

Coder dans l'enseignement des sciences



<Mentions légales>

<Co-édité par>

Science on Stage Deutschland e. V.

Am Borsigturm 15

13507 Berlin, Allemagne

et

Atlantique Editions de L'Actualité scientifique

Espace Mendès France

1 place de la Cathédrale

CS 80964 - 86038 Poitiers cedex, France

<Coordinateur principal>

↳ **Dr Jörg Gutschank**, Leibniz Gymnasium
Dortmund International School, Dortmund, Allemagne
Président de Science on Stage Deutschland e. V.

<Coordinateurs>

↳ **Sebastian Funk**, Villa Wewersbusch, Velbert-Langenberg,
Allemagne, Comité Science on Stage Deutschland e. V.
↳ **Jean-Luc Richter**, Lycée Jean-Baptiste Schwilgué,
Sélestat, France, Vice-Président de Science on Stage France
↳ **Bernard Schriek** (ret.), Marien-Gymnasium, Werl,
Allemagne

<Coordination générale et révision>

↳ **Daniela Neumann**, chef de projet
Science on Stage Deutschland e. V.
↳ **Stefanie Schlunk**, directrice exécutive
Science on Stage Deutschland e. V.
↳ **Johanna Schulze**, directrice exécutive adjointe
Science on Stage Deutschland e. V.

Version française relue par Philippe Jeanjacquot et
Jean-Luc Richter de Science on Stage France

Pour faciliter la lecture, les codes Scratch, Snap! et
MakeCode ont été traduits. Les versions en ligne restent
en anglais pour permettre leur bonne exécution.

<Design>

WEBERSUPIRAN.berlin

<Illustration>

Rupert Tacke, Tricom Kommunikation und Verlag GmbH

<Crédits>

Les auteurs ont vérifié tous les aspects du droit d'auteur des
images et des textes utilisés dans cette publication au meilleur
de leur connaissance.

<Parcainé par>

SAP SE

<Soutiens à l'édition française>



<Imprimé par>

Raynaud Imprimeurs, Coulonges-sur-l'Autize (France)

<Veuillez commander auprès de>

contact@editionsatlantique.com

<ISBN PDF>

978-2-911320-75-0

<Prix>

1,00 €

Cette publication est destinée à un usage pédagogique, elle
est distribuée gratuitement sur demande auprès de l'éditeur.

Ce travail est sous licence Creative Commons Attribution-
ShareAlike 4.0 International License :

<https://creativecommons.org/licenses/by-sa/4.0/>.



Première édition publiée en 2019

© Science on Stage Deutschland e. V.

www.science-on-stage.eu

<Contenu>

Mots de bienvenue de la <04>
Commission européenne

Accueil de SAP SE <05>

Avant-propos <06>

Auteurs <07>



Les sciences fondamentales

en 1 et 0

Science friction <30>

Roulements sonores <36>

Moteur physique <42>

BMS-Boîte Magique pour les Sciences <48>

Environnement 4.0

<08> Coder H₂O

<14> Le monde de l'eau

<20> VPLS – Sauvegarde de la vie des
plantes en vacances

<26> Gant magique



Microcontrôler le monde

<54> CoALA-Code un petit animal

<58> Données liquides

<62> Capitaine télécommandé



<68> Comment coder

<72> Formation en informatique avec Snap!

<73> Meet and Code

<74> Autres ressources et projets

<75> Science on Stage Europe

<Mots de bienvenue de la Commission européenne>

Je suis très heureux d'apporter mon soutien à cette brochure concernant l'enseignement des sciences, de la technologie, de l'ingénierie et des mathématiques – les matières importantes des STEM – et, plus largement, au projet *Coder dans l'enseignement des sciences*.

Les disciplines STEM sont cruciales pour la construction d'une Europe compétitive et résiliente pour l'avenir. Le développement de la connaissance et de l'excellence dans ce domaine est un élément clé de notre ambition de construire un véritable espace européen de l'éducation d'ici 2025. Pourtant, nous sommes confrontés à un déficit de compétences. Nous devons faire davantage pour promouvoir les sujets STEM en Europe. En particulier, l'enseignement des matières STEM doit être un choix de carrière attrayant, et nous avons besoin de plus de modèles, en particulier de femmes.

Il ne s'agit pas seulement de croissance économique et de développement. L'enseignement des STEM doit être inclusif et permettre à des élèves de capacités et d'origines diverses de s'engager et de tirer le meilleur parti de leurs talents. Pour construire l'avenir, les jeunes ont besoin des compétences et des attitudes appropriées – et la maîtrise des STEM doit être au cœur de ces compétences.

Grâce à la science, nous pouvons apprendre tellement de choses : sur notre passé et notre présent, ce qui nous permet de façonner un avenir meilleur et mieux informé. Pour que les générations futures puissent vivre dans une société plus consciente du monde dans lequel nous vivons.

Je tiens à remercier Science on Stage Deutschland e.V., le Réseau européen des professeurs de sciences, qui est à la base de cette opération, et SAP SE, qui soutient les projets de codage.

Des enseignants de sept pays européens ont aidé à développer des exemples concrets et des conseils pratiques sur la façon d'acquérir des compétences en codage. Cela montre ce qui est possible lorsque nous nous réunissons, ce que nous avons en commun et ce que nous avons à apprendre les uns des autres.

Des projets tels que *Coder dans l'enseignement des sciences* ont un rôle clé à jouer dans la promotion des matières STEM dans les écoles, en aidant les jeunes Européens à acquérir les compétences essentielles dont ils ont besoin pour réussir dans la vie. Je félicite toutes les personnes impliquées et j'espère que votre exemple inspirera les autres.

Tibor Navracsics

Commissaire européen chargé de l'éducation,
de la culture, de la jeunesse et du sport



<Accueil de SAP SE>

Selon les prévisions du Forum économique mondial, environ 65% des enfants qui entrent aujourd'hui à l'école primaire exerceront des professions qui n'existent pas encore. Néanmoins, il est clair que ces enfants trouveront plus facilement leurs marques s'ils possèdent certaines compétences techniques ; la numérisation de notre économie ne peut tout simplement pas être arrêtée. C'est pourquoi, outre la lecture, l'écriture et les mathématiques, le travail avec les nouvelles technologies est devenu un sujet clé pour l'éducation.

SAP participe depuis des années à des initiatives européennes pour l'éducation et la formation des enfants et des jeunes, notamment dans les domaines de la robotique (*First Lego League*) et des technologies de programmation (*Meet and Code*). Notre objectif est d'initier les jeunes aux nouvelles technologies de manière ludique et de leur faciliter un meilleur départ dans leur future carrière.

De nos jours, de nombreux enseignants souhaitent également doter leurs élèves de connaissances techniques et numériques de base qu'ils pourront emporter avec eux au cours de

leur voyage. Cela ne signifie pas nécessairement que ces enseignants doivent être des experts ; au contraire, ils ont besoin de matériels de travail pratiques et éprouvés pour acquérir des compétences techniques dans différentes matières et à différents niveaux d'apprentissage.

De cette manière, nous facilitons également l'acquisition de compétences numériques dans la vie scolaire quotidienne et fournissons du matériel pédagogique pour les matières STEM, par exemple. Nous avons déjà lancé de nombreux projets pour soutenir les écoles avec Science on Stage Allemagne, y compris le matériel pédagogique que vous avez entre les mains, *Coder dans l'enseignement des sciences*.

Je me réjouis de la coopération avec Science on Stage Allemagne sur une démarche aussi orientée vers des projets et je suis convaincu que ce livret sera également un succès total. Je voudrais également remercier les enseignants qui y ont apporté leur soutien.

Michael Kleinemeier

Membre du conseil d'administration, SAP SE



<Avant-propos>

Cette brochure est spéciale à bien des égards. Permettez-moi de vous expliquer ce que je veux signifier par là.

Le codage est une compétence de base dans le monde moderne d'aujourd'hui et est particulièrement important en sciences, technologie, ingénierie et mathématiques (STEM). La programmation de machines devient une compétence de plus en plus recherchée et nécessaire dans tous les domaines de notre vie et ne peut plus être laissée aux seuls spécialistes en informatique. Par conséquent, le codage doit être enseigné non seulement dans les cours d'informatique, mais aussi dans toutes les autres matières. Toutefois, les programmes d'études européens pour les matières STEM ne répondent généralement pas à ce besoin. Dans le cadre du projet Science on Stage, le réseau européen des professeurs de sciences, *Coder dans l'enseignement des sciences*, a développé des concepts d'enseignement avec comme objectif de combler ce déficit de compétences. L'objectif global du réseau est de fournir une plate-forme aux enseignants européens de STEM pour échanger des idées sur les meilleures pratiques. Science on Stage touche 100 000 éducateurs dans plus de 30 pays membres.

Certains des meilleurs enseignants d'Europe ont apporté leurs idées à cette brochure. 23 enseignants de sept pays se sont rencontrés en personne pour échanger leurs idées sur le codage dans l'enseignement des sciences au cours de ce projet de 18 mois. Tous les enseignants participants ont investi beaucoup de temps et d'efforts dans ce processus.

Un professeur de sciences et un professeur d'informatique de chacun des sept pays se sont associés pour échanger des concepts d'enseignement avec une équipe d'au moins un autre pays. Ils ont discuté et évalué ces concepts dans leurs propres classes dans les deux pays pour s'assurer que le matériel que vous tenez dans vos mains sera utile dans les leçons réelles, et qu'il a été bien testé par nos experts, les enseignants eux-mêmes.

L'accent est mis sur la programmation de petits appareils électroniques comme Arduino, Calliope mini, ou les ordinateurs Raspberry Pi. Ce sont des appareils peu coûteux qui peuvent résoudre des tâches en profondeur, ce qui les rend idéaux pour les écoles.

Les participants ont développé 11 unités d'enseignement dans les domaines des « Sciences fondamentales en 1 et 0 », « Microcontrôler du monde » et « Environnement 4.0 ». Ce sont d'excellents exemples de ce que vous pourriez faire en ce qui concerne les divers programmes d'études en biologie, en chimie et en physique.

Le projet *Coder dans l'enseignement des STEM* n'a été possible que grâce aux efforts de nos participants enthousiastes, qui ont fait tout ce travail pendant leur temps libre et en plus de leur travail régulier d'enseignant. Merci à tous pour cette publication inspirante. Un grand merci également aux autres coordinateurs, Jean-Luc Richter, Bernd Schriek et Sebastian Funk, qui ont tous fait un excellent travail en condensant les pensées et les idées hétérogènes d'enseignants d'horizons culturels et scientifiques différents en un ouvrage homogène. Je voudrais également remercier SAP SE et Gabriele Hartmann et Jens Mönig en particulier. Une telle publication n'aurait pas été possible sans le soutien continu de ces partenaires professionnels.

Coder dans l'enseignement des sciences est une brochure spéciale qui a été développée et testée par des enseignants pour leurs collègues en Europe et au-delà. Laissez-la vous inspirer pour commencer vos propres projets dans votre classe au plus vite !

Dr Jörg Gutschank

Président de Science on Stage Deutschland e. V.



<Auteurs>

<Nom de famille>	<Nom>	<Pays>	<Section>
Abad Nebot	Immaculada	Espagne	Microcontrôler le monde
Botelho	Lúcio	Portugal	Environnement 4.0
Compte Jové	Pere	Espagne	Microcontrôler le monde
Ferretes	Liliana	Portugal	Environnement 4.0
Funk	Sebastian	Allemagne	Coordinateur
Georgoulakis	Georgios	Grèce	Les sciences fondamentales en 1 et 0
Giurgea	Mihaela Irina	Roumanie	Les sciences fondamentales en 1 et 0
Gutschank	Jörg	Allemagne	Coordonnateur Les sciences fondamentales en 1 et 0
Hančl	Mirek	Allemagne	Microcontrôler le monde
Ivarra	Luc	Belgique	Les sciences fondamentales en 1 et 0
Karagiorgou	Eleftheria	Grèce	Microcontrôler le monde
Lőkös	Annamária	Roumanie	Environnement 4.0
Meier	Andreas	Allemagne	Environnement 4.0
Mestvirishvili	Ilia	Georgia	Les sciences fondamentales en 1 et 0
Nicolini	Marco	Belgique / Italie	Les sciences fondamentales en 1 et 0
Padin	Beatriz	Espagne	Environnement 4.0
Poncela	Elena	Espagne	Environnement 4.0
Rațiu	Camelia Ioana	Roumanie	Les sciences fondamentales en 1 et 0
Reis	Jorge	Portugal	Environnement 4.0
Richter	Jean-Luc	France	Coordinateur Environnement 4.0
Schriek	Bernard	Allemagne	Coordinateur Microcontrôler le monde
Shapakidze	David	Georgia	Les sciences fondamentales en 1 et 0
Toma	Corina Lavinia	Roumanie	Les sciences fondamentales en 1 et 0
Tsiliki	Sevasti	Grèce	Microcontrôler le monde
Tsoutsoudakis	Astrinos	Grèce	Les sciences fondamentales en 1 et 0
van der Byl	Sonja	Allemagne	Environnement 4.0
Winckler	Julia	Allemagne	Microcontrôler le monde

Un grand merci à Gabriele Hartmann et Jens Mönig de SAP SE pour leur soutien !

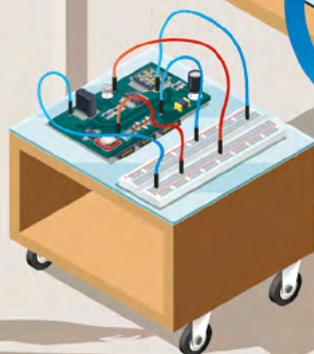
Nous remercions Holger Bach et Paul Nugent pour leurs précieux conseils lors de l'édition !



Coder H₂O

<Auteur> Beatriz Padin

<Auteur> Elena Poncela



<Info>

<Mots-clés> eau, capteurs, efficacité, acquisition de données, environnement, évaporation, condensation, solutions, mélanges, conception, chaleur et température, conducteurs et isolateurs thermiques, énergie solaire, rayonnement infrarouge (IR), réflexion

<Disciplines> physique, sciences de l'environnement, chimie, informatique, mathématiques

<Niveau d'âge des élèves> 13–15 ans

<Matériel> Arduino^[1], Calliope mini^[2], Raspberry Pi^[3]

<Langage> Arduino^[4], Python^[5], programmation par blocs

<Niveau de programmation> moyen

- ↳ Effets de la chaleur : changements d'état
- ↳ Différence entre les sources d'énergie renouvelables (énergie solaire) et non renouvelables
- ↳ Le rayonnement infrarouge et son rôle dans le transport de la chaleur du soleil
- ↳ Le rayonnement et sa réflexion sur certaines surfaces
- ↳ Procédés de séparation de mélanges
- ↳ Solutions : quelles sont-elles, concentration (g/L et pourcentage massique, etc.)

Selon les connaissances préalables des élèves, ils découvriront certains de ces concepts par eux-mêmes, tandis que d'autres devront être expliqués par l'enseignant et validés expérimentalement par les élèves.

<Résumé>

Les élèves conçoivent, construisent et testent un distillateur solaire pour purifier l'eau. Ils programmeront des capteurs pour mesurer l'efficacité de leurs distillateurs solaires.

<Introduction des concepts>

Nous couvrirons les concepts de physique suivants :

- ↳ Changements d'état (notamment évaporation et condensation) et leurs principales caractéristiques
- ↳ Facteurs qui affectent le processus d'évaporation (température, surface, etc.)

L'objectif de notre projet est de concevoir et construire le distillateur solaire le plus efficace pour purifier l'eau. Seule l'énergie solaire peut être utilisée. Tout d'abord, l'efficacité du distillateur solaire sera déterminée en calculant le pourcentage en volume de l'eau purifiée qui est obtenu. Ensuite, les élèves programmeront différents capteurs pour analyser l'efficacité de leurs conceptions.

La tâche principale de cette activité est de programmer les capteurs, de sorte que les enseignants auront besoin de compétences en programmation et de quelques connaissances de base en informatique pour ce faire.



© Distillateur solaire

Ils devront savoir comment construire le circuit nécessaire pour connecter les capteurs à la carte microcontrôleur. Selon leur niveau de compétence, ils peuvent utiliser la programmation par blocs (Calliope mini^[2], Snap4Arduino^[6], etc.) ou par texte (Arduino^[4], Python^[5], etc.) pour programmer les capteurs.

<Ce que font les élèves/enseignants>

Cette unité se compose de trois parties : la conception et la construction d'un distillateur solaire, le codage des capteurs et le test du distillateur solaire.

<Première partie : Concevoir et réaliser un distillateur solaire>

Le projet sera présenté aux élèves et ils testeront leur premier exemple de distillateur solaire en classe en mesurant la quantité d'eau propre recueillie et en calculant l'efficacité de ce modèle. Ce faisant, ils passeront en revue des concepts de physique tels que les changements d'état, les solutions et l'énergie solaire.

Ensuite, les élèves seront encouragés à améliorer ce modèle initial. Pour ce faire, ils devront travailler en groupe (2–3 élèves par groupe). Cette phase de l'unité peut être divisée en différentes tâches :

1. Les élèves chercheront de l'information sur les distillateurs solaires, leur fonctionnement, les différents modèles déjà utilisés, etc. Au cours du processus de recherche, on leur demettra de réfléchir aux questions suivantes :
 - a. Processus d'évaporation : quels sont les principaux facteurs qui influencent ce processus ? Pensez à la surface où vous allez placer l'eau sale. Est-il préférable d'avoir une surface large ou étroite, une profondeur plus grete ou plus petite ? La couleur du contenant est-elle importante ?
 - b. Procédé de condensation : qu'est-ce qui est nécessaire pour produire de la condensation ? Avez-vous besoin de concevoir une grete ou une petite surface pour la condensation de l'eau ? Comment allez-vous faire en sorte que l'eau propre soit acheminée jusqu'au point où vous souhaitez la recueillir ?
 - c. Processus de rayonnement : comment pouvez-vous maximiser le rayonnement IR qui frappe votre distillateur solaire ? Comment pouvez-vous atteindre la température maximale possible dans votre distillateur solaire ? Pensez à utiliser une surface recouverte d'une feuille d'aluminium pour réfléchir la lumière du soleil dans le distillateur solaire.
2. Les groupes créeront leurs plans et les expliqueront à l'enseignant.
3. Une fois que l'enseignant aura approuvé la conception, on demettra aux élèves de trouver le matériel approprié

(à l'école, à la maison, commeter en ligne, etc.) et de construire leur distillateur.

4. Les étudiants détermineront l'efficacité de leurs distillateurs solaires sans capteurs. À l'aide d'un cylindre gradué, ils mesureront non seulement le volume d'eau sale, mais aussi le volume d'eau propre recueilli et appliqueront l'équation mathématique suivante :

$$\text{Efficacité} = \frac{\text{volume d'eau collecté}}{\text{volume d'eau sale}}$$

Un guide étape par étape comprenant des tâches et des questions pour les élèves est fourni pour les deuxième et troisième parties de cette activité.

L'objectif est de donner aux élèves différents choix en ce qui concerne les capteurs, les langages de programmation et le matériel, mais cela dépend aussi de chaque école/classe (matériel disponible, connaissance du langage de programmation, etc.).

<Deuxième partie : Programmer les capteurs>

Seuls les meilleurs distillateurs solaires seront testés à l'aide de capteurs. Dans cette partie, vous devrez :

1. Sélectionnez la carte microcontrôleur, le langage de programmation et les capteurs avec lesquels vous allez travailler. Répondez aux questions suivantes pour vous aider à prendre la meilleure décision :
 - a. Allez-vous utiliser la programmation par blocs ou la programmation par texte ? Si vous choisissez la programmation par blocs, pensez à utiliser un Calliope mini^[2] ou bien à programmer un Raspberry Pi^[3] avec Scratch^[8]. Si vous optez pour la programmation texte, vous pouvez utiliser un Arduino^[4], ou un code en Python^[5] pour Raspberry Pi^[9].
 - a. Allez-vous utiliser des capteurs numériques ou analogiques ? Si vous optez pour cette dernière solution, Arduino pourrait être le meilleur choix.
2. Choisissez les capteurs que vous souhaitez utiliser. Pour simplifier les choses, veuillez ne pas sélectionner plus de deux capteurs. Quelques exemples sont la température, l'humidité, la pluie et le rayonnement infrarouge. Avant de prendre votre décision, considérez les spécifications de chaque capteur. Le signal de sortie est-il analogique ou numérique avec seulement deux valeurs possibles (vrai/faux) ? Comment le signal de sortie se rapporte-t-il à la valeur du paramètre que vous mesurez ? Sont-ils directement proportionnels ? La sortie augmente-t-elle à mesure que le paramètre diminue ?



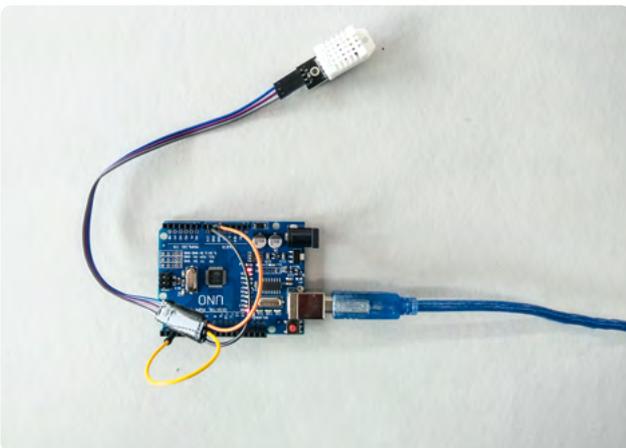
© Capteur de flamme

3. Construisez le circuit pour connecter votre capteur à la carte microcontrôle. Utilisez les ressources supplémentaires fournies^[7] ou cherchez des exemples sur Internet.
4. Codez les capteurs. Vous devez suivre ces étapes :
 - a. Notez ce que vous voulez que votre programme fasse, en tenant compte des caractéristiques des capteurs que vous avez sélectionnés. Voulez-vous que votre programme n'affiche que la valeur de la greteur mesurée ? Voulez-vous également afficher les valeurs maximum et minimum ? Votre capteur affiche-t-il la valeur réelle du paramètre ou avez-vous besoin de faire des calculs ?
 - b. Rédigez votre programme. N'oubliez pas d'écrire des commentaires sur votre code. Vous pouvez utiliser les ressources supplémentaires fournies par votre enseignant comme guide.^[7]

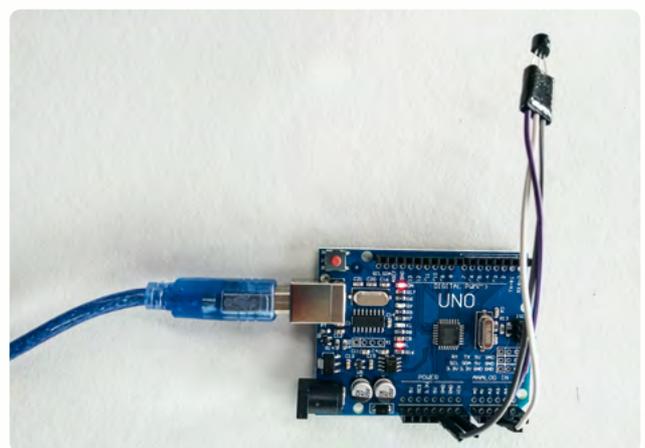
5. Testez votre code. Le programme fonctionne-t-il comme prévu ?

Exemples :

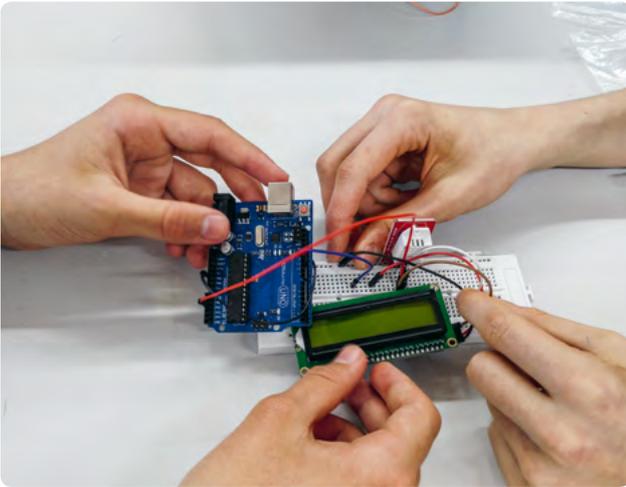
- ↳ Vous pouvez mesurer la quantité de rayonnement infrarouge qui atteint le distillateur solaire avec un capteur de flamme et un Arduino UNO^[4]. Si vous utilisez une surface recouverte d'une feuille d'aluminium, utilisez ce capteur pour vérifier si le rayonnement est réfléchi dans le distillateur solaire.
- ↳ Vous pouvez enregistrer la température maximale et l'humidité relative atteinte à l'intérieur du distillateur solaire avec le capteur d'humidité et de température DHT11 ou DHT22 et votre Arduino UNO.
- ↳ Vous pouvez écrire un programme Python 3 pour déterminer le temps qu'il faut pour que les premières gouttes d'eau se condensent avec le capteur de pluie FR-04 et Raspberry Pi^[3]. Vous pouvez également utili-



© Arduino avec capteur d'humidité



© Arduino avec sonde de température



© Arduino avec écran LCD et capteur d'humidité

ser ce capteur avec un Calliope mini et le programmer avec Snap!^[40] (langage de programmation par blocs).

<Troisième partie : Tester les distillateurs solaires à l'aide de capteurs>

Votre équipe doit utiliser les capteurs programmés pour tester et comparer votre conception avec une autre. Les distillateurs solaires doivent fonctionner dans les mêmes conditions et utiliser le même type de capteurs pour que cela se produise.

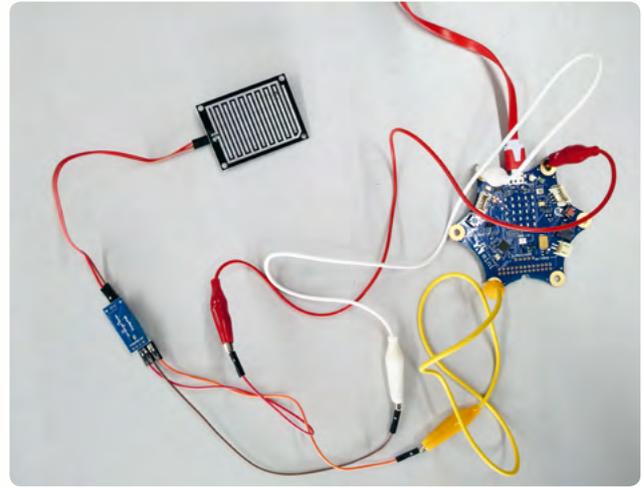
Vous analyserez les raisons pour lesquelles les efficacités sont différentes, en notant des faits clés comme les facteurs qui affectent l'évaporation, etc.

Comment améliorer l'efficacité du distillateur solaire :

- ↳ Testez-le un jour ensoleillé pendant la période de midi.
- ↳ Laissez suffisamment de temps à votre équipement pour atteindre la température la plus élevée possible.
- ↳ Assurez-vous que le distillateur solaire est étanche à l'air pour éviter toute perte de vapeur d'eau.
- ↳ Enlevez l'eau propre de façon continue pour éviter qu'elle ne s'évapore.
- ↳ Colorez l'eau sale pour vous assurer que le système solaire fonctionne toujours correctement.
- ↳ Choisissez un récipient large et noir pour l'eau sale.
- ↳ Utilisez un parapluie recouvert d'une feuille d'aluminium pour réfléchir la lumière du soleil dans le distillateur solaire.

<Résultats>

La première partie de l'activité (construction des distillateurs solaires) a donné lieu à des conceptions dont l'efficacité variait considérablement. Lors d'essais effectués par temps ensoleillé en mai/juin en Espagne, le meilleur design a permis de purifier 95 % de l'eau sale en 24 heures. Un taux d'efficacité de 54 % a également été obtenu en 4 heures. Néanmoins, certains dis-



© Capteur de pluie avec un Calliope mini

tillateurs n'ont pas collecté d'eau propre du tout en raison de diverses lacunes dans leur conception.

En appliquant les capteurs, les élèves ont obtenu les résultats suivants :

- ↳ La température maximale atteinte à l'intérieur des distillateurs solaires après une heure par une journée ensoleillée était de 65 °C.
- ↳ L'influence de la couleur du récipient sur l'eau sale a été analysée. En comparant un contenant blanc et un contenant noir laissé au soleil pendant quelques minutes, on a constaté que la température de l'eau dans le contenant noir était presque 5 °C plus élevés que dans le contenant blanc.
- ↳ La relation entre la température de l'eau et le taux d'évaporation a été étudiée à l'aide du capteur d'humidité. L'humidité relative obtenue à l'intérieur d'un distillateur solaire contenant de l'eau à température ambiante a été mesurée à 55 %. Lorsque l'eau était chauffée à 45 °C, l'humidité relative augmentait à 98 % en quelques secondes seulement.
- ↳ La quantité de rayonnement réfléchi par une surface métallique dans un distillateur solaire a été mesurée avec le capteur de flamme. Un vieux parapluie recouvert d'une feuille d'aluminium était très efficace pour réfléchir le rayonnement IR.

<Conclusion>

Les élèves utiliseront leur créativité et leur capacité de raisonnement d'ordre supérieur (HOTS) pour concevoir le distillateur solaire le plus efficace. Cette activité leur donnera également l'occasion de développer leur esprit critique et leurs compétences en résolution de problèmes, ce qui est très utile, car les élèves peuvent les utiliser au quotidien. Ils apprendront certains faits physiques clés (autoapprentissage) en observant, expérimentant, testant et analysant leurs résultats, au lieu de simplement lire des articles dans leurs manuels de physique.

Ils développeront également des capacités de raisonnement informatique tout en codant leurs capteurs. Ils feront de l'informatique physique (c.-à-d. que leurs codes interagiront avec le monde physique). Tout au long de l'activité, ils suivront les différentes étapes de la méthode scientifique décrite ci-dessus pour développer la meilleure conception de distillateur solaire et ensuite la construire avec des matériaux appropriés.

Il est important que chaque élève soit impliqué dans les tâches du projet. Certaines tâches peuvent être effectuées individuellement (recherche d'informations, collecte d'idées initiales sur la conception...) pour s'assurer d'aller au bout de la démarche.

Cependant, le principal obstacle est que certains élèves peuvent ne pas avoir les compétences de programmation appropriées (c'est pourquoi nous avons inclus des alternatives de programmation par blocs), des connaissances suffisantes pour construire les circuits (les ressources supplémentaires^[7] sont très utiles à cet égard) ou peuvent ne pas comprendre comment fonctionnent les capteurs. De plus, il se peut que le matériel ne soit pas disponible dans toutes les écoles et qu'il faille donc l'acheter.

