à modeler

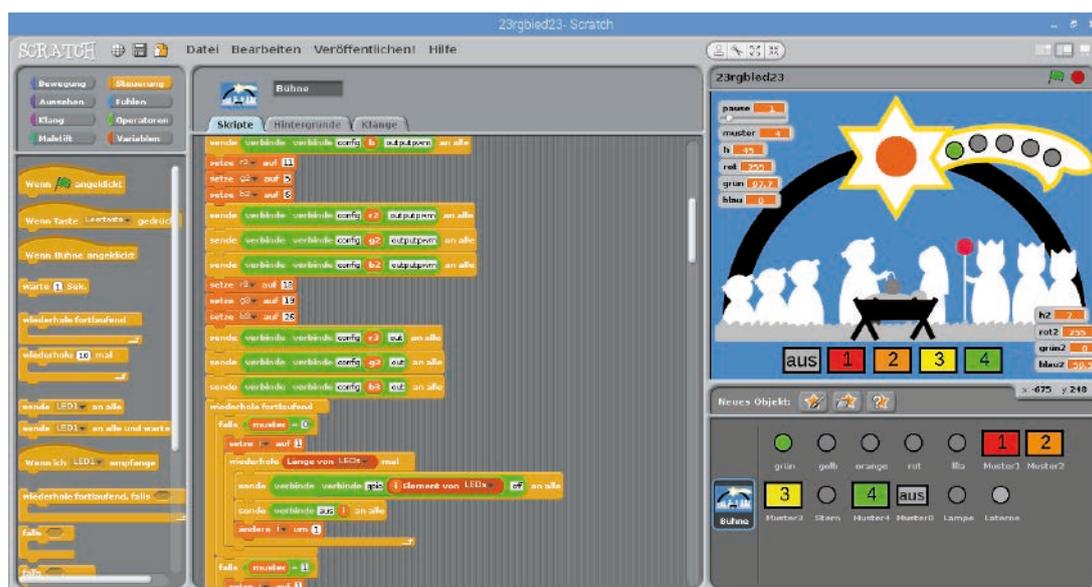


fritzing

Deux contacts de pâte à modeler contrôlent le chenillard et trois LEDs RVB commutables indépendamment.

Le programme

Encore une fois, le programme principal sur la scène initialise les broches GPIO pour les LEDs et les LED RGB. Ici seules les broches GPIO désignées avec **r3**, **g3** et **b3** sont nouvelles. Elles sont initialisées comme des sorties simples, non pas comme des sorties PWM.



Des sorties GPIO supplémentaires et un nouvel objet sur la scène.

Dès le départ, le nouvel objet **Lanterne** est caché et laisse juste une empreinte de stylo lors des changements de couleur. Ici, les deux contacts de pâte à modeler sont interrogés. Si l'on touche sur le capteur de la broche 23, la couleur du stylo est réglée sur rouge. Pour ce faire, on utilise le bloc **règle la couleur de stylo sur ...** avec le champ de couleur. Cliquez sur cette case de couleur, un symbole de pipette apparaît avec lequel vous pouvez prélever une couleur au choix de l'écran que le bloc déterminera plus tard comme la couleur du stylo. En outre, en touchant ce contact de pâte à modeler, la broche GPIO **r3** est allumée pour la couleur rouge du troisième LED RVB. Lorsque vous relâchez, cette broche s'éteint à nouveau.