Activités de vulgarisation :

- ↳ Les données recueillies par les capteurs pourraient être stockées sur une carte SD pour une analyse plus poussée.
- ↳ Un écran LCD peut être utilisé pour visualiser les mesures.
- ↳ Internet des objets (IoT) : les données collectées pourraient être envoyées par Internet en temps réel afin d'être accessibles au public.
- ↳ Des capteurs supplémentaires pourraient être utilisés, tels que CO₂ ou d'autres capteurs de gaz à effet de serre, un capteur de conductivité pour vérifier si l'eau propre est encore salée, un capteur de pH pour mesurer le pH de l'eau sale et propre, etc.
- ↳ La salinité pourrait être mesurée avec des échantillons d'eau de mer.
- ↳ Une méthode de désinfection de l'eau recueillie pourrait être ajoutée.

Les élèves pourraient utiliser les distillateurs solaires pour étudier l'effet de serre, la photosynthèse, la respiration cellulaire, les gaz idéaux, etc. De nombreux capteurs pourraient être utilisés dans d'autres projets pour mesurer des paramètres physiques ou chimiques, par exemple pour surveiller la pollution atmosphérique ou la qualité de l'eau.

<Activité de coopération>

Les distillateurs solaires sont-ils plus efficaces dans les pays où le temps est ensoleillé ? Les élèves de différentes écoles européennes pourraient partager leurs résultats à l'aide d'une carte en ligne à des fins de localisation. La salinité des différents endroits pourrait être comparée en prélevant des échantillons d'eau de mer.

<Références>

- [1] www.arduino.cc/
- [2] <https://calliope.cc/en>
- [3] www.raspberrypi.org
- [4] www.arduino.cc/reference/en/
- [5] www.python.org/
- [6] <http://snap4arduino.rocks/>
- [7] Tous les autres documents sont disponibles à l'adresse www.science-on-stage.de/coding-materials.
- [8] <https://scratch.mit.edu/>
- [9] www.raspberrypi.org/documentation/usage/python/
- [10] <https://snap.berkeley.edu/>



Le monde de l'eau

<Auteur> Lúcio Botelho

<Auteur> Liliana Fernetes

<Auteur> Jorge Reis

<Info>

<Mots-clés> eau, traitement d'images, acquisition de données, microclimat, robots

<Disciplines> mathématiques, biologie, sciences sociales, robotique, art plastique

<Niveau d'âge des élèves> 6–10 ans, 11–15 ans et 16–18 ans

<Matériel> **<niveau facile>** Calliope mini^[4], LEGO We Do 2.0^[2], petits robots d'apprentissage^[3], WeeeMake^[4]

<moyen level> LEGO EV3^[2] avec capteurs ultrasoniques et de couleur LEGO, ou Anprino^[5] avec Arduino^[6] et capteurs ultrasoniques^[7] et de couleur appropriés^[8]

<niveau avancé> ordinateur avec accès Internet

<Langage> Snap!^[9], Scratch^[10], WeeeCode^[4], Open Roberta^[11], LEGO Blocks^[2]

<Niveau de programmation> facile, moyen, avancé

<Résumé>

Cette unité a été conçue pour être transdisciplinaire par nature et pour faciliter le travail collaboratif entre élèves de différents niveaux, du primaire au secondaire. Chaque partie peut être enseignée séparément à son propre niveau. Dans l'unité d'enseignement « le monde de l'eau », les élèves découvriront tout sur le thème de l'eau en commençant par une approche théorique, en passant par le codage dans Scratch^[10], jusqu'à la programmation de robots et d'une maison écologique.

<Introduction des concepts>

Ce projet concerne l'eau, son rôle dans notre vie et notre rôle dans sa préservation. Divisé en trois niveaux (facile pour les élèves du primaire, moyen pour les élèves du collège et avancé pour les élèves du lycée), ce projet peut être adapté pour un travail en collaboration à différents niveaux scolaires et dans des activités transversales.

<Ce que font les élèves/enseignants>**<Niveau facile : D'où vient l'eau?>**

Les élèves sont mis au défi d'étudier d'où vient l'eau. L'enseignant pose des questions pour stimuler l'intérêt des élèves et alors ... l'aventure commence ! Les élèves font des recherches, apprennent et partagent ensuite leurs résultats avec leurs camarades de classe. En même temps, les élèves commencent à développer leurs capacités de raisonnement informatique avec des défis faciles qui leur apprennent à programmer des robots simples.

Après avoir terminé leur recherche, les élèves commencent à travailler en petits groupes et à construire des projets pour débu-

tants en utilisant les modes de démonstration de l'application WeDo 2.0^[12] avec un accent particulier sur les tâches liées à l'eau.

Ensuite, les élèves doivent construire une solution écologique, en utilisant toutes les briques ou tous les ensembles qu'ils veulent, dans laquelle ils présentent une approche innovante pour économiser l'eau.

Dans l'exemple ci-dessous, les élèves ont construit une maison écologique^[13] et l'ont combinée avec des briques supplémentaires et l'ensemble WeDo 2.0. Ensuite, ils ont ajouté un collecteur d'eau de pluie relié à un filtre (codé avec l'application LEGO) qui dirige l'eau vers la ferme pour que les animaux puissent boire de l'eau douce (© 1).



© 1 : Une maison écologique

En même temps, les élèves qui travaillent toujours en petits groupes, commencent à planifier et à concevoir de nouveaux tapis liés à l'eau pour de petits robots d'apprentissage qui peuvent être programmés sans ordinateur. En présentant leurs tapis aux autres élèves, ils les motivent à coder et à apprendre sur l'eau en même temps. Les élèves peuvent utiliser divers robots d'apprentissage peu coûteux pour effectuer la tâche.^[13]

Des instructions complètes sur la façon d'imprimer les tapis sont disponibles en ligne.^[14]



© 2 : Éléves dessinant des tapis

Vous trouverez également un lien vers un plan d'unité étape par étape dans la documentation en ligne supplémentaire. [14]

<Niveau moyen : Construction d'un robot de nettoyage de barrage>

Le projet de nettoyage du barrage consiste en un robot qui traverse le réservoir d'eau créé par un barrage et détecte les déchets solides.

Ce projet a deux versions, utilisant deux robots différents. La version LEGO [3] utilise le kit éducatif LEGO EV3 ; le robot Anprino [5], qui est imprimé à l'aide d'une imprimante 3D et ensuite assemblé, est la base de travail de la version Arduino [4]. Le microcontrôleur Arduino [6] et sa gamme d'accessoires sont fixés sur l'Anprino.



3 : La version LEGO

4 : La version Anprino

Commencez par construire le modèle de réservoir d'eau en utilisant du papier ou du carton. Il devrait mesurer environ 2 m x 1 m et être peint en bleu pour simuler l'eau. Construisez les rives à l'aide de carton solide pour limiter l'espace de déplacement des robots et simulez les déchets avec des morceaux de carton noir.



5 : Modèle du réservoir du barrage

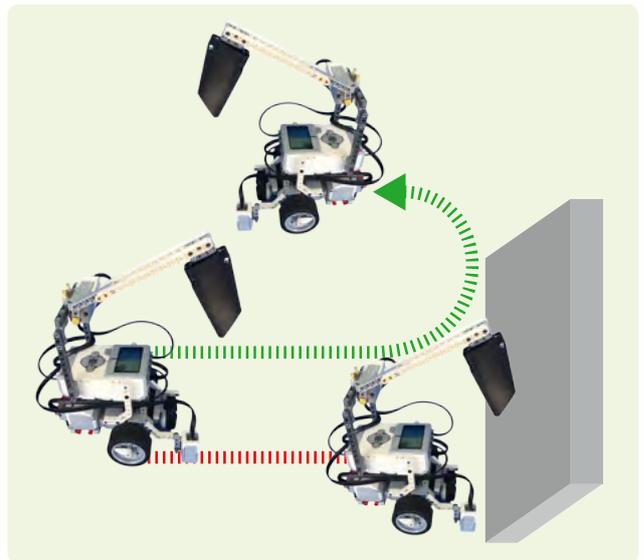
Capteur à ultrasons

Un capteur à ultrasons [7] génère des ondes sonores pour détecter et mesurer la distance des objets. Il peut également envoyer des ondes sonores pour fonctionner comme un sonar ou recevoir une onde sonore qui démarre un mode programme.

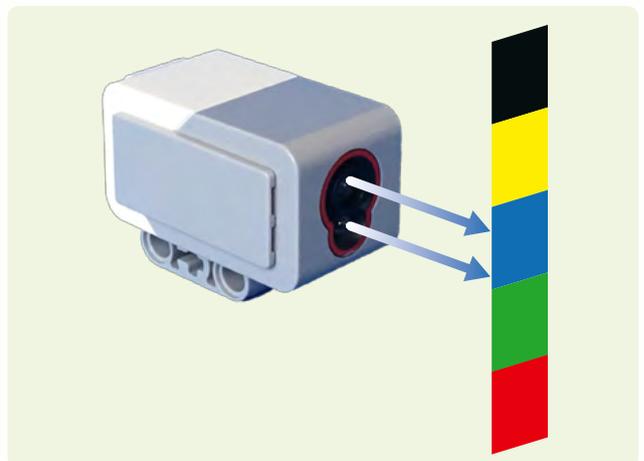
Grâce au capteur à ultrasons, le robot peut détecter les obstacles et réagir de différentes manières, en fonction du code. Par exemple, le robot peut être programmé pour s'arrêter ou changer de direction. Dans le modèle de barrage, les obstacles sont les rives en carton.



6 : Un capteur à ultrasons LEGO



7 : Arrêt du robot/Changement de direction du robot



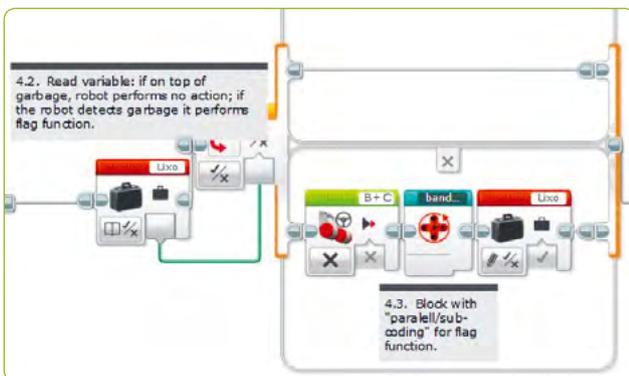
8 : Exemple d'un capteur de lecture couleur : Il distingue les couleurs en lisant leur code RGB.

Capteur de couleur

Le capteur de couleur^[8] peut détecter différentes couleurs et l'absence de lumière. Il fonctionne également comme un capteur de lumière en détectant les différentes intensités de lumière. Les élèves peuvent construire des lignes de couleurs différentes pour que le robot les suive.

Construction du code, à l'aide de blocs de programmation LEGO

Les élèves doivent construire différents modèles pour simuler différents types de déchets, tels que les déchets domestiques, industriels, touristiques, organiques, etc. L'objectif principal est de sensibiliser les élèves à la pollution des rivières et des barrages. Les élèves doivent simuler un détecteur de déchets assemblé dans un bateau, puis planifier et construire un bateau collecteur de déchets à une date ultérieure.



© 9 : Un extrait du code LEGO. Le diagramme complet est disponible en ligne.^[14]

Le robot doit émettre un son particulier pour chaque type de déchet qu'il détecte. Pour ce faire, nous utilisons le capteur de couleur et des taches de couleur spécifique qui simulent chaque type de déchet.

Les élèves peuvent consulter les publications des organisations environnementales et gouvernementales et construire les différentes taches en fonction des statistiques de pollution.

Les élèves effectuent des sorties sur le terrain vers les rivières et les barrages afin d'examiner la qualité de l'eau et la pollution éventuelle. Ils doivent ensuite simuler ces observations

dans les modèles qu'ils construisent. À l'aide du robot, ils doivent scanner et noter les résultats dans un tableau (© 10).

Lorsque les élèves ont recueilli suffisamment de résultats, ils doivent présenter leur recherche à la classe. L'objectif est de permettre aux élèves de développer leur esprit critique, leurs capacités d'investigation et de codage. Lorsque les élèves regardent ce qui se passe dans nos cours d'eau, ils voient les conséquences de siècles d'aveuglement face aux problèmes environnementaux. Pour pouvoir le comprendre, ils doivent avoir acquis les compétences environnementales requises pour intervenir dans leur communauté. Par exemple, il peut s'agir d'alerter les gens sur la nécessité de prévenir les comportements nuisibles à l'environnement et à l'eau en particulier. De plus, ils doivent également être capables de planifier et de présenter des solutions lorsqu'ils détectent des problèmes. L'objectif global est d'accroître leur civisme et leur sens de la responsabilité environnementale au sein de leur communauté.

Attention : nos étudiants ont déjà construit et testé la version LEGO et sont encore en train d'améliorer la version Arduino. Le code complet utilisé pour programmer l'Arduino est disponible en ligne.^[14]

<Niveau avancé : Programmation de jeux éducatifs liés à l'environnement>

L'objectif principal est de sensibiliser les élèves à la pollution de l'eau. Les élèves utilisent Scratch^[10] pour programmer des jeux qui motivent les joueurs à aider à préserver et à protéger l'eau et ainsi encourager les gens à ne pas jeter de déchets dans les plans d'eau.

Notre premier jeu simule un petit poisson dans l'océan. Le poisson doit se nourrir tout en évitant les autres créatures marines (requins et crabes) et les chutes de déchets (verres, boîtes de conserve, etc.). Plus il mange, plus il grossit, et plus le joueur gagne de points dans le processus.

Le poisson ne doit pas entrer en collision avec des déchets ou d'autres poissons, sinon il se blesse et reçoit un bandage. Quand il y a trois bandages, la partie est terminée. Ce jeu est

© 10 : Tableau de scan des déchets, données relatives à deux sorties différentes et déchets collectés dans chacune d'elles par les équipes de nettoyage du club environnement de l'école.

Date	Type de déchets					Zone nettoyée
	Domestique	Industriel	Indifférencié	Biologique	Autres	
Avril 2018	3.450 kg			32 kg	8 kg	100 m ²
Mai 2018	0.730 kg			6 kg		100 m ²

amusant et sensibilise non seulement les enfants, mais aussi les adultes à la quantité croissante de déchets dans les milieux aquatiques (océans, rivières, etc.).

Le deuxième jeu est basé sur un jeu vidéo bien connu, où une grenouille doit traverser une rue. Mais dans notre cas, le personnage doit traverser une rivière (en utilisant les bûches, car l'eau coule rapidement) et éviter les déchets ainsi que les autres animaux (chauves-souris et serpents). Il peut aussi manger des mouches pour gagner des points supplémentaires.

Dans ce jeu, il y a quatre scénarios différents et un scénario est choisi au hasard au début de chaque tour. Le personnage, par exemple une grenouille, a trois vies, après quoi le jeu se termine.

La section suivante contient des détails sur le programme.

Ⓒ11 montre la partie du programme qui contrôle le mouvement de certains ennemis dans les différents modes de jeu. Dans l'exemple affiché, l'ennemi disparaît lorsqu'il touche l'un des bords. Tant qu'il ne touche pas les bords, il répète le même mouvement, qui augmente également en vitesse avec un facteur d'ajustement de 0,04 lorsque le score du joueur augmente. C'est une façon très intelligente de rendre le jeu un peu plus difficile à mesure que le score augmente avec le niveau de difficulté.

Début de la partie : choisissez l'un des trois modes de jeu (Ⓒ12). En ce moment, deux jeux sont prêts et les élèves sont en train d'en développer un troisième appelé Mode de jeu 2.

Par exemple, le mode 2 pourrait se trouver dans un étang où les canards doivent attraper de la nourriture.

Les canards se nourrissent régulièrement de petits poissons et d'œufs de poissons, d'escargots, de vers, de mollusques et de petits crustacés comme les écrevisses, d'herbe, de feuilles, de mauvaises herbes, d'algues, de plantes aquatiques, de racines, de petites grenouilles, de salamandre et d'autres amphibiens. De plus, les canards doivent essayer d'éviter d'autres canards ou déchets dans l'étang (ou dans les niveaux avancés, des braconniers).

Si le poisson touche l'un des ennemis (1, 2 ou 3), il perd une vie et on entend un son.

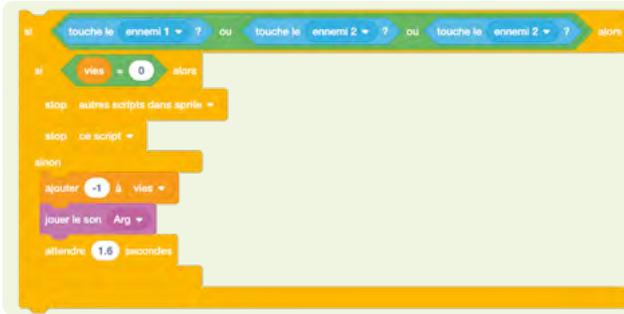
The script starts with 'when I start as a clone'. It goes to 'AI Director', shows the clone, and checks if 'Game Mode = 1'. If true, it checks if 'x coordinate > 140'. If true, it turns -90 degrees and enters a loop: repeat 10 times 'add score * -0.04 + -6 to x', then repeat until 'touch border?' 'add score * -0.04 + -6 to x', then repeat 8 times 'add score * -0.04 + -6 to x', and finally 'delete this clone'. If the mode is not 1, it turns 90 degrees and enters a loop: repeat 10 times 'add score * 0.04 + 6 to x', then repeat until 'touch border?' 'add score * 0.04 + 6 to x', and finally repeat 8 times 'add score * 0.04 + 6 to x' before deleting the clone.

Ⓒ11 : Contrôle de l'ennemi par programme Scratch

The script starts with 'when I receive "Charger Jeu"'. It waits 0.8 seconds, then checks 'Game Mode = 1'. If true, it goes to (0,0), switches to 'Poisson' costume, and sets size to 27% of original. If 'Game Mode = 2', it goes to (0,-4), switches to 'Chien' costume, and sets size to 24% of original. If 'Game Mode = 3', it goes to (0,-160), switches to 'Grenouille' costume, sets size to 70% of original, turns 90 degrees, shows the clone, and moves 5 pixels back.

Ⓒ12 : Démarrage du programme Scratch

Si le joueur perd toute sa vie, la partie est terminée, c'est-à-dire que tous les scripts sont arrêtés ([13]).



© 13 : Programme Scratch des ennemis 1-3

Le jeu est très bien programmé et construit, car le même code est utilisé pour les deux versions du jeu. Le même personnage change son apparence de requin à chauve-souris.

Ceci est possible parce que les jeux ont des objectifs similaires :

- ↳ éviter les ennemis
- ↳ attraper de la nourriture ou des mouches
- ↳ perdre 3 vies signifie que le jeu est terminé
- ↳ gagner des points (le poisson en mangeant de la nourriture pour poissons, la grenouille en mangeant des mouches et en atteignant un nouveau scénario)

Ils utilisent des clones des personnages ennemis pour pouvoir faire apparaître le même personnage dans différentes directions et avoir des comportements différents dans le jeu. Le programme complet est disponible pour téléchargement. [14]

<Conclusion>

Dans ce module, les élèves apprennent et partagent leurs connaissances sur l'eau : cycle de l'eau, pénurie d'eau, pollution, etc. Ils développent également des ressources pour surveiller, économiser et protéger l'eau. En même temps, les étudiants développent des outils d'investigation et des compétences en codage, ainsi que des compétences dans le domaine de la robotique. Lorsque les élèves les plus âgés encadrent et soutiennent les plus jeunes, ils se motivent les uns avec les autres et se mettent au défi de faire progresser leur travail. Cela contribue grandement au succès des projets.

À la fin de cette année scolaire, nous avons constaté que les élèves avaient non seulement amélioré leurs compétences en programmation, mais qu'ils étaient également plus conscients des problèmes liés à l'eau et des dangers pour les animaux et les plantes.

Il n'est pas facile de coder plusieurs jeux en un seul. Les jeux doivent avoir quelques similitudes afin que le code d'un jeu puisse être adapté pour servir tous les modes. Cependant,

c'est une façon intelligente d'économiser sur les ressources de codage.

Nous avons choisi de travailler ensemble, bien que dans des écoles différentes et éloignées les unes des autres, parce que cela nous a permis de partager des idées et d'améliorer le travail collaboratif entre élèves d'origines sociales, économiques et d'âges différents. Ce n'était pas facile de faire rencontrer les élèves ou de les réunir autant que prévu, mais cela s'est avéré être une bonne option, car cela leur a permis de partager leurs idées et leurs méthodologies et d'interagir avec des pairs inconnus, ce qui a amélioré leurs compétences en communication. Il leur a été également offert la possibilité de participer à différents concours. Une alternative aux réunions en présentiel pourrait être de communiquer en ligne par vidéoconférence. Enfin, les élèves ont pu partager leur travail avec la communauté et jouer un rôle dans le changement des attitudes locales en matière de protection de l'eau.

<Activité de coopération>

« Science on Stage, c'est partager les ressources entre les enseignants ! » Grâce à ce projet, la communauté d'enseignants a été renforcée et des ressources ainsi que des idées ont été partagées. Cela a contribué à améliorer l'apprentissage des étudiants dans toute l'Europe. Partager et collaborer est le meilleur moyen pour nous tous d'améliorer et de développer davantage ces projets.

<Références>

- [1] <https://calliope.cc/en>
- [2] <https://education.lego.com>
- [3] Peut-être des robots : Bee Bots de tts, DOC de Clementoni, Jack d'Imaginarium
- [4] www.weemake.com/
- [5] Anprino est un robot développé par l'Association Nationale Portugaise des Professeurs de Technologie de l'Information (ANPRI) ; informations et fichiers d'impression 3D www.anpri.pt/anprino/index.php/anprino-luis [29/11/2018]
- [6] www.arduino.cc/
- [7] Nous avons utilisé le capteur à ultrasons HC-SR04.
- [8] Nous avons utilisé le capteur à ultrasons BE15000624.
- [9] <https://snap.berkeley.edu/>
- [10] <https://scratch.mit.edu/>
- [11] <https://lab.open-roberta.org/>
- [12] <https://education.lego.com/en-us/downloads/wedo-2/software> [29/11/2018]
- [13] LEGO SET 31068
- [14] Tous les autres documents sont disponibles à l'adresse www.science-on-stage.de/coding-materials.

VPLS – Sauvegarde de la vie des plantes en vacances

<Auteur> Andreas Meier

<Auteur> Sonja van der Byl



<Info>

<Mots-clés> arrosage automatisé d'une plante, utilisant un microcontrôleur pour contrôler une pompe à eau

<Disciplines> informatique, sciences naturelles, technologie

<Niveau d'âge des élèves> 10–14 ans

<Matériel> ordinateur (un par élève si possible), Calliope mini^[4] avec un capteur d'humidité et de température (un par groupe)

<Langage> Scratch^[2] (en ligne ou hors ligne), éditeur pour Calliope^[3] (en ligne)

<Niveau de programmation> facile

<Résumé>

Les plantes en pot dans les bâtiments scolaires meurent souvent pendant les vacances d'été parce que personne ne s'occupe d'elles – c'est pourquoi nous avons besoin d'un dispositif pour préserver la vie végétale en vacances. Dans cette unité d'enseignement, nous allons développer un système de survie virtuel et réel pour les plantes scolaires.

<Introduction des concepts>

Le projet convient à tous les sujets STEM parce que le niveau de programmation requis est simple. Dans la première partie, l'arrosage des plantes à intervalles réguliers est réalisé de façon virtuelle. La programmation couvrira diverses structures de contrôle en informatique telles que l'orientation objet, les boucles et les conditions ainsi que l'utilisation de variables. Les élèves feront du codage de base pour la première fois et travailleront à l'aide d'un tutoriel en ligne dans « Getting Started with Scratch »^[4] (durée : 45 minutes).

La programmation orientée objet (POO) est un type de programmation qui prête des propriétés (attributs) et des capacités (méthodes) à un objet. Dans notre cas, ces objets sont un chat appelé Sprite, un arrosoir et une scène. Chaque objet appartient à une classe. Tous les objets d'une même classe ont les mêmes propriétés et capacités. Vous pouvez interpréter une classe comme un plan et un objet comme une instance, c'est-à-dire une réalisation concrète du plan. Dans Scratch, les figures sont des instances de la classe « Sprite », ou en bref, « sprites ». Le chat appelé « Sprite » est un sprite, c'est-à-dire une instance de la classe « Sprite ».

L'un des attributs d'un sprite est son costume ; dans ce cas, c'est l'image d'un chat. Un autre exemple de la classe « Sprite » pourrait avoir l'image d'un humain en costume et s'appeler Gunther. Le chat « Sprite » et l'humain « Gunther » sont tous

deux des objets (instances) de la classe « Sprite » ou en bref : les deux sont des sprites. Il n'y a que deux classes dans Scratch : la scène et les sprites. Dans ce projet, nous avons deux sprites (le chat et l'arrosoir) et la scène (📍1).



📍 1 : Dispositif pour préserver la vie végétale en vacances

La deuxième partie du projet peut être réalisée après la fin de la première partie. Elle peut également être réalisée indépendamment si les étudiants connaissent les structures de contrôle mentionnées précédemment et ont déjà acquis une première expérience de programmation avec le Calliope mini^[4]. Des capteurs pour microcontrôleur qui commandent la vanne d'une pompe à eau seront utilisés à la place des variables.

<Ce que font les élèves/enseignants>

Tout le matériel et les feuilles de travail nécessaire sont disponibles en téléchargement.^[5]

<Partie 1: Arrosage virtuel régulier des plantes>

Étape 1 : Coder un programme avec une seule variable, le temps ; apprendre à connaître des conditions unilatérales et des boucles [durée : 180 minutes].

Après avoir analysé le problème (« Comment un programme virtuel de préservation de la vie des plantes en vacances pourrait-il fonctionner ? »), on demandera aux élèves de réfléchir à la structure de base d'un tel programme. Ils prendront ensuite des notes sous la forme d'une recette (algorithme) et testeront leurs idées respectives. Ce n'est qu'après cela qu'ils s'entendront sur une structure de base commune pour le programme (voir la fiche de travail 1^[5]).

Cette phase aidera les élèves à réfléchir aux structures de base du programme qu'ils veulent coder. Les Mots-clés « liste d'énoncés », « boucles » et « conditions » sont dérivés du contexte.

↳ Les élèves vont maintenant réaliser le programme en Scratch^[2] en assemblant les composants individuels^[5]

fournis en un programme fonctionnel. Les élèves en apprendront davantage sur Scratch et les aspects suivants, durant le processus :

- ↳ orientation objet (chaque figure a son propre script, même la scène)
- ↳ structure (à quoi ressemble la structure d'une condition ou d'une boucle dans Scratch ?)
- ↳ script/costume/sons peuvent être assignés à chaque figure individuellement

De plus, les élèves en apprendront davantage sur la structure de base du programme d'arrosage des plantes tout en réfléchissant à la meilleure façon de commander les différentes parties du code pour que le programme fonctionne. Il est particulièrement important de décider quelles commandes doivent se trouver à l'intérieur de la boucle de comptage [2 & 3] ; ceci peut être fait par tâtonnements.

2

3

En se basant sur le programme assemblé dans la section précédente, les élèves pourront maintenant créer leur propre programme dans lequel le chat « Sprite » se déplace vers les

plantes et les arrose en fonction de la variable temps. Le seul fichier qui sera remis aux élèves est celui de la scène de départ.^[5]

Les élèves seront encouragés à explorer le langage de programmation de façon indépendante afin de tester leurs propres idées et d'être créatifs. Il est important que les étudiants soient capables d'utiliser le langage de programmation requis avec confiance (en fonction de leur niveau de connaissance respectif) afin qu'ils apprécient davantage l'activité.

Étape 2 : Rédiger un programme qui incorpore les variables « niveau d'eau » et « température » (temps requis : environ 270 minutes).

En guise d'introduction à la deuxième étape de l'unité d'enseignement, les élèves réfléchiront à d'autres facteurs qui déterminent la fréquence à laquelle une plante en pot doit être arrosée. La « température ambiante » et le « niveau d'eau » dans le contenant de la plante joueront certainement un rôle à cet égard en tant que variables changeantes (voir la fiche de travail 2^[5]).

Les élèves recevront un programme de travail dans lequel les scripts pour le chat « Sprite » et l'arrosoir sont pratiquement les mêmes qu'avant.

Cependant, il y aura un nouveau script pour l'étape qui contrôle la variable « niveau d'eau » à la place de la minuterie précédente. Le chat n'arrose la plante qu'une seule fois. Les élèves seront encouragés à réfléchir à la façon dont la variable « niveau d'eau » peut être définie d'une part et comment elle contrôle l'activité du chat d'autre part. D'autre part, ils seront chargés de résoudre le problème afin que la plante ne soit arrosée qu'une seule fois. Un fichier d'aide est disponible si nécessaire.^[5]

En n'utilisant qu'une seule variable, le programme restera clairement structuré. Il peut être trop difficile pour un programmeur débutant de coordonner plusieurs variables à la fois.

La structure de boucle utilisée est plus complexe qu'auparavant, car elle est liée à une condition [4]. Les élèves de-

4

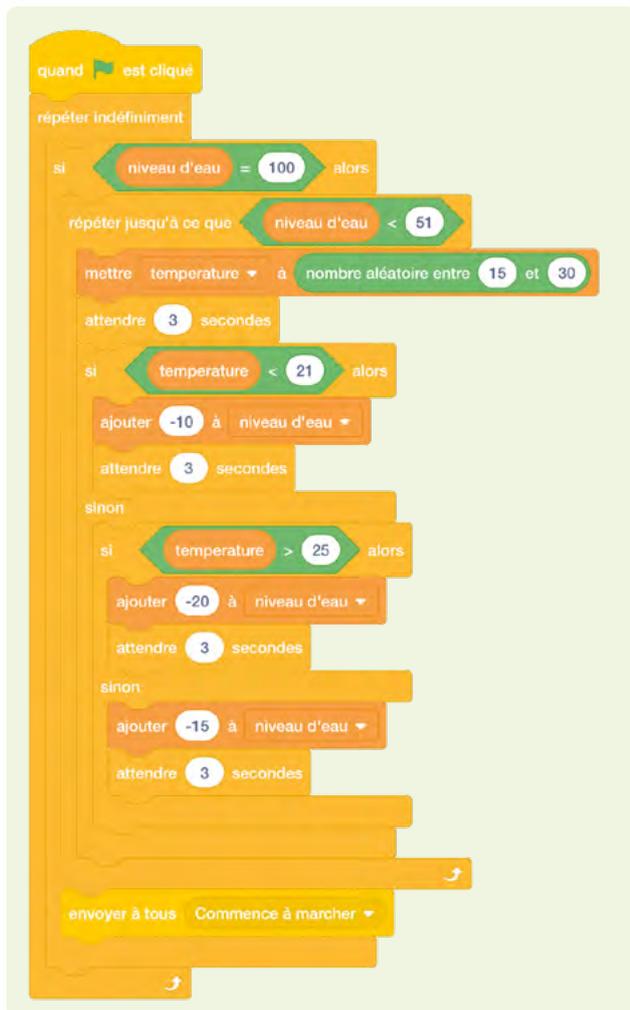
vront réfléchir attentivement à ce qui doit être répété, à la fréquence et aux conditions de répétition.

La capacité de structuration est fondamentale dans l'apprentissage de n'importe quel langage de programmation, mais elle sera enseignée d'une manière très amusante ; les étudiants seront capables de tout essayer sans conséquences négatives.

Les apprenants rapides auront également l'occasion de changer le programme et d'essayer leurs propres idées à la fin de cette phase de travail.

Dans l'étape suivante, la variable « température » joue un rôle dans le programme. Sa valeur est déterminée par un générateur de nombres aléatoires, qui fournit un nombre entre 15 °C et 30 °C. Le niveau d'eau dans le bac à plantes varie en fonction de cette valeur [fiche de travail 3^[5]].

Étant donné que la structure du « nouveau » programme d'arrosage est assez complexe, un programme désassemblé doit d'abord être réassemblé en un programme fonctionnel^[5].



© 5

Le générateur de nombres aléatoires et les conditions à deux entrées seront ensuite introduites. De plus, les élèves travaillent de façon répétée avec des variables ainsi qu'avec des conditions d'interrogation et approfondissent ainsi leur compréhension de celles-ci. Encore une fois, les élèves devront réfléchir attentivement à ce qui doit être répété, combien de fois et dans quelles conditions [5].

Comme à l'étape précédente, les étudiants auront la possibilité de personnaliser le programme résultant en travaillant selon leurs idées et leurs capacités de programmation.

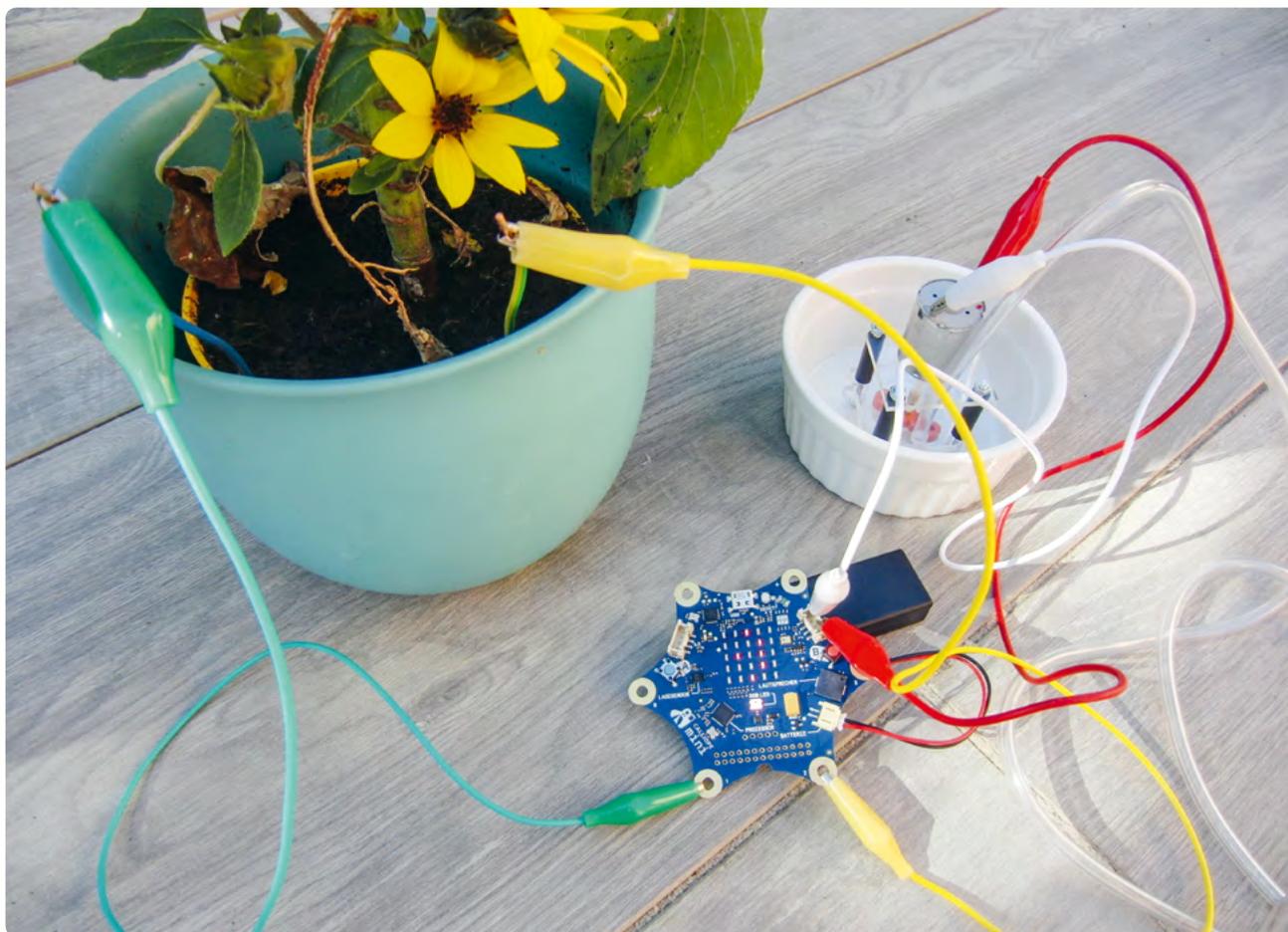
À la fin de la première partie du projet, où le dispositif pour préserver la vie végétale en vacances a été réalisé en termes virtuels, une grande variété de résultats de programmation devrait être présentée pour montrer les idées des étudiants et apprécier leurs performances.

<Partie 2 : Arrosage régulier d'une plante contrôlée par un microcontrôleur>

Au cours du projet, il est rapidement apparu que certains élèves n'étaient pas satisfaits de la solution virtuelle au problème de l'arrosage. Ils ont demandé comment arroser de vraies plantes avec l'aide de leur programme. Le programme d'arrosage virtuel est relativement facile à transférer sur un VPLS réel à l'aide d'un microcontrôleur, d'autant plus que beaucoup de ces mini-ordinateurs peuvent également être programmés avec Scratch^[2] ou une application similaire. Dans notre projet, nous utilisons le Calliope mini^[4], un microcontrôleur similaire au micro:bit^[6] de la BBC, qui contient également des fonctions plug-and-play conviviales telles que des capteurs tactiles et des connexions moteur. Cependant, le VPLS peut également être contrôlé avec tous les autres microcontrôleurs couramment utilisés dans les écoles, tels que LEGO EV3, LEGO NXT, Arduino, Raspberry Pi, Teensy, etc.^[5] C'est un processus simple pour programmer le Calliope mini, car il vous suffit de le connecter à votre ordinateur via un câble USB. Open Roberta Lab^[7], qui supporte plusieurs microcontrôleurs (des éditeurs alternatifs sont disponibles^[5]), est une interface de programmation adaptée. L'interface de programmation est disponible en plusieurs langues et vous pouvez changer de langue en cliquant sur l'icône du globe après avoir sélectionné un microcontrôleur, dans notre cas le Calliope mini.

Une version simple du VPLS pourrait fonctionner comme suit :

1. L'humidité du sol est constamment mesurée.
2. Si le sol est trop sec, une certaine quantité d'eau est pompée jusqu'à ce que le sol soit suffisamment humide.



© 6 : VPLS contrôlé par un Calliope mini

Le microcontrôleur doit donc être capable de mesurer l'humidité du sol et de contrôler le moteur d'une pompe à eau.

Le Calliope mini dispose de quatre capteurs tactiles. La physique derrière eux est qu'ils mesurent la conductivité électrique entre les points de connexion. Comme l'eau conduit l'électricité, le sol humide a une conductivité plus élevée que le sol sec. Il vous suffit d'utiliser deux fils de cuivre comme capteurs, que vous placez ensuite dans le pot de fleurs à une certaine distance l'un de l'autre. Ils sont ensuite reliés au Calliope mini à l'aide de pinces crocodile avec les contacts aux coins (-, P1). La valeur de sortie du capteur P1 (broche analogique) sera comprise entre 0 et 1023. Si la conductivité diminue jusqu'à une certaine valeur, la pompe qui arrose l'installation sera activée (voir ©6).

Un composant essentiel de la pompe est un petit moteur électrique, qui est connecté au Calliope mini soit directement, soit à l'aide d'un pilote moteur supplémentaire, selon le niveau de puissance requis. Deux ports moteurs (A, B) sont disponibles dans le menu Action/Déplacer. La quantité d'eau de la pompe peut être réglée en changeant la valeur « speed % ».

Un certain niveau d'artisanat est bien sûr nécessaire pour la construction de la pompe et de l'entraînement du moteur. Nous fournissons les manuels de construction nécessaires dans le matériel en ligne^[5]. La pompe ne coûte que quelques euros.

Pendant qu'ils travaillent avec le VPLS réel, les étudiants se poseront de nombreuses questions sur la façon de l'optimiser, ce qui approfondira leurs connaissances dans le processus, par exemple :

- ↳ Nos plantes ont-elles besoin de beaucoup d'eau, ou juste un peu ? Quelle doit être la taille du réservoir ?
- ↳ Nos plantes ont-elles besoin de beaucoup d'eau, ou juste un peu ? Quelle doit être la taille du réservoir ?
- ↳ Quel est le meilleur moment pour arroser les plantes ? Et, est-il préférable d'arroser beaucoup seulement une fois par jour ou moins, mais plusieurs fois par jour ?
- ↳ À quelle profondeur les capteurs d'humidité doivent-ils être placés dans le sol pour fournir des mesures optimales ? Quelle est la distance idéale entre les capteurs ?
- ↳ Quelle est la durée de vie de l'alimentation du VPLS ? L'efficacité énergétique de la pompe peut-elle être augmentée de manière à ce que le VPLS arrose pendant toute la durée des vacances scolaires ?

Comme vous pouvez le constater, le projet VPLS offre aux étudiants différentes approches dans les domaines de la biologie et de la physique pour de futures expériences ou travaux de projet. Une autre option intéressante serait de connecter le VPLS à Internet et de le surveiller en ligne (Internet des objets). Bien que cela irait bien au-delà de notre introduction à la programmation avec le VPLS, cela montre ce qui peut résulter d'une simple question.

<Transférabilité à d'autres langages de programmation>

Le projet peut facilement être transféré dans Snap!^[8], qui est une évolution de Scratch^[2]. Des exemples de programmation sont fournis en ligne^[5]. Ces deux langages de programmation sont particulièrement bien adaptés à un projet destiné aux programmeurs débutants, car ils sont faciles à comprendre et encouragent les étudiants à essayer des idées en utilisant le « glisser-déposer ».

<Conclusion>

Les étudiants novices en programmation feront leurs premiers pas sur le terrain et apprendront les bases importantes d'un langage de programmation dans le cadre de ce projet, qui a ses racines dans une situation de la vie réelle. Il ne s'agit pas d'apprendre la syntaxe et le vocabulaire d'un langage de programmation, mais plutôt de tester les effets de certaines structures : « Qu'est-ce qui fonctionne et pourquoi ? ».

Les erreurs sont les bienvenues, car elles sont généralement faciles à corriger et à expliquer, et aident ainsi les étudiants à comprendre le fonctionnement d'un langage de programmation.

D'une part, le cadre posé donne la sécurité aux étudiants (ceux qui ne sont pas sûrs ne résoudre que le puzzle de l'assemblage des parties de programme fournies), et d'autre part, il y a beaucoup de place pour la créativité des étudiants qui font les « tâches obligatoires » rapidement.

Quelques idées ont émergé du projet, par exemple pour construire un « vrai » robot arroseur ou pour développer un jeu d'arrosage. Quoi qu'il en soit, les élèves ont acquis une expérience initiale positive en matière de programmation qui, nous l'espérons, aura un effet durable pour eux.

Les délais fixés à notre école étaient parfois difficiles à respecter. Nous n'avions que 45 minutes par leçon pour travailler sur ce projet. La partie organisationnelle des leçons (se connecter à l'ordinateur, ouvrir des fichiers, sauvegarder des fichiers, se déconnecter de l'ordinateur) a pris 15 minutes, donc il n'y avait pas assez de temps pour la programmation et le travail réel. Vous pouvez demander l'accès professeur à

Scratch^[2], ce qui vous permettra de configurer une classe et de déposer du matériel.

<Activité de coopération>

Comme le VPLS est une introduction à la programmation, il y aura très peu de possibilités de coopération.

Dès que le projet est transféré à la commande réelle d'un microcontrôleur, la coopération entre étudiants serait utile, car le degré de difficulté du problème augmente en utilisant des composants supplémentaires (capteurs, pompe). Par exemple, les élèves plus âgés pourraient aider à construire la pompe. Des écoles de différents pays pourraient travailler ensemble sur le projet VPLS et comparer leurs résultats et solutions.

Une base de données commune pourrait être créée pour différentes plantes afin d'adapter le VPLS aux caractéristiques individuelles d'une variété d'espèces végétales. Si le VPLS est connecté à Internet, les écoles pourraient adopter des plantes d'une autre école et prendre en charge l'arrosage.

<Références>

- [1] www.calliope.cc/en
- [2] www.scratch.mit.edu/
- [3] www.calliope.cc/en/los-geht-s/editor
- [4] https://scratch.mit.edu/projects/editor/?tip_bar=getStarted [29/11/2018]
- [5] Tous les autres documents sont disponibles à l'adresse www.science-on-stage.de/coding-materials.
- [6] www.microbit.co.uk/home
- [7] <https://lab.open-roberta.org>
- [8] <https://snap.berkeley.edu/>



Gant magique

<Auteur> Annamária Lőkös

<Auteur> Camelia Ioana Rațiu

MAGIC



<Info>

<Mots-clés> expérience, environnement, température, humidité, luminosité, champ magnétique

<Disciplines> physique, chimie, biologie, écologie, écologie, informatique

<Niveau d'âge des élèves> 10–18 ans

<Niveau 1> pour l'école primaire (âge : 10–11 ans) et secondaire (collège, âge : 12–15 ans)

<Niveau 2> pour le secondaire (lycée, âge : 15–18 ans)

<Matériel> Arduino UNO^[1], capteurs compatibles avec Arduino (par ex. capteur de lumière, capteur de température, capteur de champ magnétique, capteur d'humidité, capteur de gaz), écran LCD, fils de liaison, batterie externe

<Langage> C^[2], Arduino 1.8.5^[3], Snap!^[4]

<Niveau de programmation> moyen

<Résumé>

Les jeunes sont passionnés par la technologie, alors une leçon qui combine la science et l'informatique est forcément un succès. Les élèves fabriquent et utilisent un gant avec un capteur différent à chaque doigt. Cela leur permet d'effectuer différentes expériences en ne connectant que le capteur nécessaire.

<Introduction des concepts>

L'avantage d'utiliser un appareil (un gant) équipé de plusieurs capteurs pour différentes mesures est double. Les élèves du primaire et du secondaire peuvent utiliser le gant « magique » pour mesurer la température, la luminosité, l'humidité, la présence d'un champ magnétique, l'intensité sonore, etc. Tout ce qu'ils ont à faire est de sélectionner le capteur désiré et ils sont prêts à commencer à trouver diverses utilisations pour le gant dans différents domaines d'études et matières scolaires. Le gant peut être utilisé sur le terrain, car il est alimenté par une batterie, ce qui offre aux étudiants la possibilité de faire des recherches en dehors du laboratoire. D'un autre côté, les élèves du secondaire peuvent fabriquer eux-mêmes un gant pour prendre certaines décisions. Les étudiants connaissent les notions théoriques des différentes sciences (physique, chimie, biologie, écologie), et ils apprécient vraiment l'opportunité de les étudier dans des expériences pratiques.

Les enseignants doivent présenter les concepts de base du codage d'un programme en C^[2] ou tout autre langage de programmation supporté par Arduino, y compris Snap!^[4], pour que les élèves codent l'Arduino^[1]. Pour acquérir une connaissance de base du C, s'ils le choisissent comme langage de programmation pour Arduino, les élèves peuvent regarder des tu-

toriels sur Internet. Cela les aidera à mieux comprendre comment le code devrait être écrit et renforcera aussi leur confiance, car la plupart d'entre eux seront surpris par la simplicité du code.

<Ce que font les élèves/enseignants>

<Niveau 1>

Le gant a un écran LCD avec des boutons. Les élèves enfilent le gant et sélectionnent le capteur désiré à l'aide des boutons UP/DOWN ; ils appuient ensuite sur le bouton SEL et la mesure commence. L'écran affiche la valeur. Les élèves peuvent répéter les mesures quand ils le souhaitent. Pour revenir à l'écran d'accueil, il faut appuyer sur le bouton RETOUR. Par exemple, vous pouvez voir la détermination des pôles d'un aimant à la @1a–1c.



@ 1a-c : Déterminer les pôles d'un aimant.

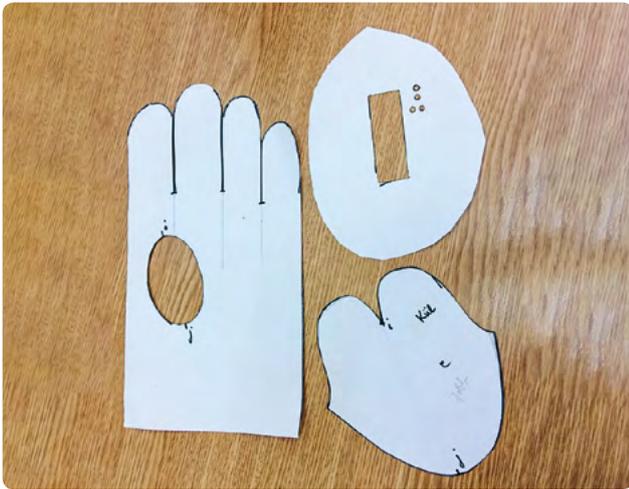
<Niveau 2>

Les élèves d'une classe peuvent être divisés en quatre groupes. Un groupe coupe et coud le gant, le deuxième groupe fait le circuit, le troisième groupe fait le codage et le dernier groupe calibre les capteurs.

<Fabrication du gant>

Les élèves ont réalisé un modèle (@2) après avoir consulté de la documentation sur plusieurs sites Web^[5]. Ils ont plié le matériau (le cuir dans notre cas, mais d'autres matériaux

peuvent être utilisés] en trois et l'ont coupé avec le gabarit. Pour obtenir le bon gant, les élèves ont cousu deux des côtés ensemble. Ils ont cousu la dernière pièce de tissu après avoir monté l'Arduino avec l'écran LCD et les capteurs à l'arrière de la paume. Les élèves découpent l'ouverture pour l'écran LCD et les boutons de cette troisième pièce.



© 2 : Gabarits pour le gant

<Construire le circuit>

Les élèves ont fait le circuit, à partir d'un schéma qu'ils ont discuté et analysé avec l'enseignant au préalable. Le circuit peut être fixe (étamé) ou non. Il doit tenir compte de la connexion correcte des capteurs à la carte Arduino, à savoir le GND du capteur au GND sur la carte Arduino, le VCC du capteur au 5V sur l'Arduino, et le OUT sur le capteur à l'une des ANALOG IN (A0, A1, A2, A3, A4 ou A5) de la carte. Si un capteur doit être connecté à l'ENTRÉE NUMÉRIQUE, il faut veiller à ne pas utiliser l'une des entrées utilisées pour l'écran LCD, car des erreurs de fonctionnement peuvent se produire. Dans notre exemple, nous avons connecté les capteurs suivants : température (A1), lumière (A2), humidité (A3), magnétisme (A4) et proximité (A5) (voir © 3a–3e).^[6]

<Écrire le code du circuit>

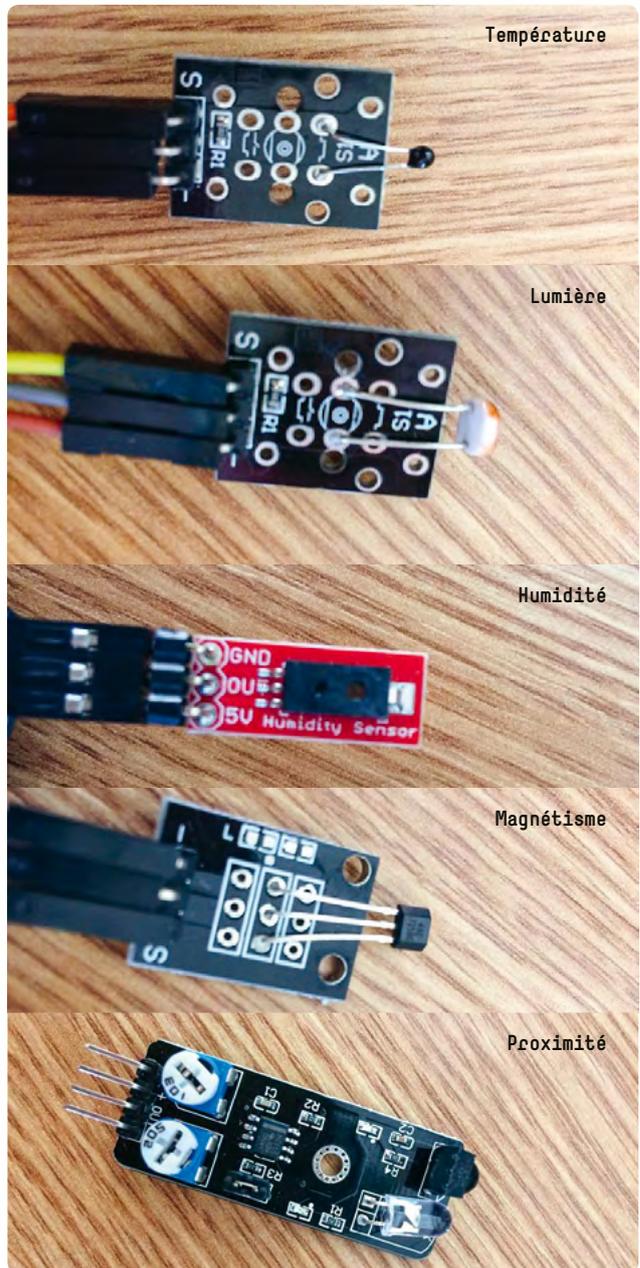
Les lycéens qui étudient le langage de programmation C^[2] peuvent facilement programmer Arduino^[1]. Il existe de nombreux tutoriels disponibles en ligne dans une multitude de langues. Par exemple, nos étudiants ont utilisé un site Web en roumain^[7]. Des tutoriels en anglais sont entre autres proposés sur le site Arduino ou sur les sites des distributeurs. Il y a aussi beaucoup d'autres sites où les élèves peuvent trouver des tutoriels.

Le professeur peut les guider à travers la façon d'écrire le programme pour Arduino, et vous pouvez trouver le code complet que nous avons utilisé en ligne.^[8]

<Étalonnage des capteurs>

Il existe des capteurs calibrés, mais aussi des capteurs non calibrés ; il est parfois plus agréable pour les élèves de trouver un moyen de les calibrer. Ils ont trouvé des formules d'étalonnage pour certains des capteurs sur Internet. Par exemple, il existe une formule pour le bloc du capteur d'humidité^[6], car la fonction par laquelle les valeurs affichées varient n'est pas linéaire.

En ce qui concerne l'étalonnage du capteur de température, les étudiants ont suivi les valeurs affichées par le capteur. Ils ont utilisé un thermomètre étalonné en laboratoire et ont associé la valeur affichée à la valeur du thermomètre. Ils ont découvert que ce capteur varie linéairement et ont trouvé la formule d'étalonnage. Des exemples de formules d'étalonnage



© 3a-e : Différents capteurs

pour les sondes d'humidité et de température sont disponibles dans les documents supplémentaires fournis.^[9]

Une fois que les élèves ont étalonné les capteurs, terminé le programme et vérifié l'affichage pour s'assurer que les doigts du gant sont correctement corrélés avec les données affichées à l'écran, le gant est ensuite monté. La dernière étape consiste à coudre la couche extérieure du tissu. Les élèves ont utilisé des anneaux de fixation pour chaque capteur à chaque doigt (📍4) pour mieux fixer les capteurs.



📍 4 : Fixation des capteurs

<Algorithme à utiliser dans d'autres langues>

Si vous souhaitez utiliser un autre environnement de programmation, un diagramme avec tous les éléments nécessaires pour le programme principal est disponible en ligne.^[9]

<Conclusion>

Les élèves aiment découvrir de nouvelles choses et sont très inventifs. Ils aiment faire des mesures expérimentales, et le gant semble provenir d'un film de science-fiction. Les élèves du secondaire apprécieront les éléments de codage de l'unité et verront immédiatement les résultats d'outils pratiques qui fonctionnent vraiment.

Cette expérience a été sans précédent, les enseignants et les élèves ont appris beaucoup de choses ensemble.

Ils ont rencontré une difficulté : il n'est pas facile de trouver les bons capteurs^[6] et de les étalonner, mais il existe des solutions. Si une formule d'étalonnage ne peut être trouvée, une solution consiste à acheter des capteurs étalonnés, même s'ils sont plus coûteux.

Ce gant pourrait également être fabriqué avec un Calliope mini, ce qui le rendrait plus léger et plus petit. Nos étudiants sont intéressés pour faire un tel gant même si le langage de programmation est différent.

<Activité de coopération>

Des élèves de différentes écoles et de différents pays pourraient fabriquer de tels gants à l'aide de divers microcontrôleurs et de capteurs appropriés, puis discuter et comparer les résultats. On pourrait demander à un professeur d'art de contribuer à la conception du gant. En outre, un concours entre écoles pourrait être organisé, dans le cadre duquel les élèves proposeraient eux-mêmes différents modèles.

Le gant est facile à envoyer par la poste, ce qui permet aux élèves de différentes écoles d'expérimenter avec les gants fabriqués par leurs pairs dans d'autres écoles et d'autres pays.

<Références>

- [1] www.arduino.cc
- [2] [https://en.wikipedia.org/wiki/C_\(langage_de_programmation\)](https://en.wikipedia.org/wiki/C_(langage_de_programmation))
- [3] www.arduino.cc/en/Main/Software
- [4] <https://snap.berkeley.edu/>
- [5] Plusieurs sites web avec documentation et tutoriels sur la couture des gants :
<http://ofdreamsetseams.blogspot.ro/2012/04/1950s-het-sewn-leather-gloves.html>,
<https://so-sew-facile.com/facile-gloves-pattern-winter-comfort/>,
<http://sew-ing.com/make/gloves.html>,
www.glove.org/Modern/myfirstgloves.php,
www.instructables.com/id/How-to-Make-Gloves/
 [décembre 2018]
- [6] Nous avons utilisé les capteurs dans le « kit de capteurs 37 en 1 » pour Arduino. Le capteur d'humidité est le capteur d'humidité HIH-4030, de marque : Sparkfun, code : SEN-VRM-09.
- [7] www.robofun.ro (Tutoriels en roumain. Chaque produit a des instructions sur la façon de le connecter au circuit et sur le langage de programmation à utiliser. Le site Web contient des diagrammes et des dessins, des capteurs ou d'autres composants qui peuvent être reconnus à partir des photos, et le programme est très facile à suivre. Cela signifie que les étudiants n'ont pas besoin d'avoir une connaissance de la langue roumaine.)
- [8] <http://mthackathon.info/resources/37-SENSOR-KIT-TUTORIAL-FOR-UNO-ET-MEGA.pdf> (les tutoriels sont en anglais ; ils contiennent des instructions sur la façon de connecter les capteurs du kit à Arduino)
 [décembre 2018]
- [9] Tous les autres documents sont disponibles à l'adresse www.science-on-stage.de/coding-materials.

Science friction

<Auteur> Ilia Mestvirishvili

<Auteur> David Shapakidze



<Info>

<Mots-clés> force de frottement, distance d'arrêt, système de freinage antiblocage (ABS), programmation d'applications, acquisition de données

<Disciplines> physique, informatique, mathématiques

<Niveau d'âge des élèves> 14 ans et plus

<Matériel> Arduino^[1], servomoteur, moteur, module Bluetooth, écran moteur, écran moteur, photogate

<Langage> Environnement de programmation Arduino^[2], AppInventor^[3], Snap4Arduino^[4], Blockly^[5]

<Niveau de programmation> facile, moyen

<Résumé>

Une étude des facteurs affectant la force de frottement peut être transformée en une expérience intéressante et divertissante en construisant une voiture à faible coût contrôlée par Bluetooth avec un simple système de freinage. Cela permettra aux élèves d'observer des données réelles telles que la vitesse de la voiture (valeur absolue de la vitesse) avant d'appliquer les freins, sa distance d'arrêt et la façon dont la masse d'une voiture et le type de surface affectent la force de friction. Les élèves effectueront ensuite des expériences pour étudier la relation entre les facteurs affectant la distance d'arrêt avec une précision suffisante pour vérifier leurs propres hypothèses ou celles suggérées par l'enseignant.