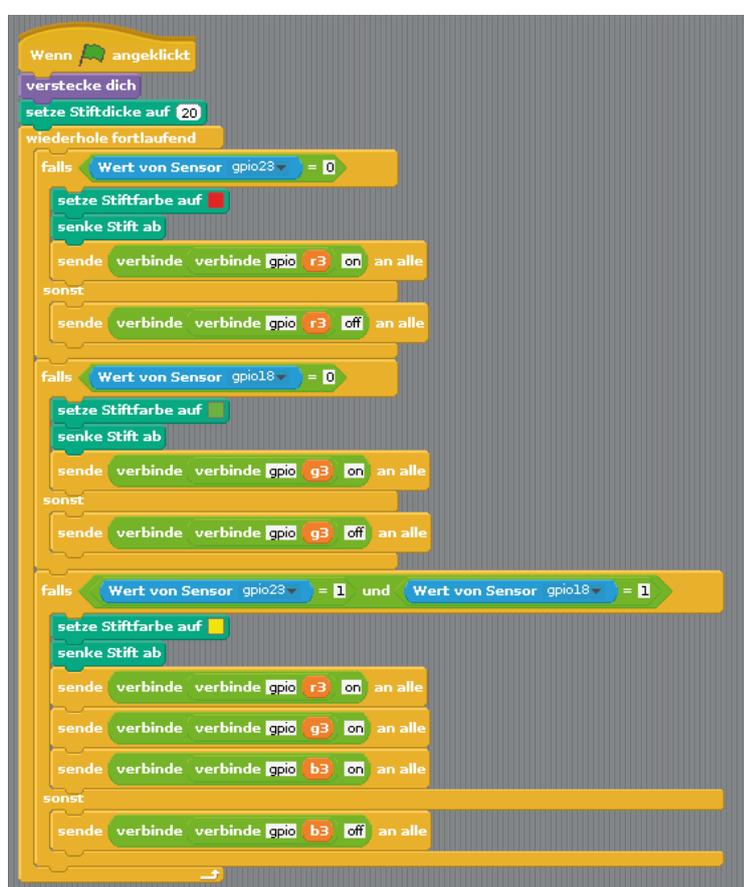
De la même manière, en touchant ce contact de pâte à modeler de la broche GPIO 18, la couleur de stylo est réglée sur vert et le composant de couleur verte du LED RVB s'allume. Si on touche simultanément les deux contacts de pâte à modeler, les broches **r3** et **g3** s'allument, le LED RVB s'allume dans une couleur jaune mixte. L'objet **Lanterne** montre la dernière couleur allumée, car ici aucune couleur n'est mélangée.

Si on ne touche aucun des deux contacts pâte à modeler - les deux valeurs de capteur sont sur **1** - le LED RVB s'allume d'un blanc brillant. Pour ce faire, les trois broches GPIO **r3**, **g3** et **b3** sont allumées. L'objet **Lanterne** montre du jaune, car le blanc est difficile à reconnaître sur un arrière-plan graphique. La condition n'est plus satisfaite, car au moins l'un des contacts de pâte à modeler est touché, la broche GPIO **b3** est éteinte afin que, si l'on touche l'un des deux contacts de pâte à modeler, les couleurs primaires rouge, vert ou un jaune mélangé s'affichent à nouveau - sans bleu mélangé.

Jusqu'à présent, le programme ne contient encore aucun moyen d'éteindre complètement les LEDs RGB. Un bloc de programme supplémentaire le fait en appuyant sur la barre d'espace.

Le bloc **Si la touche ... est appuyée** répond à une pression de touche sur le clavier. Dans la liste déroulante, sélectionnez la touche souhaitée.

Dans ce cas, un point gris à l'objet **Lanterne** est dessiné, et tous les trois composants de couleur du LED RVB sont éteints.



Les blocs de programme pour le nouvel objet **Lanterne**.



En appuyant sur la barre d'espace le LED RVB est éteint.