<Introduction des concepts>

La friction est une force très importante dans la vie de tous les jours et est enseignée en physique au collège et au lycée. Cependant, les expériences traditionnelles liées au thème de la friction sont limitées et peu stimulantes. Ce projet transformera l'exploration de la friction en un projet de groupe passionnant qui implique :

1. la construction et la mise au point d'une voiture
2. programmation d'un microcontrôleur Arduino^[1] pour mesurer la vitesse instantanée et la distance d'arrêt
3. programmation d'un téléphone mobile avec AppInventor^[3] pour envoyer, recevoir et afficher des données réelles sur l'écran du téléphone

<Ce que font les élèves/enseignants>

Comme ces trois tâches peuvent d'abord être effectuées simultanément, nous recommandons que l'enseignant divise sa classe en groupes de deux ou trois élèves, qui travaillent ensuite sur les tâches séparément, mais se réunissent pour discuter et réviser leur travail. Lorsque la friction est introduite dans un programme d'études particulier, les ensei-

gnants pourraient proposer l'unité sur la friction dans le cadre de ce projet, ce qui motivera les élèves et améliorera leur compréhension des concepts théoriques connexes. La meilleure façon de commencer le projet serait de poser les questions suivantes à vos élèves et de leur donner le temps de réfléchir et de formuler leurs propres idées, prévisions ou hypothèses :

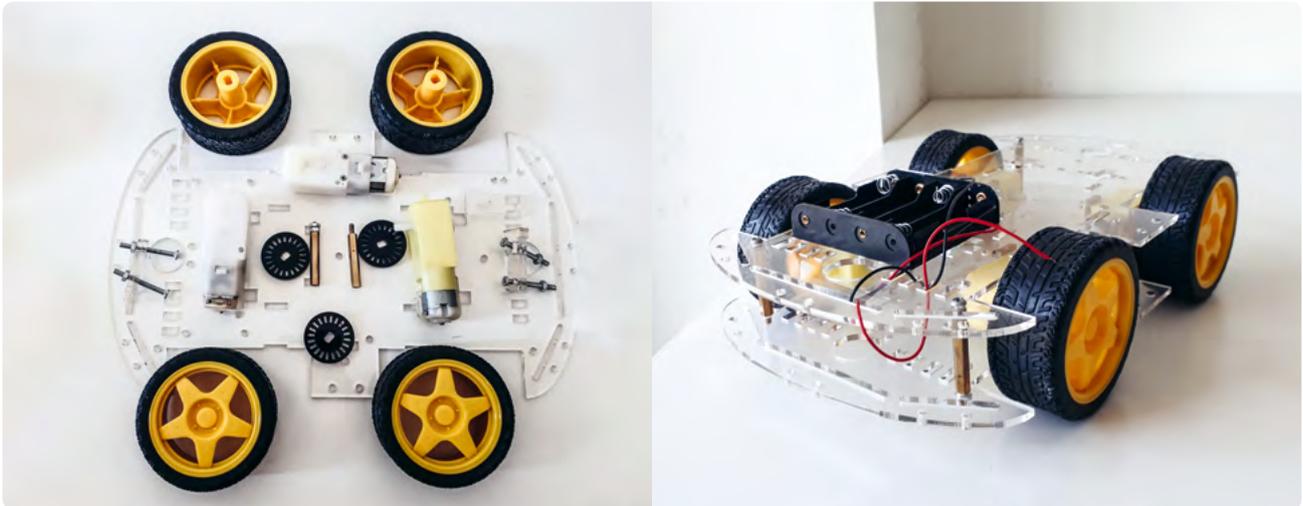
- ↳ Quelle est la relation entre la vitesse d'une voiture et la distance d'arrêt ? (Les réponses peuvent varier, bien sûr, c'est-à-dire : « la distance d'arrêt est proportionnelle à la vitesse avant d'appliquer les freins », ou certains pourraient « se souvenir » de la physique que la distance d'arrêt est en fait proportionnelle au carré de la vitesse).
- ↳ Comment l'augmentation de la masse d'une voiture affecterait-elle la distance d'arrêt, à condition que tout le reste ne soit pas affecté ? (Une réponse populaire est que l'augmentation de la masse devrait augmenter la distance d'arrêt.)
- ↳ Comment appliquer les freins pour arrêter une voiture le plus rapidement possible ? (Réponses possibles : la meilleure façon est d'arrêter complètement les roues ; si on les fait tourner dans la direction opposée au mouvement, cela arrêtera la voiture plus rapidement ; etc.)
- ↳ Si les freins avant et arrière sont identiques, arrêteront-ils la voiture en même temps ? (Les élèves peuvent réfléchir sur leur propre expérience avec les bicyclettes.)
- ↳ Toute autre question qui pourrait venir de l'enseignant ou des élèves.

Une fois que les élèves auront noté leurs idées initiales, l'étape suivante consistera à réfléchir à la façon de construire une voiture simple et aux données dont ils auront besoin pour vérifier et développer davantage leurs idées initiales. L'enseignant peut faciliter ce processus et suggérer aux élèves de construire une voiture capable de recueillir et d'envoyer des données pertinentes à un téléphone, qui à son tour peut contrôler la voiture ainsi que recevoir et afficher les données.

En fonction de leurs intérêts, de leurs compétences et de leurs préférences, les élèves peuvent, à ce stade du projet, se répartir dans les groupes mentionnés précédemment, s'ils le souhaitent. Cependant, un seul groupe d'élèves peut également accomplir toutes ces tâches. Les prochaines étapes de l'élaboration du projet seront les mêmes pour les deux scénarios.

<Construction d'un châssis et montage des composants électriques>

Cette approche consiste à construire une voiture à partir du kit de châssis Arduino^[1] peu coûteux et facilement disponible représenté sur la  1. Les enseignants et les élèves sont encouragés à essayer différentes approches de conception et de mise en œuvre, par exemple, différentes façons de collecter et d'en-



© 1 : Un kit châssis complet

voyer des données, de contrôler la voiture à distance, ainsi que différents logiciels et langages pour écrire le code nécessaire.

Après avoir assemblé la voiture et décidé où monter le contrôleur Arduino et le moteur, les élèves devront réfléchir à différentes façons de mesurer la vitesse de la voiture. L'approche recommandée est de faire un remue-ménages et de donner aux élèves l'occasion de proposer leurs propres idées. Avec l'aide appropriée de l'enseignant, la meilleure et la plus simple façon de le faire est d'utiliser un photogate (photo résistance et grille rotative) pour mesurer la vitesse de rotation d'une roue arrière libre. De cette façon, les élèves pourront déterminer à la fois la vitesse instantanée et la distance parcourue. Pour ce faire, il suffit d'utiliser les matériaux du kit de châssis Arduino 4 roues, ou de les développer séparément si l'enseignant le souhaite.

Quelques calculs mathématiques seront nécessaires pour transformer le nombre d'événements de blocage comptés par le photogate en vitesse ou en distance. Le kit comprend un disque avec 22 trous et une roue d'un diamètre de $5,1 \pm 0,1$ cm. Il n'est pas difficile de calculer qu'une impulsion du photogate, c'est-à-dire une roue qui tourne de $1/22$ de tour complet, correspond à une distance $d = 0,72$ cm. En même temps, le photogate mesure et envoie un intervalle de temps t en millisecondes entre deux impulsions consécutives. La vitesse instantanée peut être calculée en divisant $0,72$ cm par cet intervalle de temps.

Les étapes suivantes peuvent être utilisées, que les étudiants travaillent dans des groupes différents ou dans un seul. Un seul groupe passerait en revue toutes les étapes l'une après l'autre, tandis que différents groupes se répartiraient les trois tâches.

<Programmation Arduino>

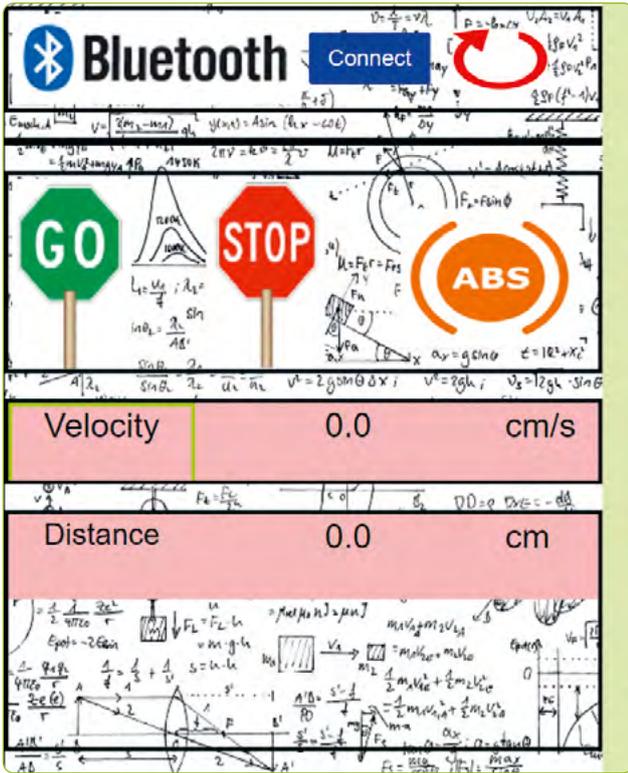
Le groupe de programmation de l'Arduino travaillera sur le codage selon l'approche suivante :

1. Définissez les actions et par conséquent les méthodes ou fonctions pour collecter et envoyer les données requises via Bluetooth.
2. Rédigez et testez chaque méthode séparément.
3. Mettez tout ensemble.

Les débutants pourraient commencer avec TinkerCad^[6], qui permet la conception et la simulation de circuits Arduino en ligne, évitant ainsi les problèmes de burn-out et de court-circuit lors du prototypage.

Les sections suivantes décrivent chaque partie plus en détail :

1. Les actions requises sont : démarrer et arrêter un moteur, serrer et desserrer les freins, mesurer la distance, mesurer la vitesse, envoyer et recevoir les données via Bluetooth.
2. La partie cruciale ici est d'écrire un code pour mesurer la vitesse et la distance parcourue au cours d'une même expérience. Les deux utilisent les impulsions d'un photogate et sont activées lorsque « 2 » est reçu de l'application téléphonique via Bluetooth :
 - ↳ Pour mesurer la distance, il y a un compteur qui commence à compter les impulsions envoyées à l'Arduino à partir d'une roue arrière tournant librement après que les freins ont été appliqués aux roues avant.
 - ↳ Les intervalles de temps entre les impulsions sont utilisés pour mesurer la vitesse instantanée. Une roue arrière tourne de $0,72$ cm en une impulsion, il faut donc la diviser par l'intervalle de temps entre les impulsions.
 - ↳ La fonctionnalité d'un système de freinage antiblocage (ABS) pourrait être mise en œuvre en serrant les freins et en les relâchant une fois pendant divers intervalles de temps compris entre 50 et 200 ms (optimisés expéri-



© 2 : Interface utilisateur de l'application

mentalement], ce qui, dans la plupart des cas, conduit à une distance d'arrêt réduite.

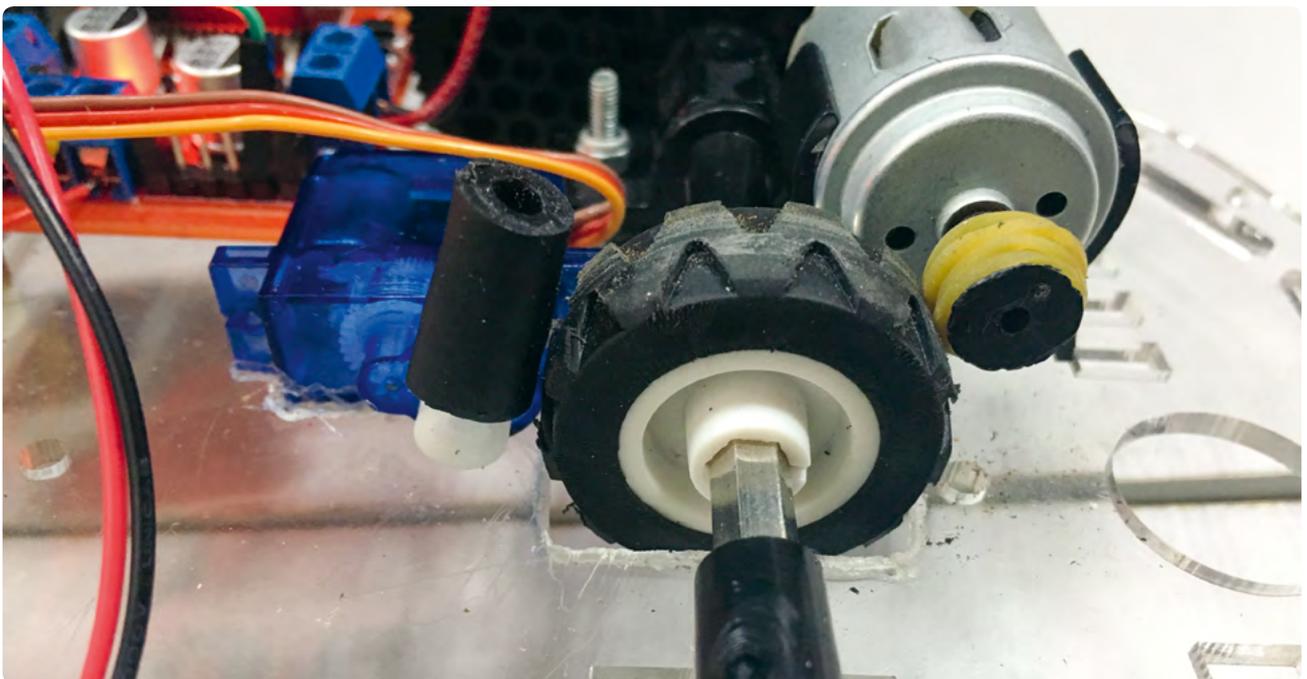
3. Lorsque vous créez un programme pour Arduino, vous devez vous assurer que tout se passe dans une seule grande boucle. Par conséquent, si le programme est interrompu à une étape particulière, cela affectera toutes les étapes suivantes.

4. L'exemple de code et les références à d'autres sources pour chacune de ces fonctionnalités sont disponibles en ligne^[7] mais, avec un peu d'aide de leurs enseignants, les élèves devraient être encouragés à essayer d'écrire leur propre code.

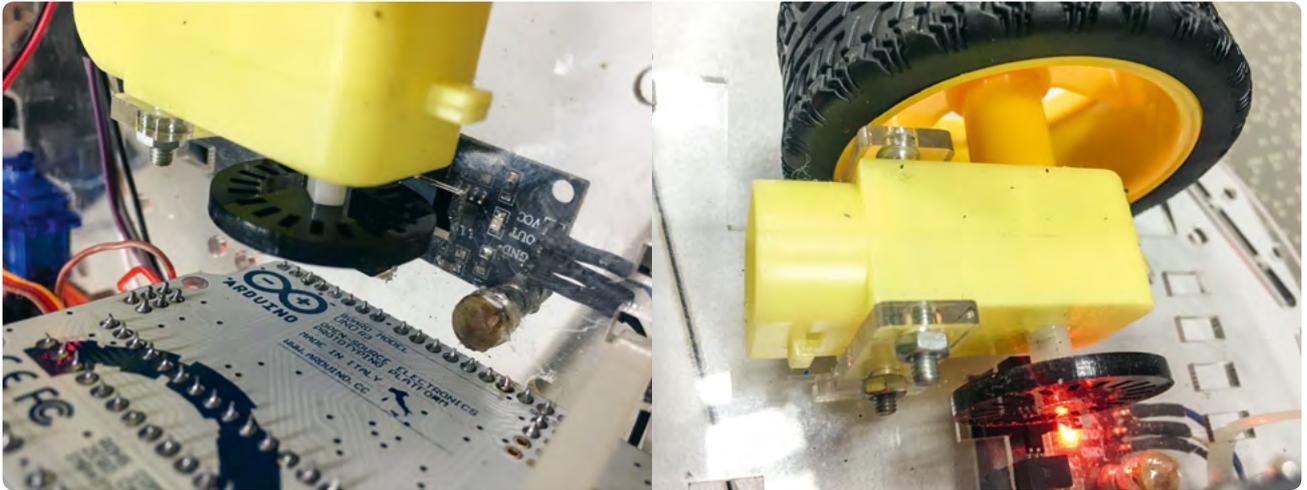
<Programmation Android>

Le groupe de programmation Android peut explorer Applinventor^[3] et réfléchir à des moyens d'afficher les données à l'écran (interface utilisateur, UI). Les élèves décideront où et comment disposer les boutons pour contrôler la voiture ainsi que les panneaux et les étiquettes pour afficher les données sur un écran [2]. Le code du programme Applinventor est fourni en ligne^[7] et possède les fonctionnalités suivantes :

1. Appuyer sur le bouton **START** envoie « 1 » à l'Arduino^[1] via Bluetooth et démarre le moteur dans la voiture.
2. Le fait d'appuyer sur le bouton **STOP** envoie « 2 » à l'Arduino via Bluetooth, qui arrête le moteur et applique ensuite les freins.
3. Le fait d'appuyer sur le bouton **ABS** envoie « 3 » à l'Arduino via Bluetooth, qui arrête le moteur et applique ensuite les freins à intervalles réguliers (simulant une fonction ABS).
4. Après avoir appuyé sur le bouton **STOP** ou le bouton **ABS**, les données reçues sur la vitesse instantanée avant l'arrêt et la distance parcourue par la voiture après avoir freiné, c'est-à-dire la distance d'arrêt, seront affichées sur deux panneaux correspondants avec respectivement les étiquettes « vitesse » et « distance ».



© 3 : Un gros plan du système de freinage



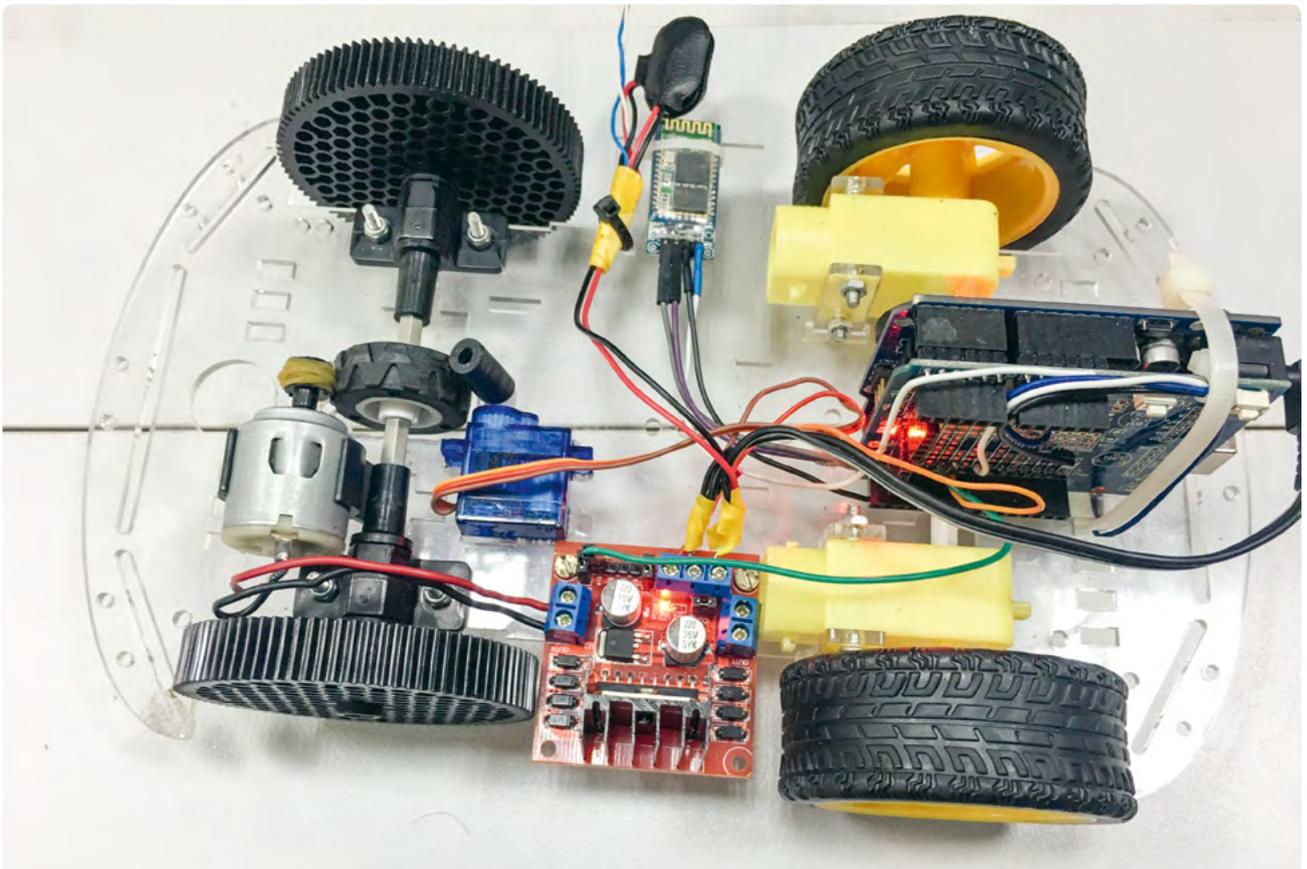
© 4 : Le photogate

5. Le bouton RESET envoie « 0 » à l'Arduino via Bluetooth, efface les données de vitesse et de distance, puis réinitialise l'Arduino.

Nous vous recommandons d'utiliser le code fourni comme référence pour les enseignants et de donner aux élèves la possibilité d'explorer AppInventor^[3] et d'écrire leur propre code basé sur les fonctionnalités ci-dessus.

<Construction de voitures>

Le groupe de construction automobile sera invité à réfléchir à des emplacements appropriés et à des moyens de fixer des composants tels que le moteur, le photogate, le servo, les batteries, le module Bluetooth, le bouclier moteur et, enfin, la carte Arduino elle-même. Lorsque le servo tourne, il est important qu'il pousse fermement une poignée avec un tube en caoutchouc sur le disque rotatif, qui est monté sur l'axe de la roue avant ([@3]).



© 5 : La voiture assemblée

Il peut fournir assez de force pour arrêter les roues immédiatement. La position du photogate est également importante. Veillez à ce qu'il compte correctement toutes les impulsions – le capteur Arduino photogate recommandé possède une LED intégrée qui clignote lorsque quelque chose pénètre dans l'espace photogate. Pour cette raison, veillez à ce que le photogate compte correctement les rotations lorsque les roues arrière tournent [④].

Veillez également à ce que les roues arrière tournent aussi librement que possible à tout moment. Rappelez-vous que nous calculons la vitesse et la distance en utilisant la rotation libre des roues arrière. Enfin, la voiture doit ressembler à l'exemple de la ⑤ une fois qu'elle a été construite.

Nous vous recommandons de conserver des lignes directrices détaillées à titre de référence et de donner aux élèves l'occasion de faire un remue-méninges et de mettre en œuvre leurs propres solutions.

<Conclusion>

Ce projet est une façon amusante pour les élèves d'apprendre des concepts fondamentaux de la physique comme la force de frottement cinétique et statique et, en même temps, une technologie relativement moderne comme l'ABS, où la physique, l'électronique, la programmation et la conception se réunissent pour explorer les facteurs qui influent sur la distance de freinage d'une voiture. Il est toujours difficile d'interpréter et d'analyser des données expérimentales réelles. Elle implique des concepts clés tels que l'incertitude, la validité, la reproductibilité et la visualisation. Le projet permet aux élèves d'expérimenter et de comprendre la force de friction dans une leçon pratique et stimulante.

<Activité de coopération>

Ce projet offre un grand potentiel de collaboration, puisque ses trois éléments indépendants – la conception de la voiture, le codage de l'Arduino^[1] et le codage avec AppInventor^[3] – pourraient être encore développés et améliorés. Tous les partenaires coopérants bénéficieraient de la contribution des autres parties à l'un ou l'autre de ces éléments.

Une autre option de coopération pourrait être une compétition entre les équipes scolaires pour savoir qui pourrait faire une voiture avec les mêmes roues et la même masse s'arrêter plus rapidement, à condition que la surface « route » et la vitesse avant l'arrêt soient les mêmes.

Merci beaucoup à nos collègues grecs : Astrinos Tsoutsoudoukakis pour ses suggestions importantes sur la physique impliquée dans ce projet et Georgios Georgoulakis

pour ses conseils de codage extrêmement utiles. Nous tenons également à remercier Jörg Gutschank pour ses nombreux commentaires et son soutien, qui ont rendu ce projet plus intéressant et reproductible.

<Références>

- [1] www.arduino.cc
- [2] www.arduino.cc/en/Guide/Environment
- [3] <http://appinventor.mit.edu>
- [4] <http://snap4arduino.rocks/>
- [5] <https://developers.google.com/blockly/>
- [6] www.tinkercad.com/circuits
- [7] Tous les autres documents sont disponibles à l'adresse www.science-on-stage.de/coding-materials.

Roulements sonores

<Auteur> Georgios Georgoulakis

<Auteur> Astrinos Tsoutsoudakis



<Info>

<Mots-clés> science fondamentale, acquisition de données, mouvement circulaire, ondes sonores, géométrie, trigonométrie, mouvement circulaire

<Disciplines> physique, mathématiques, informatique

<Niveau d'âge des élèves> 14–17 ans

<Matériel> Microcontrôleur Arduino^[1] ou similaire (avec les pilotes appropriés installés), microphone, buzzer à haut rendement, perceuse électrique avec sa base, matériaux pour l'installation de disques en bois

<Langage> Snap4Arduino^[2]

<Niveau de programmation> moyen

<Résumé>

Ce scénario pédagogique interdisciplinaire combine la physique et l'informatique. Il peut être utilisé soit dans un cours d'informatique, soit dans un cours de physique et il implique, avec le calcul d'autres grandeurs physiques de mouvement circulaire uniforme, le calcul de la vitesse linéaire de rotation selon deux méthodes différentes.

La première de ces méthodes détecte la fréquence à laquelle le signal d'un détecteur de proximité infrarouge est bloqué par une petite bande métallique fixée à un point particulier d'un disque rotatif, mesurant ainsi la période. La seconde méthode exploite le décalage Doppler d'une source sonore placée sur le disque.

<Introduction des concepts>

L'étude expérimentale des grandeurs physiques (période T , fréquence f , vitesse linéaire v et vitesse angulaire ω) du mouvement circulaire uniforme est basée sur les connaissances acquises dans les lycées grecs (âge des élèves : 14–17 ans) et dans les programmes des autres écoles secondaires européennes. Il faut également connaître l'effet Doppler. La fréquence et les grandeurs de la vitesse angulaire et linéaire sont obtenues en utilisant des formules bien connues :

$$f = \frac{1}{T}, \omega = \frac{2\pi}{T} \text{ et } v = \frac{2\pi r}{T}$$

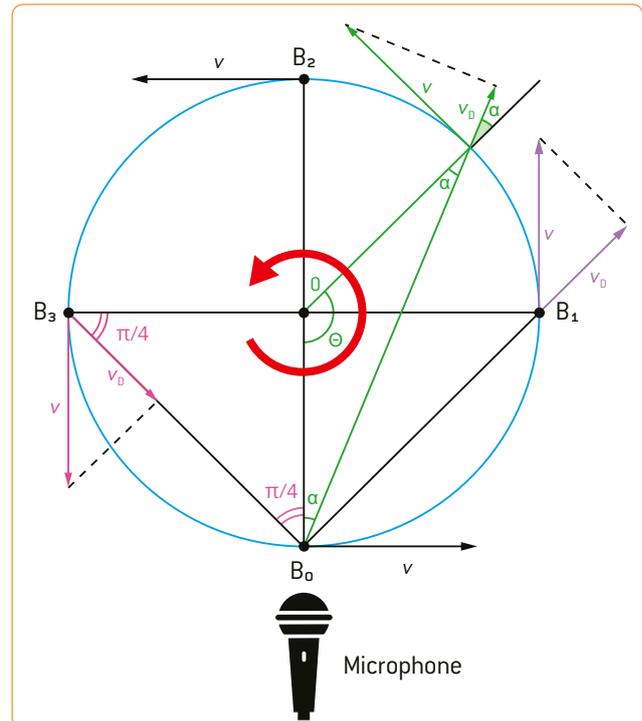
T provient de l'horloge interne du microprocesseur, c'est-à-dire du temps écoulé entre la détection entre deux signaux. Le rayon r représente la distance entre la bande métallique, ou le buzzer, et le centre du disque.

<Expérience sur l'effet Doppler>

L'effet Doppler est le changement de fréquence ou de longueur d'onde d'une onde si la source se déplace relativement à un observateur. Un exemple courant de ce phénomène au

quotidien est le changement de fréquence d'une sirène sur une ambulance en mouvement. Lorsque l'ambulance s'approche, la fréquence du son entendu est plus élevée que le son original ; cependant, lorsque l'ambulance s'éloigne, elle est plus faible. Le son reçu n'est le même qu'au moment où l'ambulance passe devant l'observateur.

Nous utilisons un buzzer sur un disque rotatif comme source et un microphone statique comme observateur dans notre expérience (voir [1]).



[1] : Le schéma de l'expérience

Lorsque le disque tourne dans le sens inverse des aiguilles d'une montre ([1]), la composante de la vitesse sur la corde, joignant le point B_0 au point où se trouve le buzzer B , augmente de zéro à maximum au point B_1 et diminue ensuite jusqu'à zéro au point B_2 . Cette composante de vitesse est la vitesse de recul réelle pour l'effet Doppler. Du point B_2 au point B_3 , la composante de vitesse, qui représente maintenant la vitesse d'approche, augmente en partant de zéro jusqu'à maximum au point B_3 , puis diminue de nouveau jusqu'à zéro au point B_0 .

Pour calculer la vitesse linéaire en appliquant la formule de décalage Doppler pour une source en mouvement au point B_3 et un observateur au repos. La vitesse linéaire reste constamment perpendiculaire au rayon du cercle et l'angle de $\frac{\pi}{4}$ est déterminé par les propriétés géométriques du triangle formé à angle droit et isocèle B_3OB_0 .

$$f = f_0 \cdot \left(\frac{v_s}{v_s - v_D} \right) \Rightarrow f = \frac{f_0 \cdot v_s}{v_s - v_D} \Rightarrow f \cdot v_s - f \cdot v_D = f_0 \cdot v_s$$

$$\Rightarrow f \cdot v_D = (f - f_0) \cdot v_s \Rightarrow v \cdot \cos\left(\frac{\pi}{4}\right) = \left(\frac{f - f_0}{f} \right) v_s$$

$$\Rightarrow v = \left(\frac{f - f_0}{f} \right) \frac{v_s}{\cos\left(\frac{\pi}{4}\right)}$$

f : fréquence mesurée

f_0 : fréquence émise

v : vitesse

v_s : vitesse du son

v_D : vitesse d'inclinaison/vitesse d'approche

Comme la mise en œuvre d'une transformation de Fourier rapide pour extraire le contenu en fréquence du son produit dépasse largement les capacités de codage des élèves, un logiciel de montage audio gratuit comme Audacity^[3] fournira un fichier texte adéquat contenant toutes les données nécessaires.

<Pour aller plus loin>

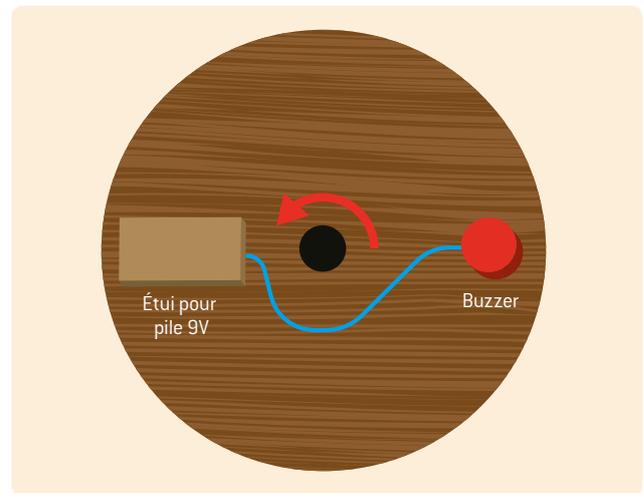
Une autre méthode qui utilise un système de tube Pitot et un capteur de pression différentielle a été délibérément exclue de cette unité d'enseignement pour simplifier les choses, mais toutes les informations requises sont disponibles en ligne^[4]. Le matériel en ligne offre une description détaillée du montage expérimental, des idées de construction alternatives, ainsi qu'une documentation théorique et une analyse étape par étape des procédures utilisées.

<Ce que font les élèves/enseignants>

Dans la section physique de l'unité d'enseignement, les élèves mesurent les grandeurs physiques des mouvements circulaires à différents rayons et explorent l'effet Doppler. Cependant, ils conçoivent et assemblent d'abord un montage expérimental simple.

<La mise en place du disque en bois>

Les élèves construisent l'installation composée d'un disque en bois entraîné par une perceuse électrique et muni d'un avertisseur sonore relié à une pile 9V. Un capteur de proximité infrarouge indépendant, mais très proche, alimente le microcontrôleur d'un signal pour chaque rotation complète, tandis qu'un microphone bon marché enregistre le son qui est pro-



© 3 : Disque vu d'en haut



© 2 : Montage expérimental de base

duit. Le décalage Doppler doit idéalement être audible pour la vitesse de rotation choisie, qui doit être maintenue basse pour des raisons de sécurité. [ⓐ2 & 3]

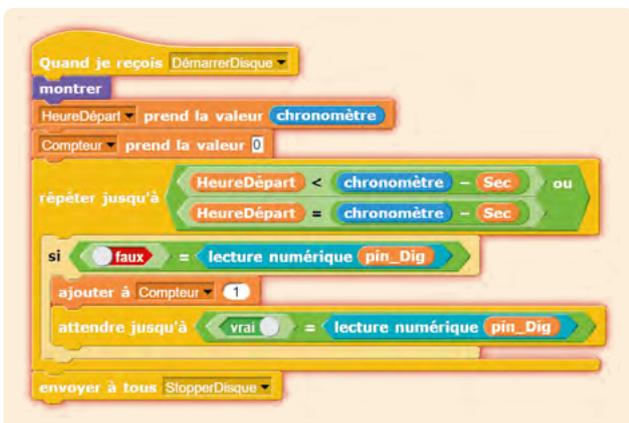
Une interface attrayante et conviviale a été développée, comme le montre la ⓐ4, pour saisir tous les paramètres nécessaires et extraire les valeurs calculées.



ⓐ 4 : L'interface de l'expérience

Les étudiants doivent posséder des compétences de base en programmation et une certaine expérience de l'utilisation des langages de programmation par blocs (tels que Scratch ou Snap!). Nous donnons aux étudiants un modèle de base pour travailler leur code afin de nous assurer qu'ils se concentrent sur les objectifs de l'unité d'enseignement et non sur l'interface utilisateur et l'apparence du programme.

C'est pourquoi nous fournissons le modèle de base en Snap!^[5], un fichier xml du projet et un document de travail pour donner aux élèves des instructions de base sur le modèle de formulaire et décrire ce que l'on attend d'eux. Le modèle et les affectations sont disponibles en ligne.^[4]

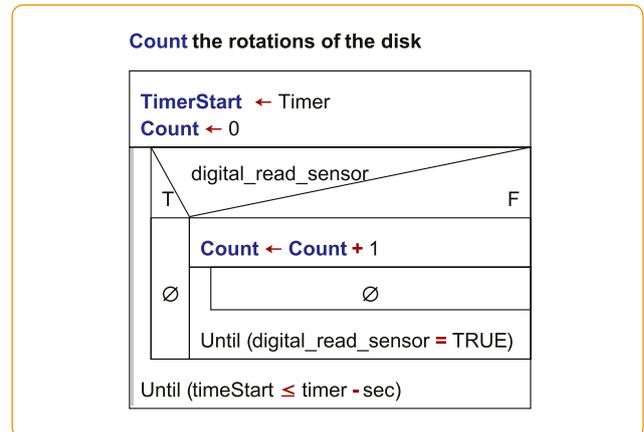


ⓐ 5 : Le calcul de la période de rotation

Les étudiants vérifient et valident les données acquises, se connectent et communiquent avec les dispositifs externes, re-

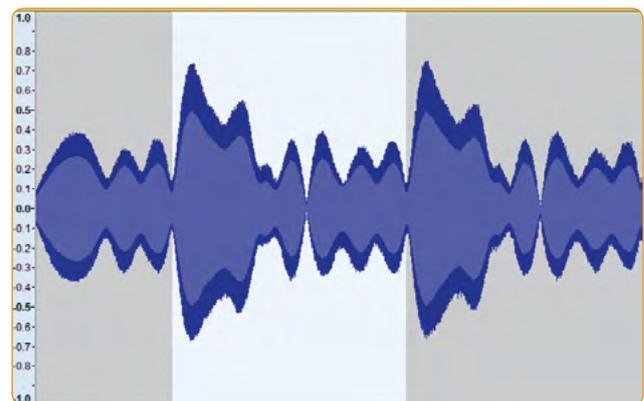
çoivent, traitent les données des capteurs et écrivent un algorithme de recherche en série simple.

Le programme fini est également disponible pour l'enseignant et peut être téléchargé comme fichier de référence^[4]. La ⓐ5 contient un exemple de capture d'écran de l'environnement de programmation Snap4Arduino^[2]. La ⓐ6 le diagramme Nassi-Shneiderman correspondant.



ⓐ 6 : Diagramme Nassi-Shneiderman pour la période

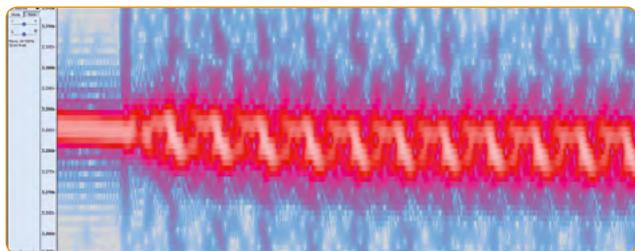
Comme mentionné précédemment, nous vous recommandons d'utiliser un logiciel d'édition audio gratuit comme Audacity^[3] pour extraire le contenu en fréquence du son produit. Le logiciel fournit un fichier texte adéquat avec toutes les données nécessaires pour que les élèves apprennent à traiter un signal sonore à l'aide d'un logiciel spécialisé.



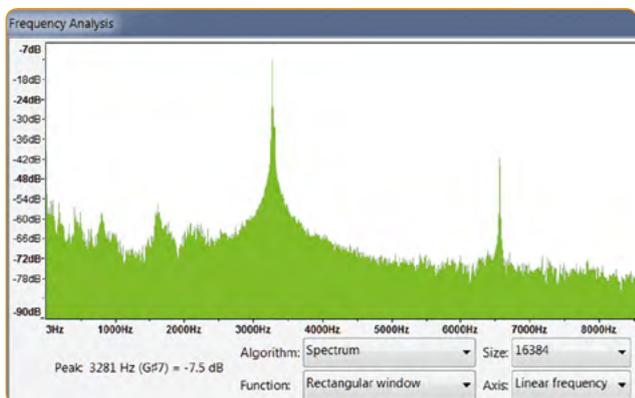
ⓐ 7 : Une forme d'onde sonore enregistrée

L'importation du signal et le traitement essentiel sont illustrés aux ⓐ7-9 que la ⓐ10 représente une partie de l'exportation finale des données. Afin d'éviter une analyse approfondie et de permettre une meilleure compréhension de la part des étudiants, une hypothèse approximative doit être faite ici. La fréquence surlignée en jaune, qui possède le niveau acoustique maximum, est la fréquence du buzzer au repos ou celle mesurée aux points B₀ et B₂, tandis que le bleu et le vert représentent les pics les plus proches de cette fréquence primaire [ⓐ10].

Cependant, le code proposé ne recherche que la valeur en vert surligné, mais pour une meilleure précision, il peut être facilement modifié pour trouver les deux. Pour accélérer les choses, la section des données peut également être réduite à seulement 50–100 valeurs au-dessus et au-dessous de la fréquence du niveau maximum.

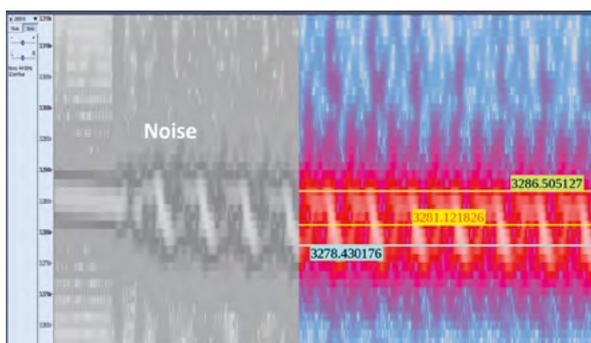


8 : Un spectrogramme montrant le décalage Doppler



9 : Analyse de fréquence par transformée de Fourier rapide

Fréquence [Hz]	Niveau [dB]
3273.046875	-27.597595
3275.738525	-22.331339
3278.430176	-12.437067
3281.121826	-7.5547090
3283.813477	-10.041918
3286.505127	-9.7750780
3289.196777	-16.848948
3291.888428	-26.916197



10 : Données exportées

Le traitement des données du spectre acoustique est illustré aux 11 et 12. des informations détaillées, par exemple sur les variables utilisées, sont disponibles en ligne. [4]

```

Quand je suis cliqué
  dire N'oubliez pas le clic droit sur la chaîne et charger les données pendant 2 sec.
  FR_LISTE prend la valeur découpe chaîne entre les espaces blancs
  F.L. prend la valeur
  garder les items tels que < 0 de FR_LISTE
  A.L. prend la valeur
  garder les items tels que < 0 de FR_LISTE
  max prend la valeur 1000
  j prend la valeur 2
  répéter jusqu'à j > longueur de A.L.
    Id_num prend la valeur
    regroupe
      élément 1 de découpe élément j de A.L. entre les
      élément 2 de découpe élément j de A.L. entre les
    si Id_num > max
      max prend la valeur Id_num
      max2 prend la valeur j
    ajouter à j 1
    j prend la valeur max + 1
    max2 prend la valeur -1000
  répéter jusqu'à j > longueur de A.L.
    Id_num prend la valeur
    regroupe
      élément 1 de découpe élément j de A.L. entre les
      élément 2 de découpe élément j de A.L. entre les
    si Id_num > max2
      max2 prend la valeur Id_num
      max2 prend la valeur j
    ajouter à j 1
  
```

11 : La partie Doppler de l'expérience[4]

<Algorithme à utiliser dans d'autres langues>

Le gabarit de base permettra un transfert facile vers n'importe quel autre langage de programmation à condition qu'il y ait une bibliothèque de base pour la communication avec le microcontrôleur. Par conséquent, le choix du microcontrôleur n'aura pas d'effet significatif sur le projet.

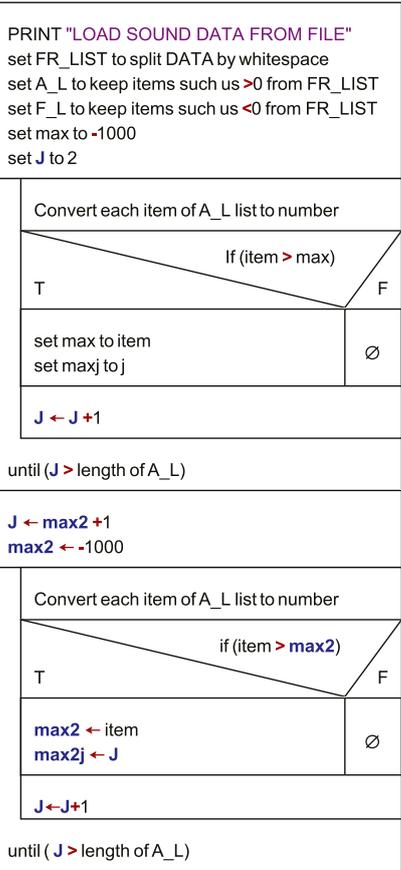
<Conclusion>

Il s'agit d'un projet peu coûteux, facile à monter et à utiliser ; il sera, espérons-le, intéressant et stimulant pour les étudiants.

<Activité de coopération>

La plateforme Science on Stage a pour but d'échanger des idées pédagogiques et de mettre en œuvre des approches pédagogiques innovantes. Le co-enseignement avec Ilia Mestvirishvili et David Shapakidze, une superbe équipe partenaire géorgienne, peut s'avérer difficile en raison de la distance et des problèmes de calendrier, mais cela nous a déjà permis de développer de nouvelles techniques. Malgré l'absence commune de formation en éducation spécialisée, il peut être bon de modifier le projet pour le rendre accessible à tous les élèves.

Sound Data Processing



© 12: Diagramme Nassi-Shneiderman pour le traitement des données sonores

<Références>

- [1] www.arduino.cc
 - [2] <http://snap4arduino.rocks>
 - [3] www.audacityteam.org
 - [4] Tous les autres documents sont disponibles à l'adresse www.science-on-stage.de/coding-materials.
 - [5] <https://snap.berkeley.edu>
- ↳ www.physicsclassroom.com/mmedia/circmot/ucm.cfm
- ↳ <https://education.pasco.com/epub/PhysicsNGSS/BookInd-904.html>
- ↳ <http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/dopp.html>
- ↳ http://newton.phys.uaic.ro/data/pdf/Doppler_experiment.pdf
- ↳ <https://manual.audacityteam.org/man/tutorials.html>
- [tout décembre 2018]

Moteur physique

<Auteur> Mihaela Irina Giurgea

<Auteur> Corina Lavinia Toma



<Info>

<Mots-clés> animation, sprite, blocs, boucles, graphes, lois gravitationnelles, collisions, chute libre, force de friction, lancer oblique, mouvement, momentum, opérateurs, variables

<Disciplines> informatique, physique, mathématiques, TIC

<Niveau d'âge des élèves> 14–16 ans

<Matériel> ordinateur

<Langage> Scratch^[1]

<Niveau de programmation> facile, moyen

<Résumé>

Que penseriez-vous si nous vous disions que vos élèves pourraient apprendre deux matières apparemment très différentes, la physique et l'informatique, plus facilement et en même temps ? Dans cette unité, le « moteur », c'est-à-dire l'environnement de programmation Scratch^[1], est l'outil magique qui aidera les élèves à créer des applications intéressantes sur les phénomènes naturels quotidiens pour mieux comprendre les lois de la physique et améliorer ainsi leurs compétences en programmation.

<Introduction des concepts>

Pourquoi avons-nous utilisé Scratch^[1] ? Scratch est un environnement de programmation visuel qui anime des sprites (objets graphiques) à l'aide de blocs sur l'écran de l'ordinateur et aide les étudiants à créer des applications plus facilement qu'avec des environnements de programmation classiques (C++, Java, etc.). De plus, notre « moteur » nous aide à enseigner deux matières que les étudiants considèrent difficiles : la physique et l'informatique.

Les étudiants qui ont participé à l'élaboration de cette unité d'enseignement avaient déjà une certaine expérience du codage, ce qui leur a permis d'apprendre à utiliser Scratch. Ils l'utilisent aussi bien dans leurs cours quotidiens que dans leurs cours optionnels d'informatique. En outre, les connaissances en physique requises font partie du programme d'études standard.

<Ce que font les élèves/enseignants>

L'unité porte sur l'alternance de séquences d'apprentissage impliquant le codage et la physique.

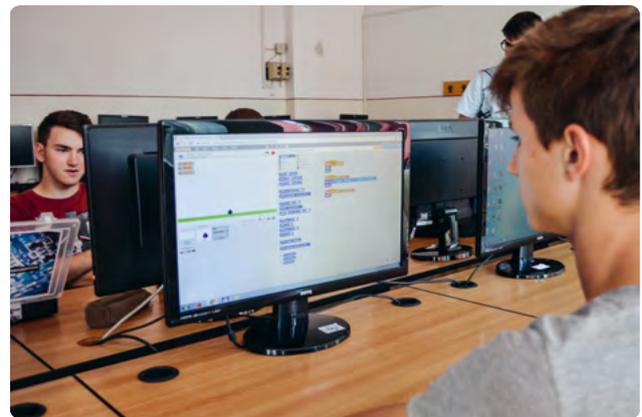
Tout d'abord, le professeur d'informatique a présenté les bases de la réalisation d'un projet dans Scratch^[1]. Les élèves se sont familiarisés avec plusieurs mots clés liés à l'environnement

Scratch : scène, sprites, costumes et mouvement. Vous pouvez suivre les instructions de la première application enseignée dans Scratch, une application sans formules physiques. ^[2]

Les élèves devaient comprendre les interactions entre les sprites et leur synchronisation ainsi que le fonctionnement d'un système de coordonnées. Vous pouvez trouver un tutoriel complet pour Scratch en ligne. ^[3]

Pour la partie physique de cette unité, les étudiants ont appliqué les principes théoriques qui sous-tendent les phénomènes du monde qui les entoure. Pour cette raison, le professeur de physique a suggéré une grande variété de sujets^[4], sur lesquels les élèves ont ensuite concerté : les formules nécessaires, l'animation possible, le design, etc.

Au début, les élèves travaillaient individuellement avec l'aide de leurs professeurs. Lorsque les projets avaient besoin d'être améliorés, les élèves étaient guidés par les enseignants et leurs camarades de classe. ^[1]

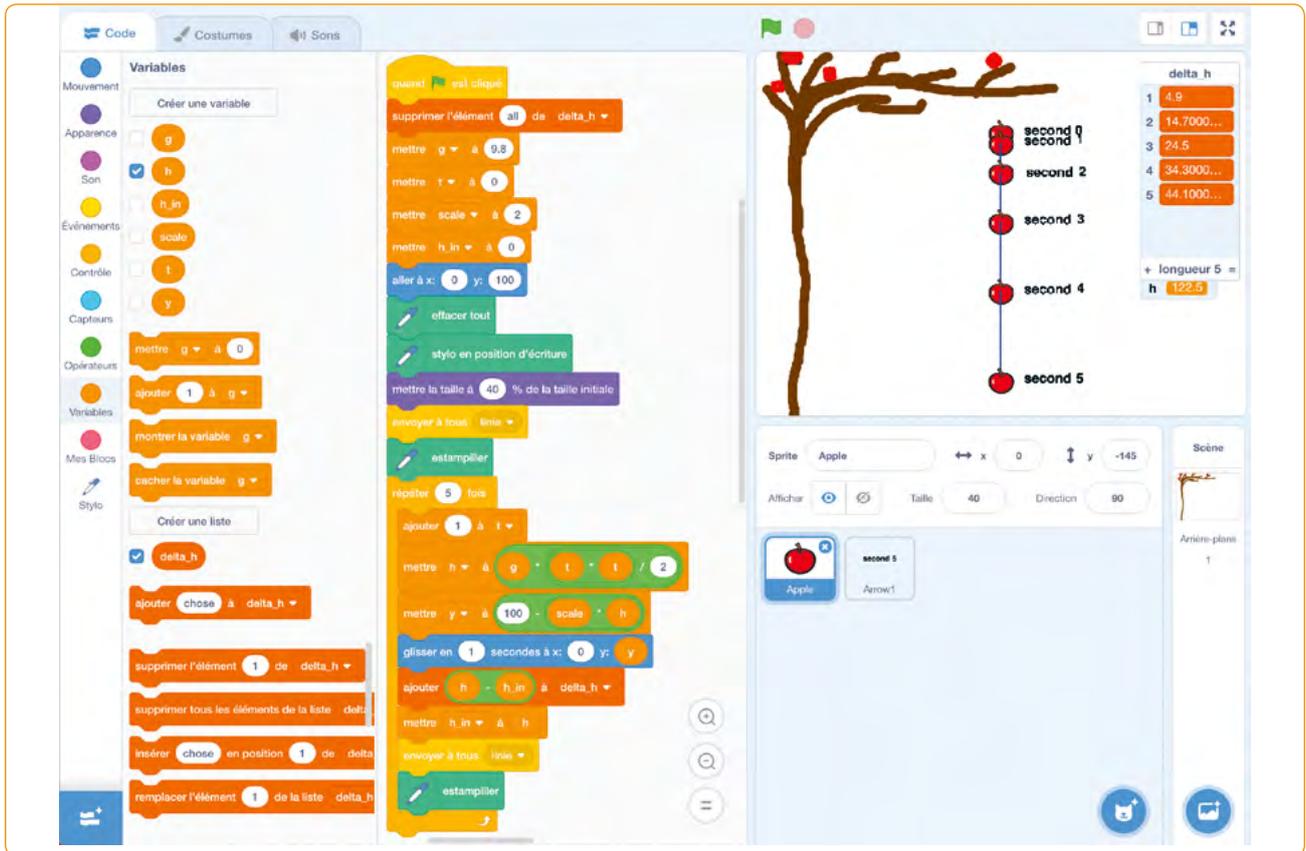


© 1 : Travail individuel

Ensuite, chacun a présenté son projet à la classe et a reçu les commentaires de ses pairs. Cela a permis aux élèves de déterminer plus facilement quelles parties des projets devaient être améliorées : le codage ou la physique.

À la fin de notre projet, les élèves plus âgés sont devenus enseignants pour les plus jeunes (12–13 ans) en présentant des applications appropriées et en les testant pendant les cours de physique. Les élèves plus âgés ont également reçu des suggestions des élèves plus jeunes. Les meilleures de ces simulations étudiantes sont disponibles sur la plateforme Scratch. ^[2]

Les sections suivantes contiennent des exemples de la façon dont nous avons abordé la physique et les parties de programmation.



Ⓢ 2 : Problème de chute libre

<Application 1: Problème de chute libre (pomme de Newton)>

Tous les élèves ont entendu parler de la chute libre de cet objet historique : La pomme de Newton. L'application de la Ⓢ2 s'inspire d'un problème classique : quelle est la distance parcourue chaque seconde par la pomme de Newton en chute libre ?

Théorie de la physique

Nous considérons un mouvement linéaire avec une accélération gravitationnelle constante $g = 9,8 \frac{m}{s^2}$.

Après un temps t , la distance parcourue $h(t)$ de la pomme est $h(t) = \frac{gt^2}{2}$.

Le point initial $h(0)$ est fixé sur la branche de l'arbre dont la pomme s'est détachée. Ensuite, nous calculons la distance parcourue sur une période plus longue,

$$t + \Delta t : h(t + \Delta t) = \frac{g(t + \Delta t)^2}{2}$$

La formule générale pour la distance $\Delta h(t)$, la distance parcourue par la pomme pendant Δt , est :

$$\Delta h(t) = h(t + \Delta t) - h(t) = \frac{g(2t\Delta t + \Delta t^2)}{2}$$

Ensuite, nous utilisons les données du problème spécifique pour personnaliser la formule générale ; dans ce cas, $1s$ pour Δt . Pour la première seconde, $t = t_{in} = 0$ donne $\Delta h_1 = 4,9 m$, pour la seconde suivante, $t = 1s$ donne $\Delta h_2 = 3 \cdot 4,9 m = 14,7 m$ etc. Par induction mathématique, on peut calculer la distance parcourue à la n ème seconde en considérant $t = (n - 1) s$:

$$\Delta h_n = \frac{9,8(2n - 1)}{2} m.$$

Ensuite, les élèves peuvent calculer, et aussi voir dans notre animation, que la distance parcourue augmente de la même valeur chaque seconde, $9,8 m$.

Comment coder cela ?

Variables utilisées :

g : accélération gravitationnelle

t : un compteur pendant quelques secondes (avec des valeurs : $0, 1, 2, 3, 4, 5$)

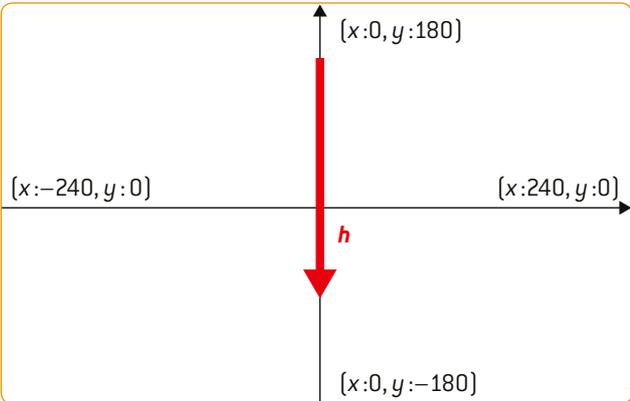
h : la distance parcourue après t secondes

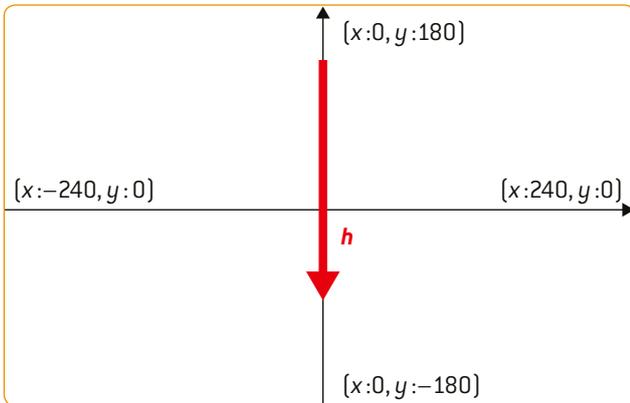
h_{in} : la position initiale de la pomme

$delta_h$: une liste (tableau) avec toutes les distances parcourues en une seconde

y : la coordonnée y de la pomme

Observation : dans cette application, la coordonnée x reste constante = 0 , ce qui vous permet de déplacer plus facilement la trajectoire vers la gauche ou la droite sur la scène.

Au début, la pomme se trouve au point avec les coordonnées (0,180). Le point initial et la direction de la distance parcourue $h(t)$ sont indiqués sur la 3.



 3 : Orientation dans le système de coordonnées

En utilisant une boucle 5 fois, nous avons recalculé la distance parcourue par la pomme après chaque seconde, en calculant la nouvelle coordonnée y et en considérant les caractéristiques de l'écran. Voir 4 pour l'algorithme de codage.

Défi

Les élèves modifient Δt et le temps de parcours t (notre pom-pomier devrait être très haut ou déplacer le problème sur une autre planète avec sa propre accélération gravitationnelle). Ils pourraient ajouter la force de frottement de l'air et envisager une accélération gravitationnelle variable en faisant tomber la pomme d'un ballon météorologique à une altitude supérieure.

Free_fall

```

delta_h: array of real
g ← 9.8
t ← 0
h_in ← 0
scale ← 2
go to (0,100)
clear()
pen(down)
stamp()

while (t < 5)
    t ← t + 1
    h ← g * t * t / 2
    y ← 100 - scale * h
    glide (0, y)
    add (delta_h, h - h_in)
    h_in ← h
    broadcast(linie)
    stamp()
    
```

 4 : Problème de chute libre

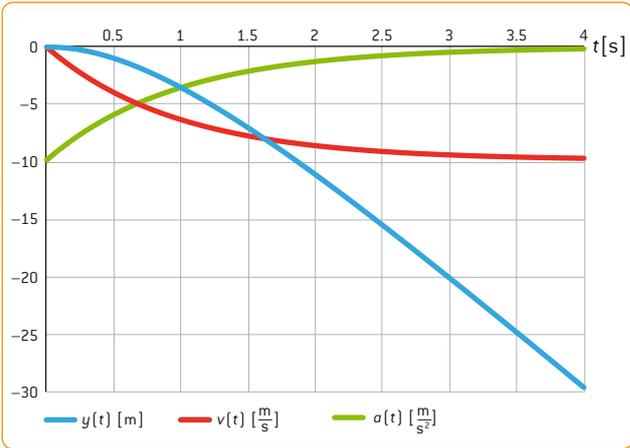
<Application 2 : Chute de gouttes d'eau>

Les jours de pluie, tout le monde peut observer la chute des gouttes d'eau. Les étudiants ont analysé le mouvement li-

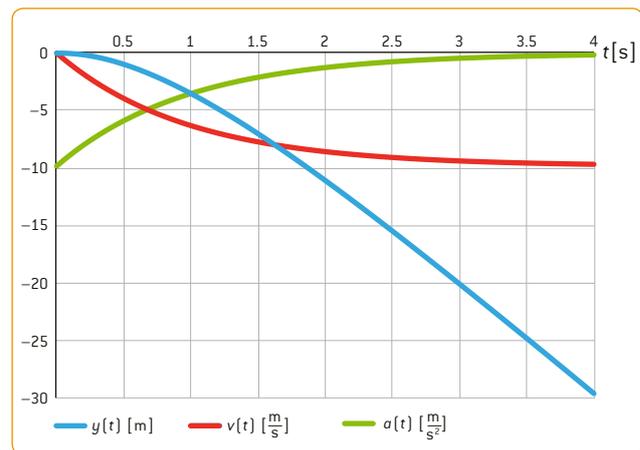
néaire d'une goutte avec notre simulation. Au début, ils ont vu que la chute de la goutte d'eau est accélérée, mais avec une accélération décroissante. Au bout d'un certain temps, la vitesse de la goutte a atteint sa limite, la vitesse finale v_t , lorsque l'accélération a atteint zéro. Puis la goutte d'eau a continué à se déplacer à cette vitesse constante. Comment pouvez-vous expliquer cela ?

Théorie de la physique

Dans la partie accélérée du mouvement, deux forces agissent en sens inverse sur la goutte : la force gravitationnelle $G = mg$ (m : masse de la goutte, g : accélération gravitationnelle) et la force de friction $F_f = kv$ (k : constante de proportionnalité, v : vitesse instantanée). L'accélération de la chute devient : $a = g - \frac{k}{m} v$.

Dans notre simulation, nous considérons une grosse goutte qui mesure environ 5 mm de diamètre avec la vitesse terminale $v_t = 9,8 \frac{m}{s}$. Dans ce cas, la constante $\frac{k}{m} = \frac{1}{s}$. L'accélération diminue lorsque la vitesse augmente (voir 5). Les valeurs initiales sont $a = 9,8 \frac{m}{s^2}$, $v = 0$ et $y = 0$.

Nous utilisons un petit programme en C++^[4] pour calculer l'accélération et la vitesse instantanée, où nous considérons l'accélération et la constante de vitesse pour de très petits intervalles de temps Δt (comme 0,05 s). Dans ce cas, la vitesse augmente avec $\Delta v = a\Delta t$ et la distance parcourue avec $\Delta y = v\Delta t$ pour chaque Δt choisi (méthode pas à pas).



 5 : Relation entre la distance parcourue, la vitesse et l'accélération

Comment coder cela ?

1. Peindre un sprite goutte d'eau.
2. Tracez une ligne verte horizontale au bas de la toile de fond.
1. Faites un sprite avec le message « acceleration=0 », qui apparaît lorsque l'accélération a une valeur d'environ 0.
2. Écrivez le code pour le sprite de la goutte. La baisse commence au point $(0, y_{init})$. En utilisant une boucle, on recalcule $a(t)$, $v(t)$, $y(t)$. Nous utilisons la distance atteinte par

la goutte après chaque Δt , en calculant la nouvelle coordonnée y et en tenant compte des caractéristiques de l'écran. L'accélération diminue et quand elle est d'environ 0, la boucle est terminée. Ensuite, le sprite tombe à une vitesse constante jusqu'à ce qu'il touche la ligne verte du fond.

📄6 : fournit une explication claire du code.

```

Drop
g ← 9.8
kOverM ← 1
deltaT ← 0.05
eps ← 0.17
v ← 0
t ← 0
y ← 0
a ← -g * kOverM * v
vf ← -9.8

(abs(a) > eps)

t ← t + deltaT
y ← y + deltaT * v
v ← v + deltaT * a
a ← -g * kOverM * v

y ← y + deltaT * vf

until (touch (ground))
    
```

📄6 : Chute de gouttes d'eau

Défi

Les élèves peuvent améliorer cette application en ajoutant une variable pour la masse de la goutte d'eau (le diamètre de la goutte d'eau peut généralement prendre des valeurs de 1 mm à 5 mm)^[6] et un autre type de force de friction : $F_f = \frac{kv^2}{2}$.