24ème Jour

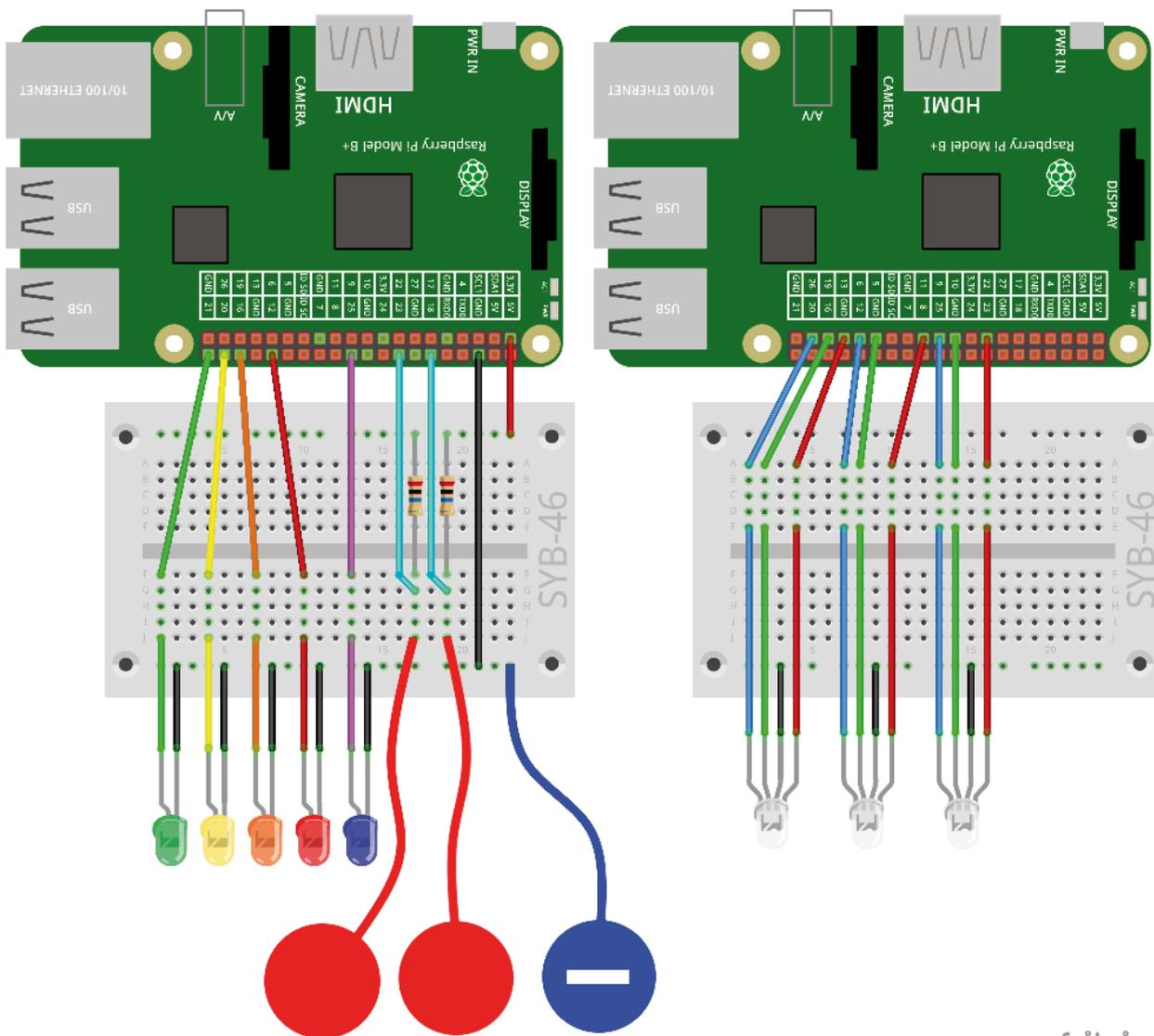
Aujourd'hui dans le calendrier de l'Avent

- code de 1 télécharger

Chants de Noël sur le Raspberry Pi

Le dernier jour, le programme est étendu à deux fonctions intéressantes. En cliquant sur l'un des cinq objets LED sur la scène, cinq chansons de Noël différentes sont jouées. Le LED dans la lanterne peut être réglé interactivement par une barre de couleur arc-en-ciel sur la scène.