Les élèves pourraient également ajouter d'autres gouttes de différentes masses et comparer leur chute.

<Application 3 : Collision élastique>

Il y a beaucoup d'exemples de corps qui entrent en collision autour de nous. Ces collisions sont compliquées, mais nous considérons les collisions élastiques qui sont applicables dans la vie réelle pour les boules de billard ou d'acier, ou en théorie pour les collisions de molécules lorsque les étudiants étudient le modèle de gaz idéal.

Théorie de la physique

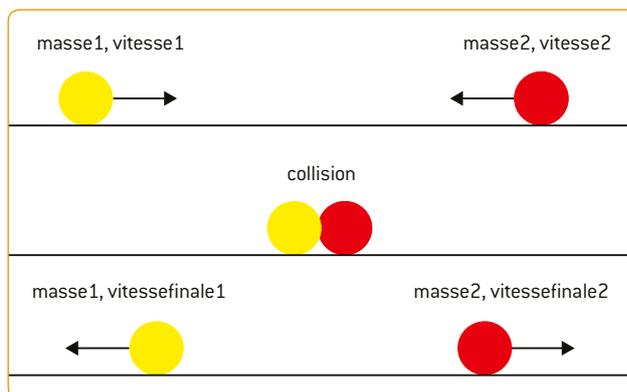
Le momentum linéaire et l'énergie cinétique sont conservés pour deux boules avec la masse m_1 et m_2 , les vitesses initiales (\vec{v}_1) et (\vec{v}_2) et les vitesses finales (\vec{v}_{1f}) et (\vec{v}_{2f}). [7]

$$m_1 \vec{v}_1 + m_2 \vec{v}_2 = m_1 \vec{v}_{1f} + m_2 \vec{v}_{2f} \text{ et}$$

$$\frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 = \frac{1}{2} m_1 v_{1f}^2 + \frac{1}{2} m_2 v_{2f}^2$$

Si tous les mouvements ont lieu le long de la même ligne (déplacement sur l'axe des x), on peut utiliser les signes + ou - pour indiquer les directions. La notation vectorielle n'est pas nécessaire dans le cas d'une collision en ligne droite, et les vitesses finales peuvent être calculées avec les équations suivantes :

$$v_{1f} = 2 \frac{m_1 v_1 + m_2 v_2}{m_1 + m_2} - v_1 \text{ et } v_{2f} = 2 \frac{m_1 v_1 + m_2 v_2}{m_1 + m_2} - v_2.$$

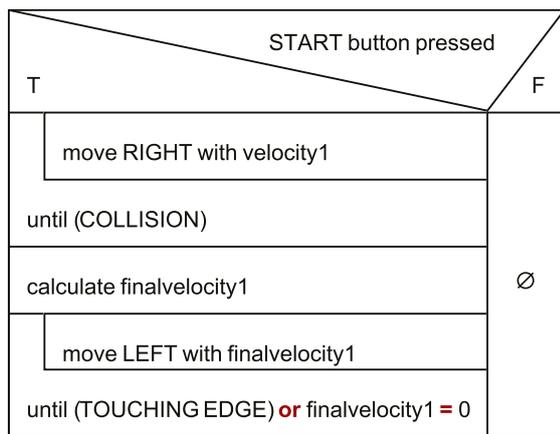


📄7 : Collision élastique

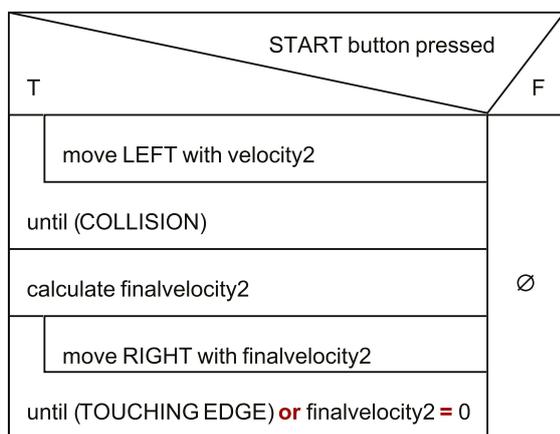
Comment coder cela ?

1. Choisissez deux sprites pour les boules (Ball1 et Ball2) et un sprite pour le bouton START (Start sprite).
2. Utiliser des variables : *masse1*, *masse2*, *vitesse1*, *vitesse2* (la masse et la vitesse initiale) pour chaque objet. Rendre les curseurs de variables visibles et définir les valeurs minimum et maximum pour eux.
3. Saisissez la masse et les vitesses initiales pour chaque objet.
4. Appuyez sur la touche START. À ce moment, le sprite diffuse un message. Lorsqu'ils reçoivent le message, chaque boule se déplace vers l'autre en utilisant la formule *distance = vitesse × temps*.
5. Calculez les vitesses finales des boules et utilisez-les pour déplacer les boules dans la bonne direction jusqu'à ce qu'une boule touche le bord et quitte la scène ou reste à sa place, car sa nouvelle vitesse est 0.

Les 📄8 et 9 donnent un aperçu clair de la façon dont les deux boules sont animées dans Scratch^[4].

Elastic_Collision_Ball1

© 8 : Collision élastique pour Ball1

Elastic_Collision_Ball2

© 9 : Collision élastique pour Ball2

Deux exemples d'utilisation de cette application :

1. Choisissez une vitesse 0 et une masse égale pour les boules ; après cette collision, vous constaterez que la boule en mouvement s'arrête et que l'autre boulette se déplace avec la même vitesse que la première boule avant l'impact.
2. Les boules ont des vitesses différentes et une masse égale ; après la collision, vous verrez que les objets prennent la valeur de l'autre pour la vitesse.

Dans les deux exemples, les billes échangent leur momentum.

Défi

Les élèves pourraient changer la taille des balles directement proportionnelle à leur masse ; ils pourraient faire une application de collision élastique bidimensionnelle (simulation pour l'effet Compton) ou programmer une simulation pour la collision d'une balle avec un mur (loi de réflexion en mécanique).

Vous pourriez poursuivre l'étude avec une autre application, c'est-à-dire une collision inélastique.^[4]

<Conclusion>**<Pour les étudiants>****Avantages**

Les étudiants ont appris la théorie de la physique d'une manière plus agréable et ont été capables de mieux comprendre les phénomènes naturels en utilisant des simulations dans Scratch. Ils ont en même temps approfondi leurs connaissances en informatique et en physique.

Inconvénients

Les étudiants travaillaient seuls et davantage à la maison. Ils ont reçu des commentaires à l'école.

<Pour les enseignants>**Avantages**

Nous avons constaté un réel intérêt pour la création d'une application originale et l'apprentissage plus que dans les cours classiques.

Inconvénients

Il nous a été difficile de coordonner l'ensemble de la classe en raison de la grande variété des sujets de physique et des boques très spécifiques dans chaque application. Nous pensons qu'il serait préférable de donner à tous les élèves le même sujet et de les encourager à l'améliorer au mieux de leurs différents niveaux de compétence.

<Activité de coopération>

Des élèves de différentes écoles et de différents pays ont pu résoudre les défis des projets et en créer de nouveaux avec d'autres idées en rapport avec le sujet original. Toutes ces applications pourraient être placées au même endroit sur la plateforme Scratch, puis un concours pourrait être organisé pour déterminer les meilleures d'entre elles. Les enseignants doivent également tenir compte de la complexité du codage et de la physique lorsqu'ils évaluent le travail de leurs élèves.

<Références>

- [1] <https://scratch.mit.edu/>
- [2] Tous les autres documents sont disponibles à l'adresse www.science-on-stage.de/coding-materials.
- [3] <https://en.scratch-wiki.info/>
- [4] https://scratch.mit.edu/users/SonS_Coding
- [5] <http://hypertextbook.com/facts/2007/EvanKaplan.shtml> [29/11/2018]
- [6] <https://journals.ametsoc.org/doi/pdf/10.1175/1520-0450%281969%29008%3C0249%3ATVORA%3E2.0.CO%3B2> [29/11/2018]

BMS - Boîte Magique pour les sciences

<Auteur> Luc Ivacca

<Auteur> Marco Nicolini



<Info>

<Mots-clés> microcontrôleur, transducteur, capteur, actionneur, signal, grandeur physique, boucle, instruction, séquentiel, processus, calibration, entrée, sortie, lecture, écriture, analogique, numérique, linéarité, conversion, plaque d'expérimentation, broche, soudure, interface homme-machine

<Disciplines> physique, électronique, mathématiques, TIC, biologie

<Niveau d'âge des élèves> 14–18 ans

<Matériel> Arduino UNO^[1] avec Arduino DUE^[2] et/ou TI-Nspire CX CAS avec TI-Innovator Hub

<Langage> C++ [en utilisant Arduino IDE^[3]] et/ou TI-Basic

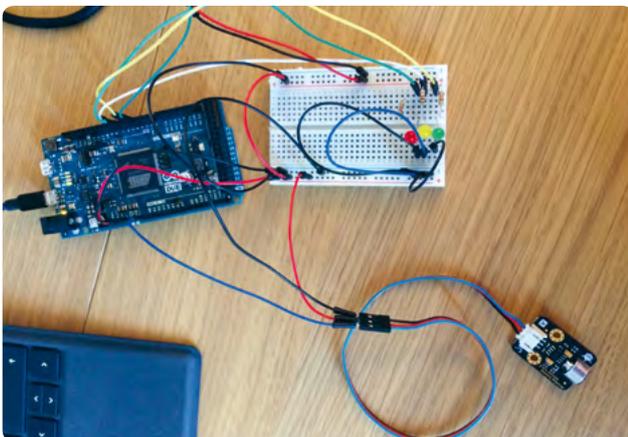
<Niveau de programmation> moyen, avec une section audio pour les étudiants avancés

Une liste des abréviations, termes spéciaux et acronymes utilisés dans cette unité est disponible en ligne.^[4]

<Résumé>

Les étudiants apprennent à programmer pour un environnement matériel et logiciel construit par eux-mêmes (basé sur Arduino) et pour un ordinateur de poche prêt à l'emploi (calculatrice TI-Nspire CX CAS avec son extension, le TI-Innovator Hub). Les deux sont utilisés comme dispositifs pour la collecte, la conversion et la transduction de données de capteurs pour manipuler, lire, convertir et actionner facilement des grandeurs physiques.

En utilisant la plate-forme Arduino, les étudiants codent avec des capteurs qui acquièrent des grandeurs physiques comme signaux d'entrée, et des actionneurs qui réagissent à l'acquisition et produisent un signal de sortie comme grandeur physique après qu'un microcontrôleur ait traité le signal détecté pour régler la bonne sortie (voir ①).



① : Carte Arduino

Le TI-Innovator Hub est un boîtier « prêt à l'emploi » qui permet aux étudiants d'apprendre les bases de la programmation. Il doit être branché à un ordinateur TI-Nspire CX. Il dispose d'une bonne interface d'E/S, qui comprend un capteur de luminosité, deux LEDs et un buzzer intégré, qui produit un son d'une fréquence donnée (voir ②).



② : TI-Nspire et TI-Innovator Hub

<Introduction des concepts>

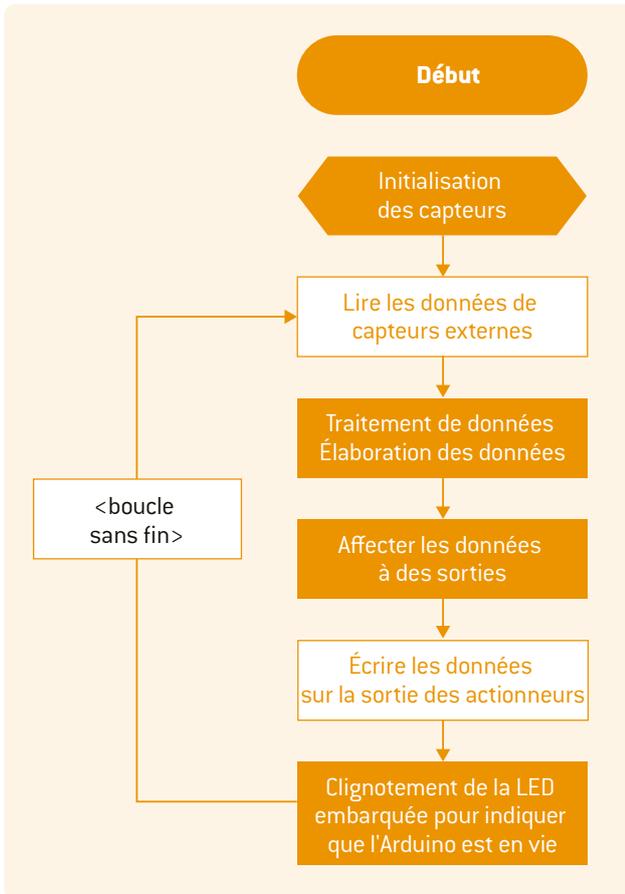
L'unité présente aux élèves le monde du codage des problèmes physiques, basé sur la détection et la mesure de grandeurs physiques, le traitement des données, la réaction et la décision d'effectuer une action avec des actionneurs.

Le code est normalement basé sur une boucle infinie (la machine est généralement « vivante » tant qu'elle est alimentée, et elle doit fonctionner tout le temps), où les trois actions de mesure, traitement et action sont effectuées dans cet ordre.

Les étudiants apprennent qu'ils peuvent écrire n'importe quel code avec

1. des instructions séquentielles
 2. des boucles (while ... do; repeat ... until)
 3. des instructions (si ... alors ... alors ... autrement)
- comme indiqué dans le théorème de Böhm-Jacopini (voir « Informations complémentaires »^[4]).

Le deuxième objectif de l'unité est de présenter aux élèves les microcontrôleurs. Ils apprennent à configurer des capteurs



© 3 : Organigramme

d'entrée et des actionneurs de sortie, en utilisant des ports numériques ou analogiques, et à écrire un programme simple qui lit les entrées, traite les données et écrit les sorties. Ils peuvent choisir d'émettre des signaux sonores ou lumineux. Ceux-ci peuvent être interprétés comme des « alarmes » qui émettent un avertissement basé sur l'entrée lue par les capteurs.

Les étudiants utilisent l'environnement de développement intégré Arduino^[1] (Arduino IDE^[3]) pour programmer en C++, avec des bibliothèques de fonctions prêtes à l'emploi qui rendent le codage plus facile et plus rapide.

Une plaque d'expérimentation (voir « Informations complémentaires »^[4]) doit être disponible pour permettre aux étudiants de prototyper et de connecter facilement les broches des capteurs aux E/S Arduino, à l'alimentation 5V et à la broche GND.

La structure de n'importe quel programme, en métalangage, est illustrée à la ©3.

Veuillez noter que la méta-instruction « *While (TRUE)* » est une astuce pour dire à n'importe quel processeur de répéter

les instructions incluses indéfiniment (tant que le microcontrôleur est sous tension).

Le troisième objectif principal est d'apprendre à convertir une quantité physique acquise en une autre quantité (par exemple l'intensité lumineuse en son), prête à être transmise à l'environnement extérieur, en suivant ces étapes :

1. Le signal physique (lumière, son, force, énergie...) est acquis par le capteur et converti en signal électrique.
2. Le signal électrique est transformé en un nombre à la disposition du processeur.
3. Le nombre est traité et transformé par le processeur en un autre nombre, puis utilisé pour effectuer une action avec un transducteur actionneur.
4. Les actionneurs convertissent le nombre en signaux électriques prêts à être émis.
5. Le signal électrique est finalement transformé en un signal physique (par ex. son, lumière).

En 1 et 5, les signaux doivent être convertis d'une forme à une autre. Dans ces phases, la linéarité de la transformation, ou la dépendance « proche de la linéarité », est extrêmement importante (voir « Informations complémentaires »^[4]).

Voir « Informations complémentaires »^[4] pour une explication détaillée de la dernière ligne du métacodage (« Clignotement de la LED embarquée »).

Le signal d'entrée est généralement appelé « stimulus » et provient de l'environnement dans lequel les capteurs sont placés pour obtenir des données. Le processeur et le code sont conçus pour « réagir » au stimulus par des opérations mathématiques/logiques (effectuées par les instructions du code, traitées par le microcontrôleur) et pour fournir cette « réponse » à l'environnement. Dans nos activités, l'environnement est l'espace qui entoure l'Arduino, capable de « voir », « entendre » et « sentir les forces » grâce aux capteurs.

Travailler avec le TI-Innovator Hub permet aux étudiants de se concentrer davantage sur la partie codage de leur travail, car les microcontrôleurs et les capteurs sont déjà installés et prêts à l'emploi.

<Ce que font les élèves/enseignants>

Nous vous recommandons de commencer par un remue-méninges pour recueillir les idées naïves de vos élèves sur les capteurs et la commande automatique des machines. La collecte de ces idées, l'acquisition d'une expérience pratique avec les capteurs et la commande automatique de la machine, puis la comparaison des résultats avec leurs idées anté-

rieures (peut-être incorrectes) est un bon moyen d'aider les élèves à comprendre vraiment le processus.

Vous pouvez préparer un formulaire avec des questions comme :

- ↳ Savez-vous comment un thermostat contrôle la température dans une pièce ?
- ↳ À quoi sert un système de stationnement sonore basé sur des avertisseurs ? Comment le conducteur réagit-il lorsque le son est émis par le système ?
- ↳ Avez-vous une plaque à induction dans votre cuisine ? Que signifie une LED allumée ?

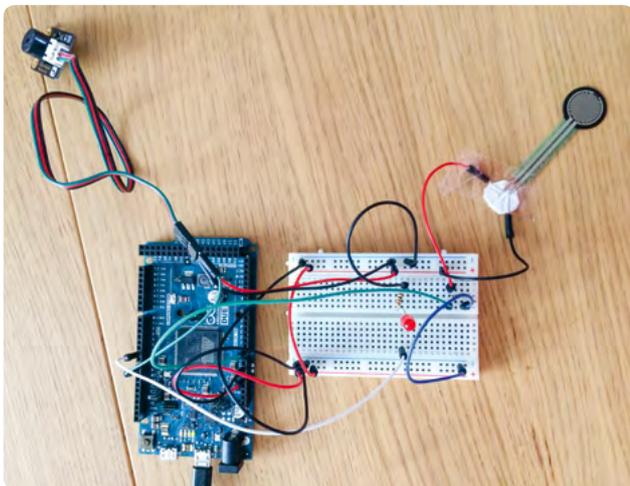
Voir « PEC et liste de questions »^[4] pour un exemple de liste complète de questions et de références à la méthodologie d'enseignement du PEC (Prévision, Expérience et Correction).

<Phase théorique avec Arduino^[4]>

L'enseignant introduit la programmation C++^[3] avec la structure et les instructions de base pour que les élèves puissent écrire une boucle simple en utilisant les instructions *analogRead*, *digitalRead*, *analogWrite*, *digitalWrite*, *si...alors...sinon*, *boucle*, *while*.

Partie matérielle

L'enseignant présente la disposition du microcontrôleur, en



© 4 : Arduino, plaque d'appui et capteurs

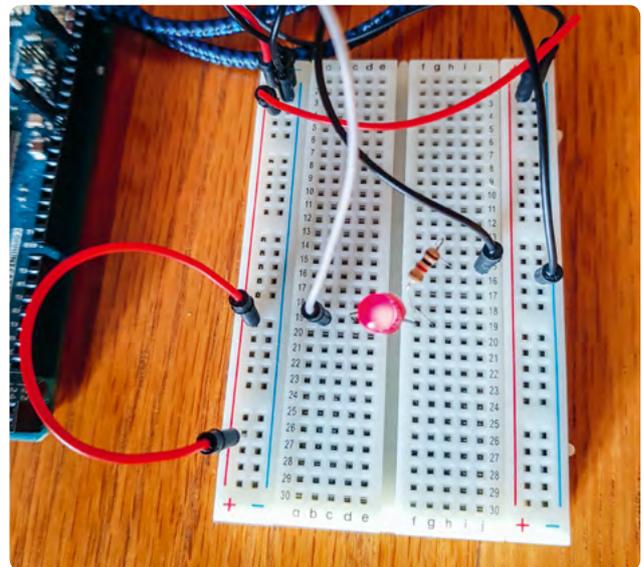
montrant le microprocesseur, les broches d'E/S analogiques (connexions) et les broches d'entrée/sortie numériques (connexions).

Les élèves apprennent que n'importe quel capteur/actionneur a généralement plusieurs :

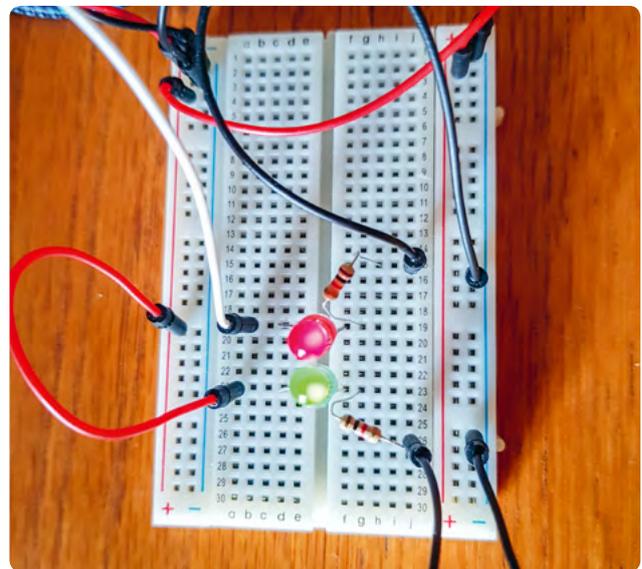
- ↳ Sortie 5V ou 3,3V de l'Arduino pour obtenir l'alimentation électrique
- ↳ Signal GND (masse) pour qu'un courant puisse circuler et

↳ Une autre broche d'entrée numérique ou analogique si elle est utilisée pour lire (obtenir) des données externes ou

↳ Une autre broche de sortie numérique ou analogique si elle est utilisée pour effectuer des actions qui génèrent une sortie, par exemple émettre un son ou une lumière, ou faire quoi que ce soit d'autre qui signale une situation (une opération d'« écriture »)



© 5 : Détail de la plaque d'expérimentation



© 6 : Plaque d'expérimentation avec LEDs

Partie logicielle

L'enseignant présente un programme simple qui lit un capteur et écrit un actionneur, où une association claire entre les broches physiques et l'adresse physique à bord du microcontrôleur peut facilement être établie par les élèves.

Un exemple d'instructions prêtes à l'emploi est disponible en ligne [« Exemple de programme 1 »^[4]].

Les élèves doivent garder à l'esprit que le processeur exécute les instructions du code l'une après l'autre et dans l'ordre dans lequel elles sont écrites. Seule l'instruction « loop » modifie ce principe, car elle indique au processeur de répéter continuellement les instructions entre crochets tant que le microcontrôleur est sous tension.

<Phase pratique avec Arduino>

Les étudiants réalisent une acquisition avec le microcontrôleur, la plaque d'expérimentation et les capteurs. L'enseignant présente la structure de la plaque d'expérimentation, en montrant toutes les connexions disponibles et comment les élèves peuvent prendre les signaux 5V, GND et I/O de l'Arduino^[4] sur la plaque d'expérimentation. Les élèves sont invités à copier l'exemple de codage donné et à essayer, tester et déboguer le code avec les E/S connectées.

Partie matérielle

Les élèves ont besoin du microcontrôleur, des capteurs et de câbles courts (10 cm) pour permettre une connexion facile entre les broches des capteurs et les « trous » de la plaque d'expérimentation. Il se peut qu'ils aient parfois besoin de souder des câbles supplémentaires aux capteurs, mais de nombreux capteurs n'en ont pas besoin.

Les élèves utilisent les câbles courts pour connecter l'alimentation 5V et le signal GND à la plaque d'expérimentation, ainsi que les broches analogiques/numériques de l'Arduino^[4] à certains « trous » sur la plaque d'expérimentation. Cela permet aux capteurs de se loger facilement sur la plaque d'expérimentation et de recevoir les signaux électriques nécessaires (voir 6).

Partie logicielle

Après avoir vérifié que les connexions microcontrôleur – plaque d'expérimentation ont été correctement établies, avec la correspondance exacte entre le numéro logique de la broche du microcontrôleur et la broche du capteur sur la plaque d'immatriculation, les élèves essaient d'écrire le code simple fourni (« Exemple de programme 1 »^[4]).



7 : Un capteur de lumière

Algorithmes avec Arduino

Nous avons préparé plusieurs conversions de signaux d'une forme physique à une autre.

Conversion d'un signal lumineux analogique en un signal lumineux numérique dans une LED (modulé avec une fonction PWM) et un son : la fréquence sonore émise augmente avec l'intensité de la lumière. Application pratique : réveil, assistance aux personnes aveugles. (Voir 7)

Conversion d'une force détectée par un capteur de force en un signal lumineux numérique dans une LED (modulée avec une fonction PWM) et un son : l'intensité lumineuse augmente avec l'intensité de la force. Application pratique : alarme de charge de poids. (Voir 8)



8 : Un buzzer numérique

Conversion d'un signal de bruit externe en un signal lumineux numérique dans une LED. Plus le son est élevé, plus la fréquence lumineuse est élevée. Application pratique : lutte contre les nuisances sonores.

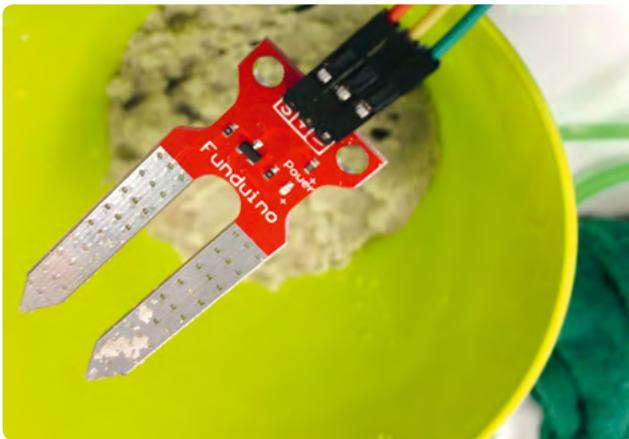
Conversion d'une distance mesurée avec un capteur de distance en un son. Application technique : capteurs de stationnement. (Voir 9)



9 : Un capteur de distance

Conversion d'un signal de température en signal sonore et lumineux. Application technique : contrôle de la température du four.

Conversion de la concentration d'humidité du sol en un signal lumineux. Application technique : irrigation et arrosage des plantes, alarmes et contrôle. (Voir [10])



[10] : Un capteur d'humidité du sol

Voir « Exemple de programme 2 »^[4] pour le code utilisé pour ces applications sur la carte Arduino.

Tous ces algorithmes servent de systèmes de contrôle du signal et d'alerte qui surveillent un environnement sélectionné sur une grandeur physique et émettent un avertissement basé sur l'entrée (stimulus) lue.

<Phase théorique avec le TI-Innovator>

Partie matérielle

L'enseignant peut montrer aux élèves à quel point il est facile de connecter les capteurs.

Partie logicielle

La programmation de base peut se faire uniquement sur la calculatrice ; les étudiants n'ont qu'à maîtriser les instructions de TI-Basic fournies ci-dessus. Ensuite, le concentrateur peut être connecté et les élèves apprendront à communiquer avec lui, c'est-à-dire à utiliser les instructions « read » et « get » pour acquérir les données et « set » pour contrôler les sorties.

<Phase pratique avec le Concentrateur TI-Innovator Hub>

Matériel et Partie logicielle

Les étudiants commencent par des exemples de base pour se familiariser avec le hub avant de travailler sur des problèmes plus ouverts. Les élèves apprennent à contrôler les différentes sorties à l'aide de petits exercices, par exemple en contrôlant la couleur de la LED, en la faisant clignoter, en contrôlant la durée du clignotement et en produisant des sons d'une fréquence donnée. La boucle infinie est à nouveau la structure de base pour continuer les opérations indéfiniment.

Algorithmes avec le Hub TI-Innovator

Les élèves résolvent deux problèmes ouverts : le calcul d'un interrupteur automatique qui n'allume la lumière que si l'intensité lumineuse ambiante est inférieure à un certain seuil, et le calcul d'un réveil qui émet un son dont la fréquence augmente à mesure que la lumière ambiante augmente. D'autres développements sont possibles, mais des capteurs supplémentaires doivent être achetés et branchés au hub.

<Achat des capteurs>

Des informations sur où et comment acheter les capteurs sont disponibles en ligne.^[4]

<Conclusion>

À la fin de ces activités, nous avons constaté que la compréhension du codage, de la structure générale du programme ainsi que de la logique et des algorithmes de nos élèves s'était considérablement améliorée.

<Activité de coopération>

Une merveilleuse activité de coopération pourrait être de promouvoir l'auto-entrepreneuriat. Les élèves pourraient essayer d'inventer une interface homme-machine (IHM) originale et techniquement utile. Cette IHM devrait lire un stimulus provenant d'un environnement (atmosphère, maison, corps humain, etc.) et réagir en émettant un autre signal qui émet un avertissement, effectue une action ou signale une situation. Les écoles partenaires à l'étranger pourraient réaliser une étude de marché pour évaluer la demande du marché et la valeur que l'appareil peut avoir dans leur pays. Toute école participant à cet échange pourrait inventer un dispositif et faire des recherches de marché pour le produit assemblé par les autres écoles. À la fin du projet, l'outil le plus populaire pourrait être produit à petite échelle par une entreprise partenaire et vendu. L'auto-entrepreneuriat est très apprécié dans le monde entier, car il offre une excellente occasion d'enseigner ensemble des matières scientifiques, technologiques et financières.

<Références>

- [1] www.arduino.cc
- [2] www.arduino.cc/en/Guide/ArduinoDue
- [3] www.arduino.cc/en/Main/Software
- [4] Tous les autres documents sont disponibles à l'adresse www.science-on-stage.de/coding-materials.

CoALA-Code un petit animal

<Auteur> Mirek Hančl

<Auteur> Julia Winckler



<Info>

<Mots-clés> simulation, modèle IPO (input-processing-output), mesure, réflexion informatique, fabrication

<Disciplines> science, biologie, informatique

<Niveau d'âge des élèves> 9–13 ans

<Matériel> Calliope mini^[1] ou BBC micro:bit^[2]

<Atelier A> pinces crocodile, plastique rouge, câble USB et batterie pour le Calliope mini, ruban de cuivre autocollant (5 mm), carton, colle, ciseaux, petit verre à eau, affiche avec photos d'animaux

<Atelier B> pinces crocodile, câble USB et batterie pour le Calliope mini, capteur d'humidité Grove, capteur tactile Grove I2C, Grove NFC, Grove I2C hub^[3], carton, plastique artisanal rouge, petit verre à eau, affiche avec photos animales

<Langage> MakeCode^[4]

<Niveau de programmation> facile

<Résumé>

Vous auriez du mal à trouver un enfant qui ne voudrait pas posséder d'animal de compagnie. Pour savoir lequel est le meilleur choix, les élèves construisent un simulateur qui est contrôlé par un ordinateur de bord unique et qui utilise des capteurs externes pour imiter les besoins d'un animal domestique.

<Introduction des concepts>

Le thème des « animaux de compagnie » ne fait pas seulement partie du programme des écoles primaires, mais aussi des écoles secondaires de biologie, où les élèves apprennent comment les chiens ont été élevés à partir de loups, les besoins fondamentaux d'un animal et les exigences auxquelles les propriétaires doivent satisfaire. Par conséquent, un simulateur électronique pour décrire les besoins de base d'un animal de compagnie (nourriture, boisson, exercice, caresses et température corporelle correcte) serait à la fois illustratif et instructif.

Le projet CoALA n'utilise pas d'appareils prêts à l'emploi provenant de fabricants commerciaux de matériel didactique, qui ne permettent que des programmes spécifiques et limités fournis par le fabricant. Au lieu de cela, les élèves planifient, construisent et programment leur propre simulateur sous la forme de leur animal de compagnie préféré, y compris une image de l'animal, à l'aide d'un microcontrôleur dans notre cas, un Calliope mini^[1] ou un BBC micro:bit^[2] et de fournitures artisanales comme du carton, du ruban de cuivre ou des capteurs externes. Les élèves programment un algorithme pour enregistrer et évaluer les besoins de base de l'animal choisi. Selon l'algorithme, le simulateur animal montre diffé-

rents smileys (pour montrer comment l'animal se sent) ou joue des mélodies composées par lui-même.

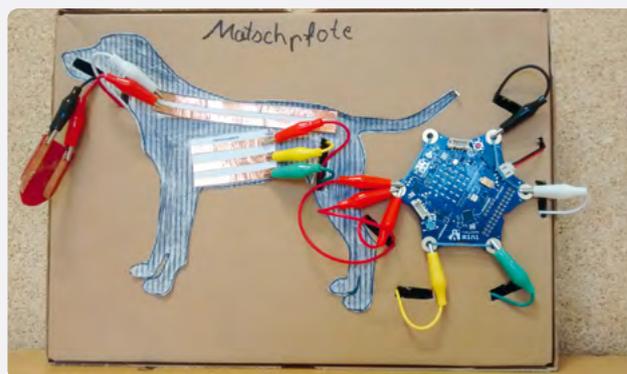
Le concept du projet « CoALA-Code a Little Animal » est celui d'un atelier. Le matériel didactique REL (Open Educational Resources) se compose de trois parties. La première partie initie les élèves aux bases de l'algorithmique et à l'utilisation du microcontrôleur Calliope mini^[1]. Dans la deuxième partie, ils explorent les besoins fondamentaux d'un animal de compagnie et la façon de les évaluer. Dans la troisième partie de l'atelier, les élèves construisent leur animal de compagnie préféré avec du carton, installent le microcontrôleur et les capteurs appropriés et créent les algorithmes appropriés à l'aide d'un langage de programmation graphique.

Pour répondre aux exigences du programme de sciences dans les écoles primaires et du programme de biologie dans les écoles secondaires, nous offrons le matériel de l'atelier en deux versions. Pour les écoles primaires (atelier A), les taux de consommation de nourriture, d'eau et de caresses sont mesurés et enregistrés à l'aide d'un ruban de cuivre conducteur et adhésif. Pour les écoles secondaires (atelier B), les élèves utilisent des capteurs externes pour mesurer l'humidité (boire), pour les interactions multi-touch (caresses) et pour la lecture sans fil des puces Near Field Communication (manger). Dans les deux versions, des capteurs intégrés mesurent le mouvement et la température.

Tout le matériel de l'atelier ainsi que des exemples de codage pour l'environnement de programmation MakeCode^[4] sont disponibles pour téléchargement en ligne.^[5]

<Ce que font les élèves/enseignants>

Pour construire le simulateur d'animal de compagnie, les élèves cherchent une image de leur animal préféré ou prennent eux-mêmes une photo. L'image imprimée est collée sur le carton et équipée de ruban adhésif en cuivre (atelier A) ou de capteurs externes (atelier B) aux endroits appropriés. La bande de cuivre adhésive ou les capteurs externes sont connectés aux ports du microcontrôleur, un programme ap-





```

lorsque le bouton B est pressé
si
  broche P0 est pressée et non broche P1 est pressée
alors
  afficher texte « Souris »
si
  non broche P0 est pressée et broche P1 est pressée
alors
  afficher texte « Saucisse »
si
  broche P0 est pressée et broche P1 est pressée
alors
  afficher texte « Oiseau »
si
  broche P3 est pressée
alors
  afficher texte « Eau! »
  montrer l'icône [smiley]
  pause (ms) 2000
  efface l'écran
  
```

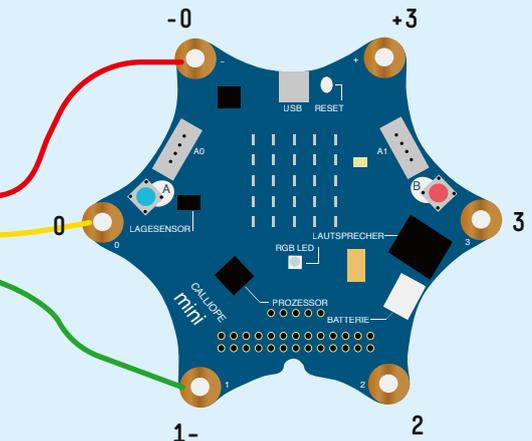
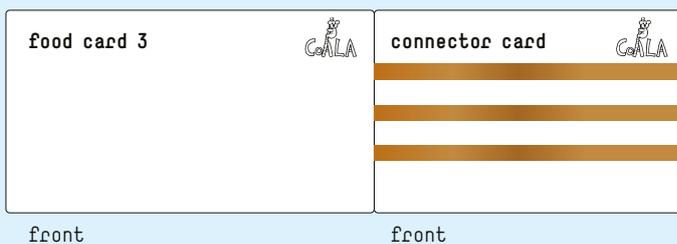
propriété est écrit sur l'ordinateur pour le rendre « intelligent ». Dans l'exemple suivant, le besoin de base « nourriture » est utilisé pour expliquer en quoi les deux versions de l'atelier diffèrent et comment l'environnement de programmation est utilisé.

Lorsque l'alimentation est simulée pour les animaux familiers, les capteurs de goût ne peuvent évidemment pas être utilisés. Au lieu de cela, le capteur approprié « lit » la nourriture offerte et l'algorithme est contrôlé par une branche conditionnelle appropriée afin que la sortie corresponde au comportement attendu de l'animal. Par conséquent, l'écran du simulateur de chat montre un smiley (visage) lorsqu'on lui donne à manger une souris et un visage triste lorsqu'il reçoit un os. Ces branches sont les mêmes pour les deux versions de l'atelier.

Cependant, les capteurs alimentaires sont complètement différents. Dans l'atelier A, des images de différentes sortes d'aliments sont fixées sur des cartes en carton. De l'autre côté, une bande de cuivre est collée sur ces cartes de sorte que la « langue » du simulateur « lit » un nombre codé binaire lorsque la carte y est tenue. Comme les connexions du « lecteur », c'est-à-dire de la « langue », sont connectées aux dif-

férentes broches du micro contrôleur, l'algorithme peut tester directement si les broches sont court-circuitées ou non : les cartes alimentaires court-circuitent différentes combinaisons de broches.

Dans l'atelier B, un capteur externe avec une puce NFC et une antenne radio est utilisé pour lire sans fil les chaînes de caractères d'une balise NFC. Cette balise peut être apposée sur une étiquette adhésive ou sur une carte à puce. Contrairement à l'atelier A, aucun nombre codé binaire n'est lu, mais le nom d'un aliment comme « poisson » ou « os ». Cela augmente considérablement les options de mise en œuvre et la complexité. Dans l'algorithme, la branche conditionnelle est contrôlée en comparant la valeur lue avec les chaînes fournies. Dans le projet CoALA, les balises NFC sont écrites via une application pour smartphone ; la lecture de la balise NFC est réduite didactiquement à un seul bloc de code dans MakeCode^[4], puis chargée comme une extension de l'environnement de programmation.

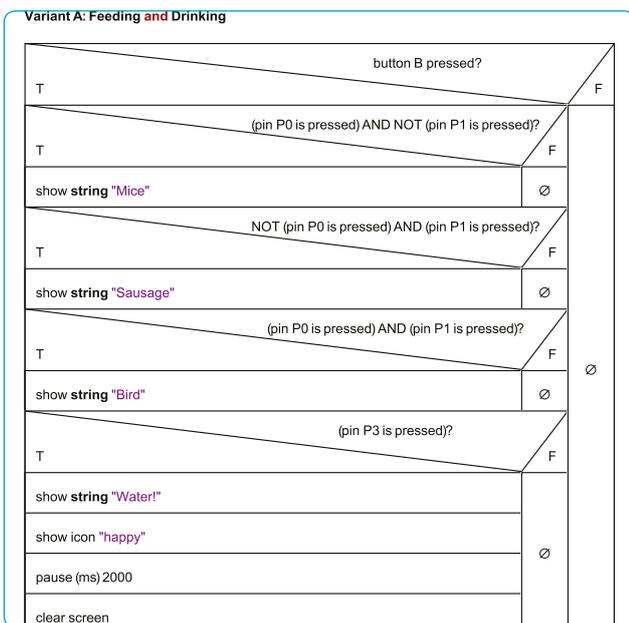




<Algorithme à utiliser dans d'autres langues>

Les exemples de codes disponibles^[5] peuvent être téléchargés dans l'éditeur de programmation en ligne MakeCode^[4] et utilisés directement. En passant de la vue bloc à la vue texte, le code source est converti en JavaScript et peut donc être facilement utilisé dans d'autres langages de programmation pour le Calliope mini^[1] ou le BBC micro:bit^[2]. Les extensions de programmation pour MakeCode, utilisées dans les documents de l'atelier pour contrôler les capteurs multi-touch et NFC, fonctionnent également pour le micro:bit de la BBC.

Enfin, les exemples de programmation sont fournis sur le site sous forme de diagrammes de structure afin que les algorithmes puissent être facilement compris et portés vers d'autres plateformes et environnements de programmation tels que Arduino.



<Conclusion>

Le projet CoALA permet aux étudiants de se familiariser avec les concepts fondamentaux de l'algorithmique – énoncés, séquences, branchements conditionnels, boucles et variables. Ils ne les apprennent pas par simple mémorisation et reproduction, mais plutôt en travaillant sur un projet éducatif passionnant et applicable dans la vie réelle. Ils utilisent des matériaux simples pour construire un simulateur personnel d'animaux de compagnie, à qui ils donnent vie à l'aide du co-

dage et de leur propre imagination. Le matériel d'atelier fourni permet d'enseigner les compétences informatiques sous une forme didactiquement simplifiée tout en offrant différents niveaux d'apprentissage, répondant ainsi aux besoins de groupes hétérogènes ou d'élèves plus âgés. Les deux versions de l'atelier peuvent être facilement mélangées.

Le matériel de ce projet a été testé avec succès en utilisant un Calliope mini^[1] et microcontrôleur BBC micro:bit^[2]. Une carte d'extension à faible coût est nécessaire pour utiliser les capteurs de rainures externes pour les mesures d'humidité, pour le NFC ou pour le multi-touch dans l'atelier B avec le micro:bit BBC.^[3]

<Activité de coopération>

L'atelier CoALA peut être utilisé dans différentes formes de coopération. Comme ce projet est destiné aussi bien aux écoles primaires qu'aux écoles secondaires, un échange pourrait avoir lieu entre différents types d'écoles. Les matériaux de l'atelier A sont conçus pour enseigner des différenciations logiques simples Oui/Non, tandis que les matériaux de l'atelier B sont conçus pour enseigner des conditions, des variables et des opérations de chaîne plus complexes et combinées.

Le matériel des ateliers peut également être combiné selon les besoins pour promouvoir la coopération au sein de groupes d'apprentissage hétérogènes. Les élèves qui ont besoin de plus de soutien pourraient, par exemple, utiliser les capteurs simples de l'atelier A, tandis que les élèves plus forts pourraient expliquer les capteurs de l'atelier B à leurs camarades de classe et mettre en pratique leurs compétences en communication dans ce processus.

Dans le cadre du projet CoALA, il y a eu une coopération transnationale entre deux écoles secondaires au cours de laquelle deux groupes d'élèves – l'un d'Allemagne et l'autre d'Espagne – ont discuté par vidéoconférence de leurs expériences avec des simulateurs d'animaux domestiques. En plus des suggestions de résolution de problèmes, le vocabulaire des noms et des besoins de base de leurs animaux de compagnie ont été échangés en anglais ainsi que dans leurs langues maternelles respectives – le codage dans l'enseignement STEM avec un cours de langue.

<Références>

- [1] <https://calliope.cc/en>
- [2] www.microbit.co.uk/home
- [3] Si vous utilisez un micro:bit BBC, vous avez également besoin d'un Grove Shield pour micro:bit.
- [4] <https://makecode.calliope.cc/?lang=en> or <https://makecode.microbit.org/?lang=en>
- [5] Tous les autres documents sont disponibles à l'adresse www.science-on-stage.de/coding-materials.



Les données liquides

<Auteur> Eleftheria Karagiorgou

<Auteur> Sevasti Tsiliki

<Info>

<Mots-clés> calcul physique, acidité, eau, liquide, eau, température, pH, enregistrement des données

<Disciplines> chimie

<Niveau d'âge des élèves> 16 ans

<Matériel> Kit de démarrage Arduino^[1], écran d'enregistrement des données, sonde de température, sonde pH, carte SD

<Langage> IDE Arduino – Câblage C^[2].

<Niveau de programmation> moyen

<Durée du projet> 7 heures d'enseignement

<Résumé>

Cette unité d'enseignement est une approche interdisciplinaire utilisant l'informatique physique et la chimie. Les étudiants jouent le rôle de chercheurs et mènent une expérience pour déterminer s'il existe une relation entre l'acidité et la température de l'eau. Cela nécessite l'utilisation d'un Arduino et de la chimie.

<Introduction des concepts>

Cette activité pédagogique a été créée pour démontrer aux élèves comment l'informatique physique peut être intégrée dans l'enseignement des STEM, et plus particulièrement dans la chimie. Elle a été réalisée dans le cadre d'une activité parascolaire lors du club Robotique et STEM de notre école.

Les étudiants ont été chargés de démontrer le rôle important que joue la température dans la mesure du pH. Lorsque la température augmente, les vibrations moléculaires augmentent, ce qui permet à l'eau de s'ioniser et de former davantage d'ions hydrogène. Il en résulte une baisse du pH.

<Méthode d'enseignement>

Enseignement scientifique basé sur l'investigation : nous voulions impliquer nos élèves dans un projet d'apprentissage actif, basé sur des questions qui génèrent ensuite d'autres questions à mesure que les élèves progressent dans le projet de recherche.

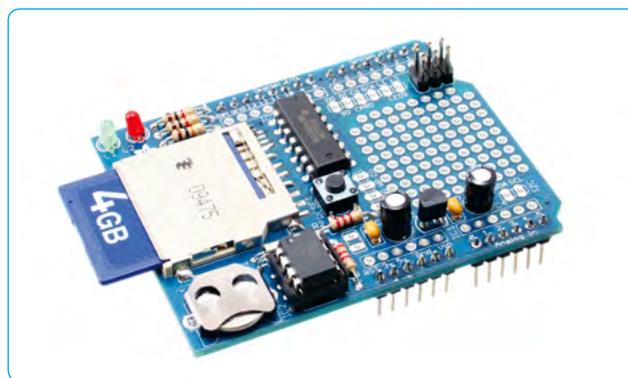
<Prérequis, Connaissances de base>

- ↳ des connaissances de base en programmation
- ↳ connaissances de base sur l'acidité et la théorie du pH

<Matériel d'enseignement/espace utilisé>

Le laboratoire de robotique éducative et de STEM de l'école contient les éléments suivants :

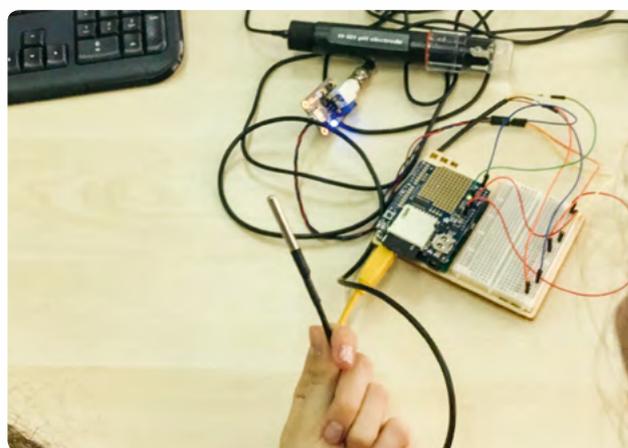
- ↳ Arduino^[1] kit de démarrage (comprenant une carte Arduino, câbles, écran LCD, etc.)
- ↳ Écran de l'enregistreur de données Adafruit pour Arduino ([@1])
- ↳ pH-mètre analogique Pro Kit pour Arduino ([@2])
- ↳ Sonde de température étanche ([@3])
- ↳ carte mémoire SD
- ↳ Un ordinateur avec un port SD, tel qu'un ordinateur portable, est nécessaire pour le codage et l'enregistrement des données
- ↳ Eau déminéralisée
- ↳ Des glaçons et une glacière pour conserver les glaçons



© 1 : Shield d'enregistreur de données Adafruit pour Arduino^[1]



© 2 : pH-mètre analogique pour Arduino



© 3 : Sonde de température étanche

<Problématique>

Y a-t-il une relation entre l'acidité et la température du liquide ?

<Questions de résolution de problèmes>

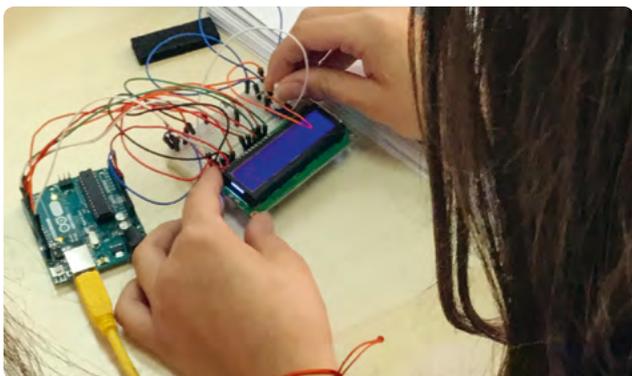
1. Comment connecter les capteurs à la carte Arduino ?
2. Comment procéder pour l'enregistrement des données ?

<Ce que font les élèves/enseignants>

<Phase préparatoire>

Durée : 1 heure

Les élèves sont divisés en groupes et reçoivent une brève introduction pour l'Arduino^[4], les capteurs utilisés (capteur de pH et capteur de température) ainsi que sur leur fonctionnement. Ils discutent également de la théorie de l'acidité, du pH-mètre et de la relation entre l'acidité et la température. On demande ensuite aux élèves de concevoir une expérience pour mesurer le niveau de fluctuation de l'acidité des liquides à mesure qu'ils changent de température.

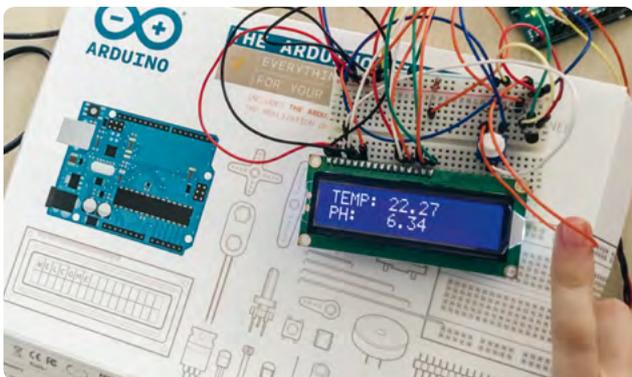


© 4 : Raccordement de l'écran LCD

<Phase 1 : Introduction à Arduino>

Durée : 1 heure

Les étudiants se familiarisent avec les circuits de l'Arduino et apprennent ensuite les bases du code de l'Arduino. Ils apprennent à connecter l'écran LCD à l'Arduino et, grâce au codage, à y afficher un message (©4 & 5)



© 5 : Affichage des mesures du capteur

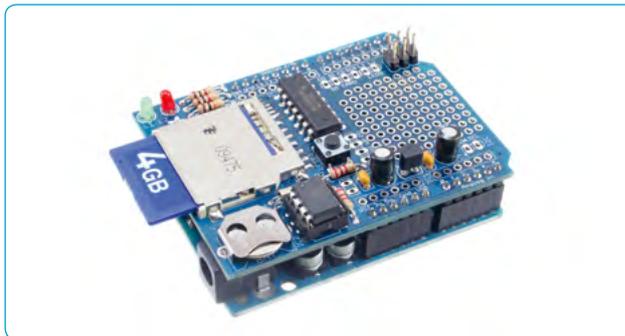
<Phase 2 : Raccordement des capteurs>

Durée : 1 heure

Les élèves apprennent comment fonctionne le capteur de pH (©6) et le capteur de température (©3) en les connectant à l'Arduino^[4] et en les codant pour présenter les données sur l'écran LCD. Il s'agit d'une phase préparatoire pour comprendre les tenants et aboutissants des capteurs.



© 6 : Une sonde de pH



© 7 : Shield d'enregistrement de données soudé à l'Arduino^[4]

<Phase 3 : Enregistrement des données>

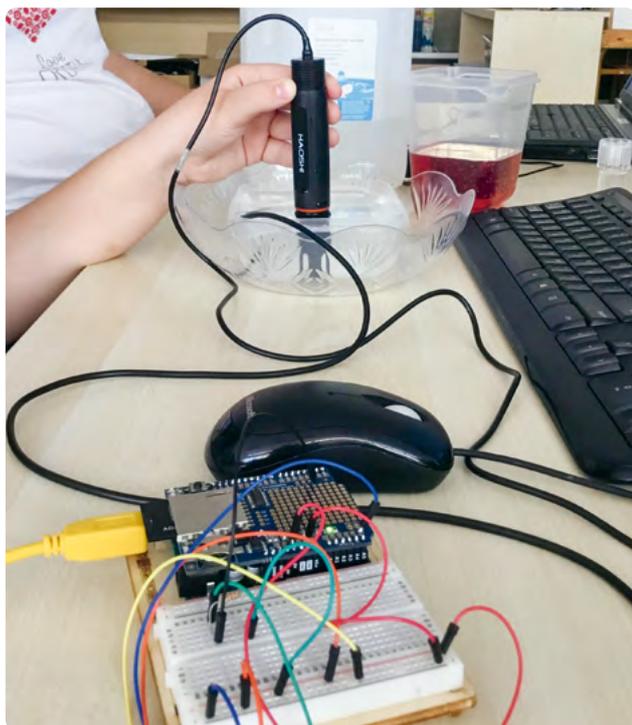
Durée : 2 heures

Les élèves fixent le shield d'enregistrement des données sur la carte Arduino^[4] avec la carte SD (©7). Ils codent le shield d'enregistrement des données, qui possède sa propre horloge en temps réel (RTC). Ils commencent l'expérience en utilisant de l'eau déminéralisée à 25°C (neutre) et mesurent ensuite le pH et la température en plongeant les capteurs dans le liquide pendant 10 secondes.

<Phase 4 : L'expérience>

Durée : 1 heure

Les étudiants effectuent des tests avec des échantillons d'eau déminéralisée, qui ont tous une température différente. Ils commencent avec de l'eau déminéralisée (à température ambiante) dans un bol, ou un bécher, refroidi par des glaçons dans un bain-marie (©8 & 9). Toutes les minutes, ils plongent les capteurs dans le liquide pendant 10 secondes. Ils répètent la procédure au moins 6 fois pour générer une grande quantité de données pour la phase de traçage de l'appareil. De cette façon, le bain-marie refroidit les liquides doucement et graduellement.



© 8 : Measurement of pH and temperature



© 9 : Un bain-marie avec glaçons

<Phase 5 : Résultats>

Durée : 1 heure

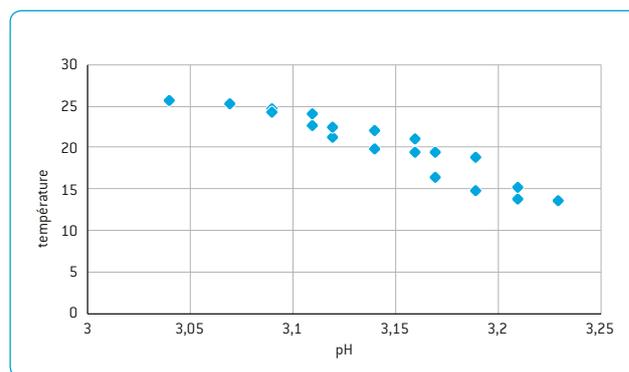
Les élèves débranchent la carte SD de l'écran d'enregistrement des données, la placent dans l'ordinateur portable pour lire le fichier de données et effectuent le traçage, qui illustre le lien possible entre la température et l'acidité à l'aide d'un tableur (p. ex. MS Excel). Les données sont sauvegardées sur la carte SD sous la forme d'un fichier .csv qui s'ouvrira sous la forme d'une feuille de calcul. L'enseignant discute des résultats de l'enregistrement des données avec les élèves et détermine s'ils ont réussi à répondre aux questions de résolution de

problèmes. Les élèves présentent leurs résultats à la classe et en discutent avec leurs pairs.

<Conclusion>

À la fin de l'activité, on s'attend à ce que les élèves comprennent les liens entre les sujets des STEM en appliquant une théorie de la chimie à une expérience de calcul physique. De plus, les élèves réalisent comment les connaissances qu'ils apprennent à l'école peuvent être mises en pratique dans le monde réel grâce à ces activités pratiques. Les compétences non techniques comme la coopération sont essentielles pour leur avenir. Enfin, cette activité offre également aux étudiants une occasion d'améliorer leurs performances dans les domaines liés aux STEM.

L'expérience pourrait être étendue en utilisant une plus grande variété de liquides. Un exemple est le vinaigre au bain-marie – soit pour un liquide froid, avec des glaçons dans le bain-marie, soit pour un vinaigre chaud, avec de l'eau chaude (voir © 10).



© 10 : Données du vinaigre

Pour ce projet, le shield d'enregistrement des données doit être fixé avec précision sur la carte Arduino. Certains de vos élèves peuvent avoir des difficultés à le faire, alors vous devez les guider tout au long du processus ou même la fixer vous-même si nécessaire.

<Références>

- [1] www.arduino.cc
- [2] www.arduino.cc/en/Main/Software
- [3] Picture : oomlout [https://commons.wikimedia.org/wiki/File:ARSH-09-DL_03.jpg], „ARSH-09-DL 03“, CC BY-SA 2.0, <https://creativecommons.org/licenses/by-sa/2.0/legalcode>
- [4] Picture : oomlout [https://commons.wikimedia.org/wiki/File:ARSH-09-DL_5703636953.jpg], „ARSH-09-DL 5703636953“, CC BY-SA 2.0, <https://creativecommons.org/licenses/by-sa/2.0/legalcode>

Captaine télécommandé

<Auteur> Immaculada Abad Nebot

<Auteur> Pere Compte Jové



<Info>

<Mots-clés> télécommande, conception 2D et 3D, modélisation, soudure de cartes électroniques, programmation de puces, programmation d'applications, imprimante 3D

<Disciplines> technologie, ingénierie

<Niveau d'âge des élèves> 14–16 ans

<Matériel> Arduino^[1], module Bluetooth, matériaux pour la construction du bateau modèle réduit

<Langage> Arduino, ArduinoBlocks^[2], AppInventor^[3]

<Niveau de programmation> moyen

<Résumé>

Les élèves concevront et construiront leur propre bateau et navigueront dans une piscine. Une fois qu'ils auront terminé ce premier défi, ils utiliseront une carte Arduino pour piloter le bateau à distance à l'aide d'une tablette ou d'un smartphone.

<Introduction des concepts>

Les élèves concevront leur propre maquette de bateau et apprendront à considérer cette tâche du point de vue de l'ingénierie. Le projet commencera par une analyse des différents types de bateaux, en utilisant Internet. Ensuite, en petits groupes, les élèves construiront un modèle stable sur l'eau. ^[4]

Il fonctionnera avec un interrupteur inverseur fait par les élèves qui permettra à l'utilisateur de contrôler deux moteurs [chacun peut tourner en avant et en arrière].

Les élèves intégreront un dispositif Bluetooth afin de pouvoir contrôler le bateau à distance avec un smartphone. En utilisant AppInventor^[3], ils programmeront une application mobile avec différents systèmes de contrôle, par exemple avec des boutons, une commande vocale ou un accéléromètre (le bateau changera de direction en fonction de la position des mains de la personne qui le contrôle).

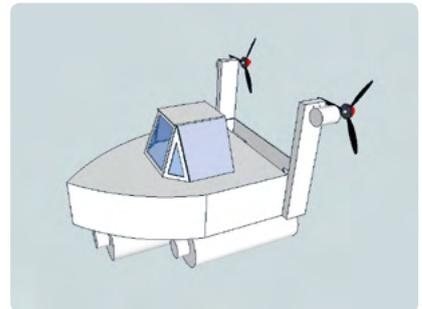
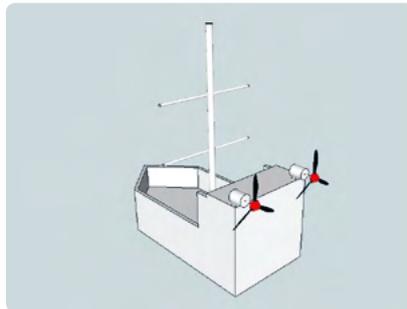
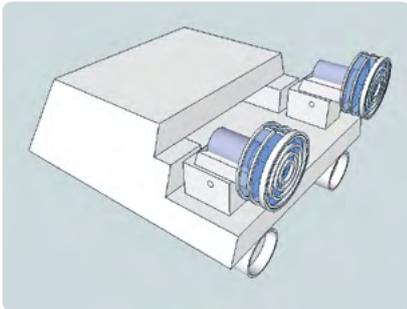
Enfin, les élèves auront l'occasion de présenter leurs dessins lors d'une exposition de modélisation navale [1] devant des experts, et de discuter avec eux des éventuelles lacunes de leurs modèles. Cet apport d'experts permettra aux étudiants d'améliorer leurs futures maquettes.