Composants : 1 platine enfichable SYB-46 ; 1 LED vert avec résistance en série ; 1 LED jaune avec résistance en série ; 1 LED orange avec résistance en série ; 1 LED rouge avec résistance en série ; 1 LED lilas avec résistance en série ; 3 LEDs RVB avec résistance en série ; 2 résistances de 20 MOhm ; 18 câbles de connexion GPIO courts (Raspberry Pi - platine enfichable) ; 22 câbles de connexion GPIO longs (platine enfichable - LEDs) ; 3 contacts en pâte à modeler



Deux contacts de pâte à modeler contrôlent le chenillard et trois LEDs RVB commutables indépendamment.

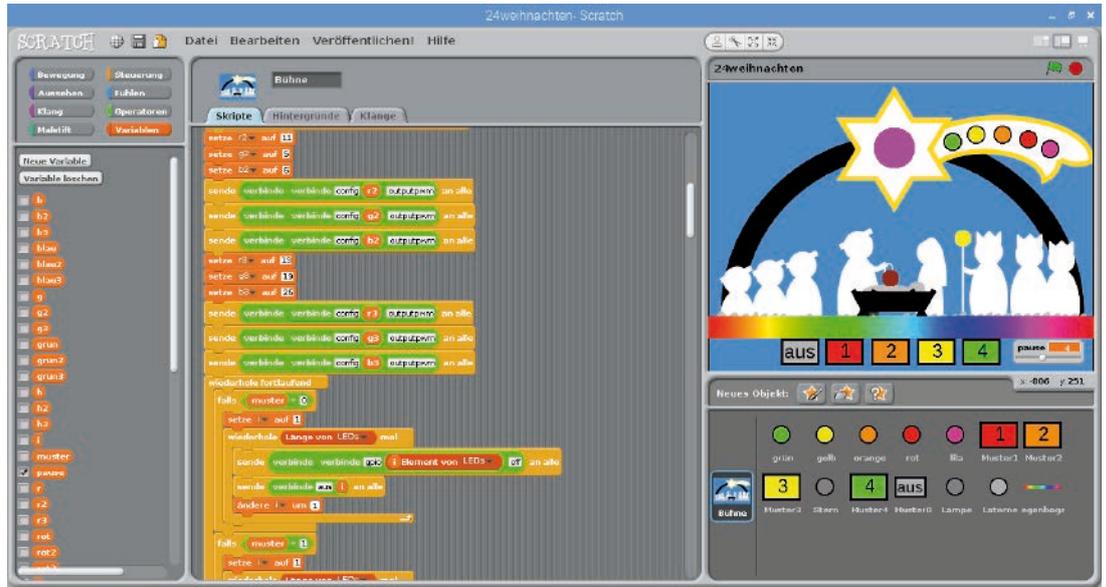
fritzing

Le programme

Les blocs de programme sur la scène sont connus pour la plupart. Le troisième LED RVB sur les broches GPIO **r3**, **g3** et **b3** sont définies cette fois comme les autres sorties PWM. Un nouveau bloc de programme sur la scène réagit au message **hsv2rgb3** et calcule les valeurs RVB pour le troisième LED RVB. Vous pouvez à nouveau facilement dupliquer ce bloc et remplacez les variables.

Pour l'objet **Arc-en-ciel** vous importez le graphique `regenbogen.png` par téléchargement. En cliquant avec la souris sur cet objet, le LED RVB doit s'allumer dans la couleur correspondante.

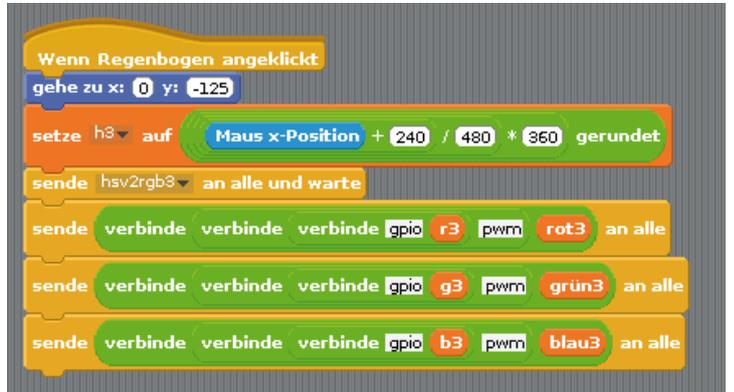
Comme il est facile de déplacer accidentellement l'objet lorsque vous cliquez dessus, il est immédiatement remis sur sa position de base après avoir cliqué. Ensuite, la valeur de H de la couleur souhaitée est déterminée à l'aide de la position x de la souris. La scène Scratch fait 480 unités de largeur et a x coordonnées de -240...240. Le bloc **Souris position x** lit les coordonnées x actuelles de la souris. De cette valeur qui peut se situer entre -240 et +240, on calcule une valeur **h3** entre 0 et 360. **Envoyer hsv2rgb3 à tous et attendre** commence à calculer les valeurs de couleur RVB correspondantes. Celles-ci sont ensuite représentées sous forme de signaux PWM sur le LED RVB.



De nouveaux blocs de programme sur la scène.

L'objet **Lanterne** répond aussi au message **hsv2rgb3** et dresse un point de couleur dans la couleur correspondante. La valeur **h3** dans la plage comprise entre 0 et 360 est convertie en une valeur de couleur dans la plage comprise entre 0 et 200. En appuyant sur la barre d'espace le LED RVB est éteint.

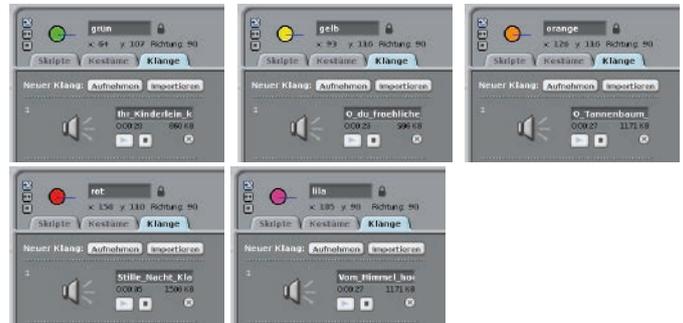
Scratch peut reproduire des données sur le MP3 Raspberry Pi qui ont été importées dans le programme. Pour cela, importez pour les objets **vert, jaune, orange, rouge et lilas** dans l'onglet **Sons** respectivement l'une des chansons de Noël que vous recevrez via le code de téléchargement dans la porte d'aujourd'hui du calendrier de l'Avent.



Les blocs pour l'objet **Arc-en-ciel**.

En cliquant sur un tel objet, la chanson de Noël correspondante devrait être reproduite.

Pour cela, dans chaque objet construisez, en plus des blocs pour le changement de costume, un autre bloc **Si ... cliqué**. Tout d'abord, il faut arrêter d'éventuelles chansons en cours afin que plusieurs chansons ne retentissent pas en même temps. Pour cela, il y a sur le bloc palette **Son** un bloc **Arrêter tous les sons**. Ensuite, un bloc **Jouer le son ... en entier** devrait reproduire la chanson importée pour cet objet.



Importer des chants de Noël pour les objets LED.

Aperçu

Nouvel aperçu de toutes les fonctions du programme :

- Les quatre boutons avec les chiffres permettent de faire clignoter les LEDs ainsi que les objets LED sur la queue de l'étoile selon divers modèles.
- Le régulateur **pause** règle la vitesse de clignotement.
- Le bouton **éteint** éteint toutes les LED et objets LED.
- Le premier contact de pâte à modeler détermine la couleur du LED RVB et du point de couleur sur l'étoile.
- Le deuxième contact de pâte à modeler détermine la couleur du LED RVB et du point de couleur sur la lampe du Roi.
- En cliquant sur la barre de multicolores vous déterminez la couleur du LED RVB et du point coloré sur la lanterne par Joseph dans la crèche.
- En appuyant sur la barre d'espace, le LED RVB s'éteint.
- Un clic sur l'un des cinq objets LED objets joue une chanson de Noël.

Joyeux Noël !



Des chants de Noël retentissent lorsqu'on clique sur les objets.