<Ce que font les élèves/enseignants>

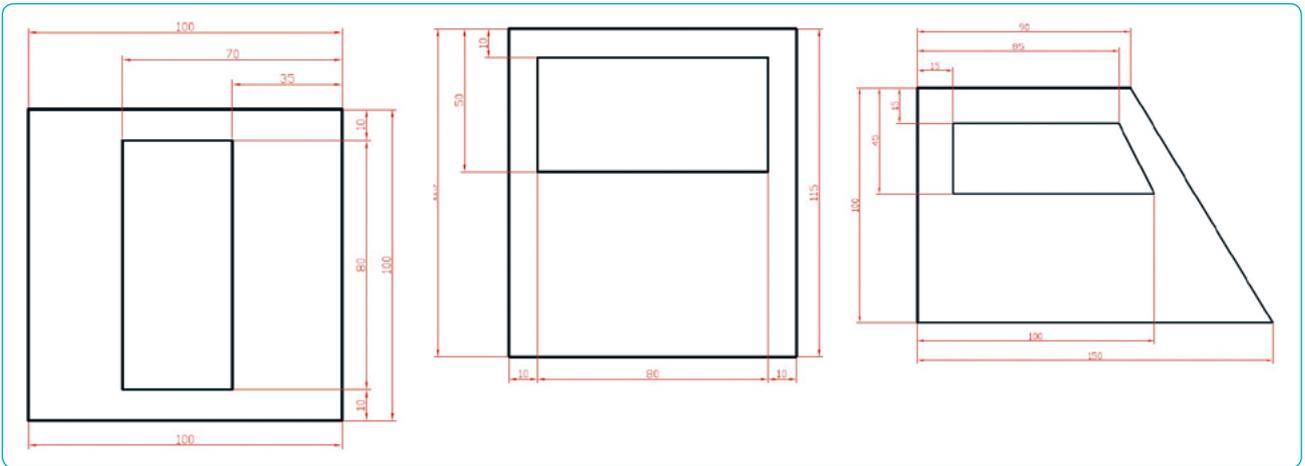
Les élèves créeront le design de leur choix après avoir étudié différentes formes de bateaux sur Internet. Ils peuvent dessiner le bateau avec un logiciel de conception 3D tel que sketchUp^[5] ou Tinkercad^[6]. Cependant, ils doivent garder à l'esprit que les bateaux doivent être très stables sur l'eau pour éviter qu'ils ne se renversent. [2]



1 : Présentation des maquettes de bateaux lors d'une exposition de modélisme naval en Espagne



© 2 : Différents modèles de bateaux 3D



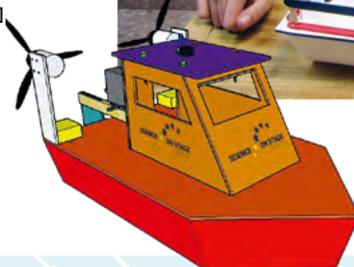
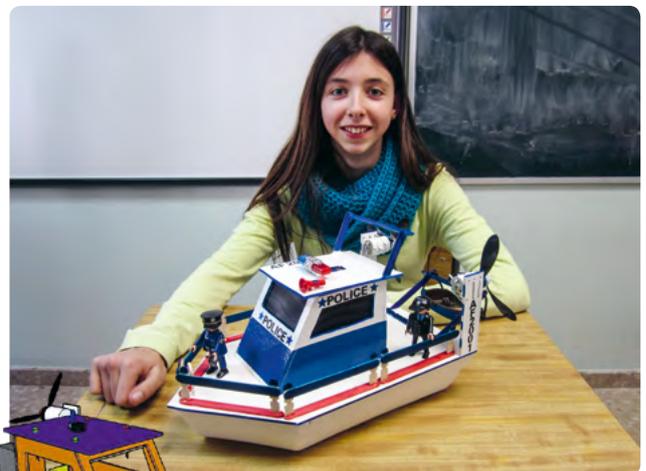
© 3 : Porte, avant et latérale de la cabine du bateau



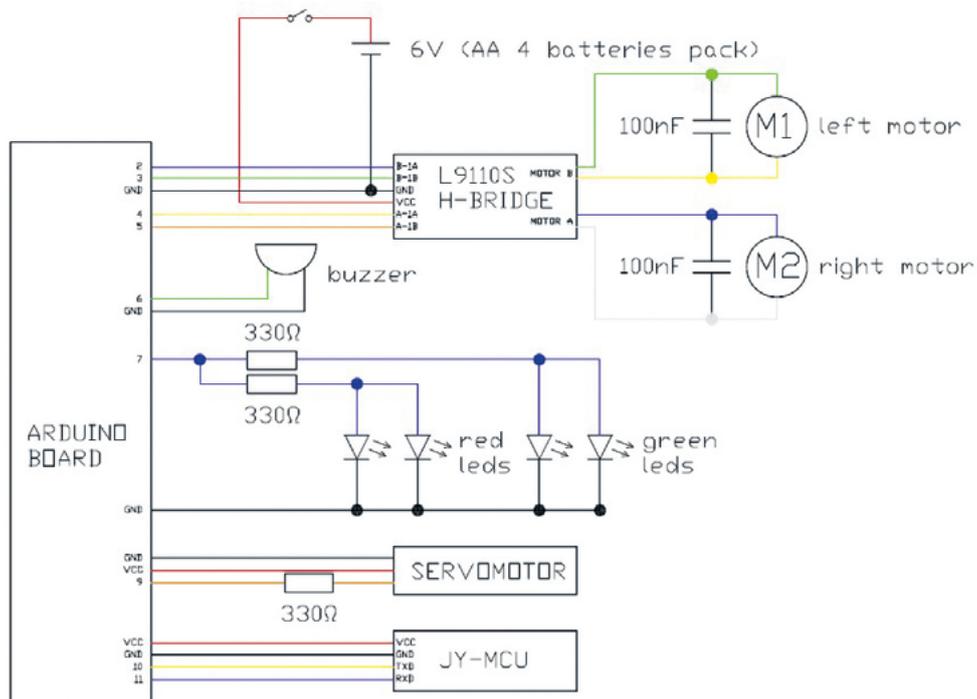
© 4a-c : Impression des hélices au centre d'impression 3D Cesire Aulatec, Barcelone

Nous fournissons en ligne les plans d'un modèle réduit de bateau ([©3]), y compris les mesures requises. Si vous souhaitez modifier ce design ou l'imprimer, vous pouvez télécharger différents formats (.skp,.stl et .gcode).^[4]

Si votre école n'a pas d'imprimante 3D, il est souvent possible de coopérer avec d'autres institutions telles que des universités, des ateliers de fabrication ou autres. Dans notre cas, les étudiants ont réalisé leur impression 3D dans un centre d'impression 3D ([© 4a-c]). Des instructions étape par étape pour la construction du bateau sont disponibles en ligne.^[4]



© 5, 6 : Bateau fini et modèle 3D

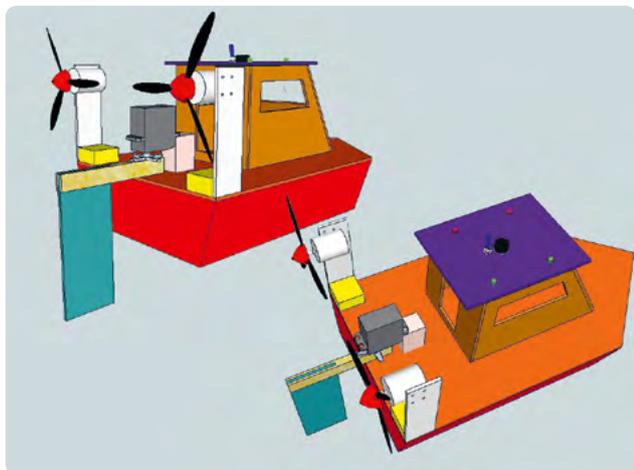


© 7 : Schéma électronique

<Circuit électronique>

Pour connecter les accessoires à la carte Arduino^[7], utiliser le schéma électronique de la ©7 et suivre les instructions correspondantes :

1. Vous pouvez installer un avertisseur sonore (Arduino broche 6) et des lampes (Arduino broche 7) sur le toit. (©8)
2. A l'arrière du bateau, vous pouvez connecter un servomoteur pour contrôler un gouvernail (broche Arduino 9). (©8)
3. Connectez le module Bluetooth conformément au schéma TXD (Arduino broche 10) et RXD (Arduino broche 11).
4. Connectez une carte contrôleur de pilote de moteur L9110S pour Arduino avec des batteries externes. (©7)



© 8 : Le bateau avec ses accessoires



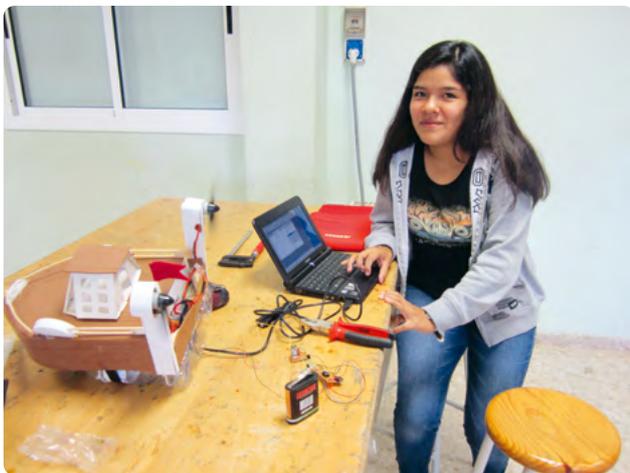
© 9 : Construction des bateaux

<Comment contrôler les moteurs et les autres caractéristiques du bateau>

Nous vous recommandons de programmer l'Arduino avec Arduino IDE^[4], ArduinoBlocks^[2] ou un autre programme similaire. Les élèves peuvent programmer les tâches suivantes :

1. Allumer et éteindre les lumières sur le toit.
2. Faites un son avec le buzzer.
3. Contrôlez la position des servos (40° à droite, 20° à droite, au milieu, 20° à gauche et 40° à gauche).
4. Les deux moteurs doivent tourner et le bateau avance.
5. Les deux moteurs doivent tourner et le bateau recule.

6. Le bateau doit changer de direction vers la droite (le moteur gauche tourne vers l'avant et le moteur droit tourne vers l'arrière).
7. Le bateau doit changer de direction vers la gauche (le moteur droit tourne vers l'avant et le moteur gauche tourne vers l'arrière).
8. Contrôlez tous les programmes par Bluetooth.



© 10 : Programmation des bateaux

<Programmer l'application pour contrôler le bateau avec un smartphone en utilisant AppInventor^[3]>

1. Programmez l'application de manière à ce qu'elle utilise Bluetooth pour se connecter avec le bateau.
2. Contrôlez les différents éléments du bateau à l'aide de boutons.
3. Contrôler le safran à l'aide d'une barre de défilement.
4. Contrôlez le bateau à l'aide de l'accéléromètre de la tablette ou du smartphone ; incliner le smartphone vers l'avant, l'arrière, la droite ou la gauche fera avancer le bateau dans la direction correspondante.
5. Utilisez l'option de reconnaissance vocale pour contrôler le bateau avec votre voix.
6. Combinez tous ces programmes.

<Liste du matériel et de l'équipement nécessaires>

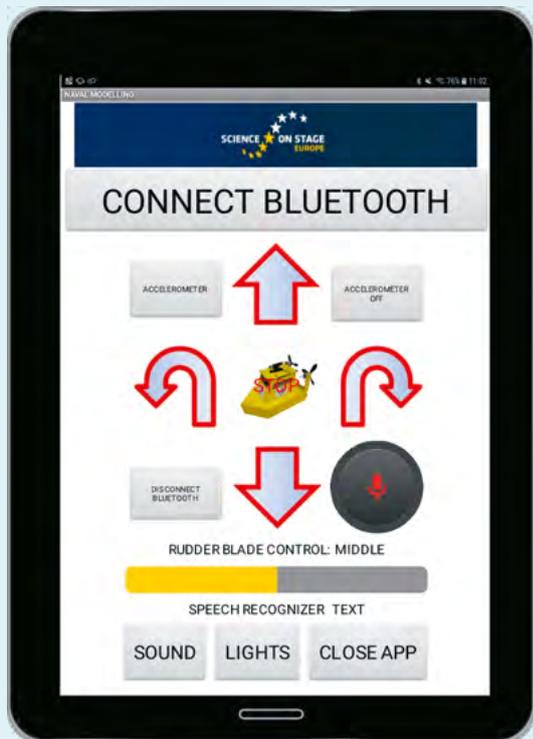
Vous pouvez trouver une liste contenant tous les matériaux nécessaires en ligne^[4]. La liste comprend des détails tels que la quantité nécessaire, la gamme de prix et l'endroit où on peut trouver les pièces.

Le matériel pour un bateau coûte environ 21 €, nous avons dépensé environ 15 € pour l'électronique et 6 € pour le reste du matériel nécessaire.

<Activité de coopération>

En éditant ce matériel pédagogique, nous avons collaboré avec Eleftheria Karagiorgou et Sevasti Tsiliki de la 7^e Senior High School de Trikala, Grèce, pour mettre en place un circuit Arduino dans un hydrobot qui nous permet de naviguer sous l'eau. Dans ce cas, il est très important de protéger les moteurs avec de la cire et d'utiliser un boîtier protecteur et étanche pour la plaque Arduino afin d'empêcher l'eau de pénétrer dans les moteurs. Nous avons équipé notre hydrobot « Argolith »^[8] du microcontrôleur Arduino UNO pour lui fournir un « cerveau » électronique et pour enregistrer les mesures de luminosité et de température sous l'eau. Le « cerveau » comprenait également une carte d'enregistrement des données, qui nous offrait une horloge en temps réel et un connecteur pour carte SD, où nous sauvegardions les données mesurées.

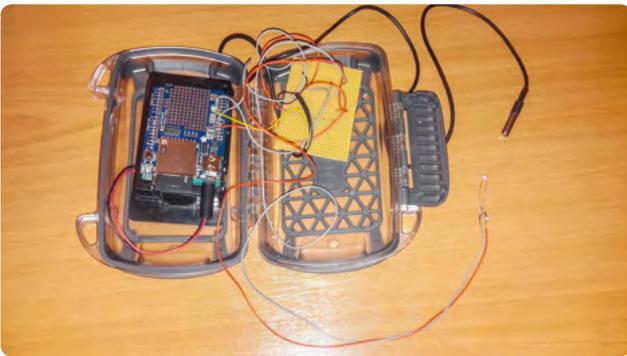
Le matériel en ligne contient une vidéo sous-marine de l'hydrobot « Argolith » lors d'un essai dans une rivière à Trikala, en Grèce.^[4]



© 11 : L'interface utilisateur de l'application



© 12 : Construction de l'hydrobot



© 13 : Arduino avec boîtier étanche

<Références>

- [1] www.arduino.cc
- [2] www.arduinoblocks.com
- [3] <http://appinventor.mit.edu>
- [4] Toutes les étapes de ce projet et des informations complémentaires : www.science-on-stage.de/coding-materials.
- [5] www.sketchup.com
- [6] www.tinkercad.com
- [7] www.arduino.cc/en/Reference/Board
- [8] A construction manual is available at <http://seaperch.mit.edu/build.php>.

Comment Coder

<Auteur> Bernard Schriek

Cet article traite de l'utilisation du codage en science. Il se peut que vous ayez besoin de matériel et de logiciels spéciaux pour utiliser les unités d'enseignement de ce livret. Il est également utile d'avoir une connaissance de base des principes fondamentaux de la programmation. Ce chapitre a pour but de vous fournir un aperçu clair des informations requises.

<Matériel>

Le codage en science concerne principalement la mesure de grandeurs physiques et chimiques à l'aide de capteurs et le contrôle de certaines sorties. Les capteurs typiques mesurent la température, le bruit, la lumière, la distance, le pH, la pression de bouton, le toucher, etc. Les sorties typiques sont une LED, un buzzer, un haut-parleur, un moteur, etc.

<Arduino>

Ces capteurs et sorties peuvent être connectés ou sont déjà intégrés à une carte microcontrôleur ou à un microordinateur monocarte. Ceux-ci peuvent être plus ou moins sophistiqués. Le plus petit microcontrôleur est l'Arduino^[1]. Il est disponible en différentes versions, mais le modèle Arduino UNO est largement utilisé dans les écoles (et aussi dans cette publication). La carte contient un microcontrôleur 8 bits qui fonctionne assez lentement. Cependant, la vitesse du processeur n'est pas importante pour la plupart des applications, car la carte communique uniquement via un bouton de réinitialisation et une LED.

La carte peut être branchée à divers autres capteurs et sorties. Les programmes pour Arduino sont écrits sur un ordinateur et envoyés à un Arduino via une connexion USB. Lorsque le programme est chargé sur l'Arduino, il peut être démarré (puis interrompu et redémarré) en appuyant sur le bouton reset. Lorsque l'Arduino est connecté via USB, il est alimenté par la connexion USB ; cependant, il a besoin d'une source d'alimentation séparée lorsqu'il est déconnecté de l'ordinateur.

Une façon très pratique d'étendre une carte Arduino est d'utiliser des « board », qui sont des cartes d'extension de circuit qui se branchent directement sur les connecteurs Arduino. Par exemple, on utilise souvent des extensions enregistreuses

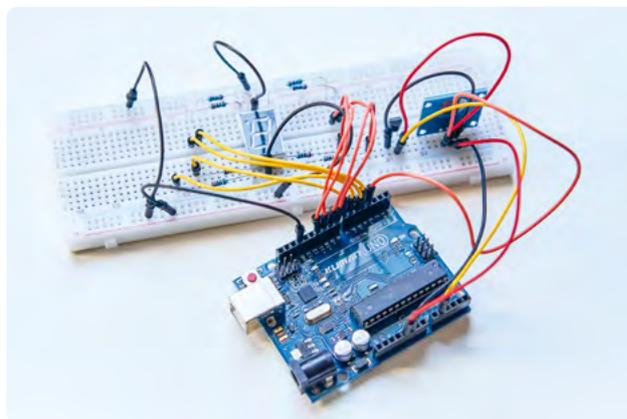
de données dotées d'une horloge en temps réel intégré. Les données sont écrites sur une carte SD qui est insérée dans l'emplacement pour carte SD de la carte.

Vous trouverez sur Internet des exemples de programmation bien documentés pour presque toutes les cartes d'extensions (blindages, capteurs et sorties).

Les ordinateurs Windows, macOS ou Linux peuvent tous être utilisés pour programmer l'Arduino. Le logiciel peut être téléchargé en ligne^[2] vous trouverez également sur Internet de nombreux exemples détaillés de programmes pour Arduino. Les programmes pour l'Arduino sont appelés « sketches ».

<Calliope mini et BBC micro:bit>

Un autre microcontrôleur est le Calliope mini^[3], qui est compatible avec le micro:bit^[4] de la BBC. La différence entre ces deux cartes est que le Calliope mini a de nombreux capteurs et sorties intégrées, donc, pour de nombreux projets, vous n'avez pas besoin de capteurs et sorties externes. Le Calliope mini dispose même de Bluetooth pour communiquer avec d'autres cartes ou smartphones. Le processeur est plus puissant et plus rapide que le processeur de l'Arduino, et il a aussi beaucoup plus de mémoire embarquée. Les programmes écrits sur un ordinateur peuvent être transférés via USB sur le Calliope mini. Le programme démarre automatiquement lorsqu'il est transféré, mais il peut également être redémarré en appuyant

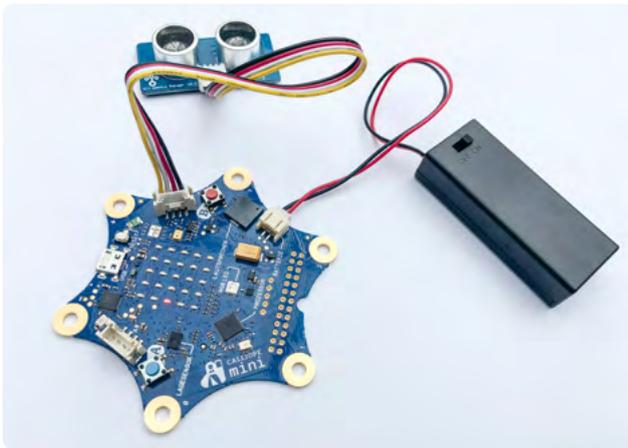


© 1 : Un Arduino

sur le bouton de réinitialisation. Le Calliope mini est équipé de nombreux capteurs, mais d'autres peuvent être fixés à l'aide de connecteurs à rainure normalisés « Groove »^[5].

Il est équipé d'une matrice LED 5×5 qui peut être utilisée pour le défilement de texte. Le Calliope mini est légèrement plus cher que l'Arduino, mais l'avantage d'avoir de nombreux capteurs et sorties à bord en vaut la peine pour un usage pédagogique.

Le Calliope mini utilise JavaScript pour la programmation, mais vous pouvez également utiliser des environnements de programmation orientés bloc qui conviennent mieux aux jeunes enfants. Ils seront décrits plus loin dans ce chapitre.



© 2 : Calliope mini avec capteur ultrasons et batterie

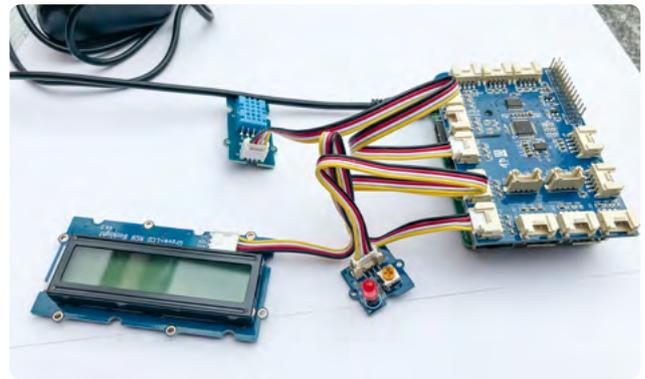
<Raspberry Pi>

Le Raspberry Pi^[6] est un célèbre ordinateur monocarte entièrement fonctionnel qui utilise Linux et Windows comme système d'exploitation. Il peut être connecté à un écran à l'aide d'un câble HDMI, et une souris et un clavier peuvent être connectés via USB. Vendu entre 50 et 60 euros, son prix est très raisonnable, mais toujours le double de celui d'un Calliope mini. Vous pouvez utiliser presque tous les langages de programmation sur le Raspberry Pi. Au lieu d'un disque dur, il utilise une carte SD pour sauvegarder les programmes et les données. Il peut être étendu avec différentes cartes d'extension



© 3 : Un Raspberry Pi

pour différents usages. Il dispose d'une connexion Wi-Fi embarquée et peut être relié à un réseau Wi-Fi local. Comme toutes les autres cartes, il a besoin d'une alimentation externe avec des piles ou un adaptateur secteur. Bien que le Raspberry Pi soit un ordinateur complet, il n'est pas aussi rapide qu'un ordinateur de bureau ou un ordinateur portable normal, et il ne possède qu'une carte SD comme mémoire externe. Il y a des milliers de projets pour cela sur Internet. Vous devriez avoir de l'expérience avec les ordinateurs pour travailler avec le Raspberry Pi.



© 4 : Raspberry Pi avec carte d'extension

<LEGO Mindstorms>

LEGO Mindstorms^[7] est un système micro-ordinateur beaucoup plus coûteux. La plupart des étudiants ont une certaine expérience de LEGO, de sorte qu'ils peuvent facilement construire des outils et des machines contrôlés par programme. LEGO possède son propre système de programmation à base d'icônes appelé logiciel EV3, basé sur LabView^[8] (un logiciel professionnel pour la mesure et le contrôle). Vous déplacez ici les blocs de programmation pour créer un programme de travail. L'utilisation et la programmation des moteurs sont très importantes lorsque vous programmez un robot EV3. En plus du logiciel LEGO original, vous pouvez également utiliser le jOS^[9], une implémentation spéciale de la machine virtuelle Java. La plupart des gens utilisent Eclipse comme outil de développement Java sur un ordinateur et il existe une extension LEGO pour cet environnement de programmation. Un grand nombre de briques de capteur et de sortie peuvent être connectées à la brique du processeur principal, mais comme tous les matériaux LEGO, elles sont beaucoup plus chères que les capteurs des cartes mentionnées précédemment.

Il existe des centaines d'autres microcontrôleurs, dont beaucoup sont conçus pour faire fonctionner des robots. Si vous êtes à la recherche d'idées de projets, nous vous recommandons de visiter hackster.io^[10]. Cependant, sachez que vous de-

vez vous inscrire gratuitement pour avoir accès à de nombreuses descriptions de projets.

<Logiciels>

Au 21^e siècle, les compétences en programmation et en codage deviendront de plus en plus importantes dans tous les domaines de la vie. L'un des objectifs de cette brochure est d'aider les enseignants à encourager leurs élèves à s'intéresser au codage. Les étudiants sont généralement très intéressés par le codage et n'ont besoin que des bons outils et langages de programmation pour réussir et en être plus motivés. Le choix du langage de programmation dépend de l'âge des étudiants. Les plus jeunes élèves auront besoin de plus d'aide visuelle pendant le processus de programmation et de débogage. C'est pourquoi les paragraphes suivants décrivent comment mettre en œuvre plusieurs concepts de codage importants, en utilisant la programmation par blocs et par texte.

<Variables>

Les variables sont utilisées pour conserver les valeurs en vue d'une utilisation ultérieure. Une bonne façon d'examiner les variables est de les imaginer comme des boîtes. Une boîte contient une valeur, elle a une certaine forme qui dépend du type de valeur (entier, flottant, chaîne, booléen, c'est-à-dire vrai ou faux), et une étiquette y est attachée contenant le nom de la variable. Nous vous suggérons d'utiliser des noms explicites pour les variables, comme la température au lieu de la lettre t, afin que d'autres personnes puissent mieux comprendre vos programmes. Dans les langages orientés bloc (Scratch^[11], Snap!^[12], MakeCode^[13]), une variable ressemble à une étiquette et vous pouvez regarder le nom et la valeur même pen-

dant l'exécution d'un programme (Ⓜ5). Ceci est très utile pour le débogage (recherche d'erreurs). Il existe également des blocs très clairs pour définir ou modifier la valeur d'une variable.

température 23

Ⓜ 5

<Affectations>

Dans certains langages de programmation orientés texte, les variables doivent être déclarées. La déclaration contient des informations sur le type de variable (entier, chaîne, ...). D'autres langages orientés texte attendent la première affectation pour décider implicitement du type de la variable.

Et l'affectation peut être trompeuse : elle est écrite comme une équation mathématique. « température = 23 » est une affectation et signifie que la température variable reçoit la valeur 23 (Ⓜ6). Si vous voulez comparer la température avec la valeur 25, écrivez « température == 25 », en utilisant deux signes égaux. Cette différence est à l'origine de nombreuses erreurs dans le développement logiciel.

afficher la variable température

température prend la valeur 23

Ⓜ 6

<Flux du programme>

La plupart des programmes informatiques consistent en des commandes qui sont traitées successivement, c'est-à-dire l'un après l'autre. Certes, de nombreux langages incluent le



traitement parallèle, où deux programmes ou plus s'exécutent en même temps dans des threads – même Scratch^[11] ou Snap!^[12] offrent ce traitement parallèle.

Mais nous allons maintenant nous pencher sur un processus unique. Les instructions sont exécutées du premier bloc ou de la première ligne au dernier bloc ou à la dernière ligne. C'est ce qu'on appelle une séquence. Mais les choses ne sont généralement pas aussi simples : vous voulez que le programme exécute différentes commandes en fonction des décisions prises à certains moments du programme. C'est ce qu'on appelle la ramification.

<Branchements>

Il y a trois types de branches : unilatérale, bilatérale et multilatérale, et elles ont toutes besoin d'une condition pour indiquer ce que vous voulez que le programme fasse. Cette condition est généralement une comparaison entre les valeurs. Le résultat est un vrai/faux booléen. La branche unilatérale ajoute simplement plusieurs instructions supplémentaires au flux du programme qui sont exécutées lorsque la condition devient vraie. La branche bilatérale ajoute deux ensembles d'instructions supplémentaires à partir desquels un seul ensemble est exécuté, en fonction du résultat de la condition. La branche multiface prend une variable, et selon la valeur de cette variable, différents ensembles d'instructions sont exécutés. Les langages orientés blocs permettent une bonne visualisation de la condition et des jeux d'instructions. Les langages orientés texte utilisent des énoncés if- ou if-else pour les deux premiers types de branches et des énoncés de cas pour les branches multilatérales.



📷 7



📷 8

<Boucles>

Les boucles, qui sont utilisées pour les ensembles d'instructions exécutés à plusieurs reprises, sont un autre instrument important utilisé pour contrôler le flux des instructions. Les boucles peuvent être distinguées par la condition qui contrôle la boucle. La condition – habituellement une comparaison –

peut être au début ou à la fin de la boucle. Si elle est au début et fausse, les instructions de boucle ne seront jamais exécutées. Si la condition est à la fin de la boucle, les instructions de boucle seront exécutées au moins une fois. Les boucles de comptage (parfois appelées boucles for) sont également utilisées lorsqu'il est clair combien de fois les instructions de boucle doivent être exécutées.



📷 9



📷 10

Les séquences d'instructions, les boucles et les branches sont normalement imbriquées. Cela signifie qu'une boucle peut être à l'intérieur d'une branche, et qu'une branche peut être à l'intérieur d'une boucle. Il existe des règles sur la façon de documenter le déroulement du programme dans un diagramme. Dans ce livret, nous utilisons des diagrammes Nassi-Shneiderman^[14] pour expliquer le flux des instructions dans nos programmes.

<Références>

- [1] www.arduino.cc
- [2] www.arduino.cc/en/Main/Software
- [3] <https://calliope.cc/en>
- [4] www.microbit.co.uk/home
- [5] http://wiki.seeedstudio.com/Grove_System/
- [6] www.raspberrypi.org
- [7] www.lego.com/en-us/mindstorms
- [8] <https://en.wikipedia.org/wiki/LabVIEW>
- [9] www.lejos.org
- [10] www.hackster.io
- [11] <https://scratch.mit.edu>
- [12] <https://snap.berkeley.edu>
- [13] <https://makecode.calliope.cc/?lang=en>
- [14] https://en.wikipedia.org/wiki/Nassi-Shneiderman_diagram

<L'enseignement de l'informatique avec Snap!>

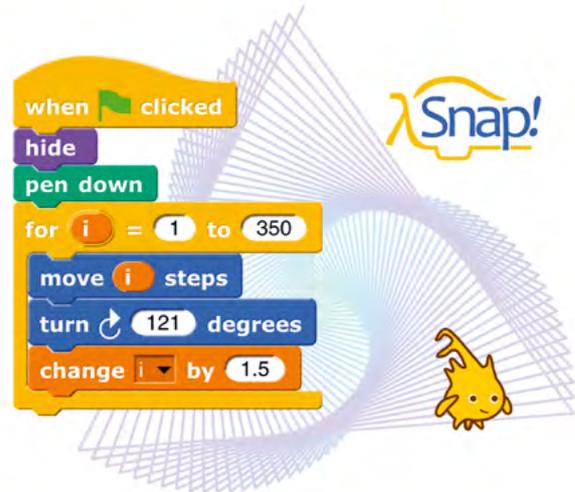
snap.berkeley.edu/run

La créativité ainsi que la maîtrise de l'informatique et des médias sont considérées comme des compétences importantes dans la révolution numérique en cours. *Snap!* est un outil qui aide les personnes de tout âge et de tout milieu à entrer en contact avec l'informatique (Computer Science : CS) et à façonner activement les développements actuels.

<Qu'est-ce que Snap! ?>

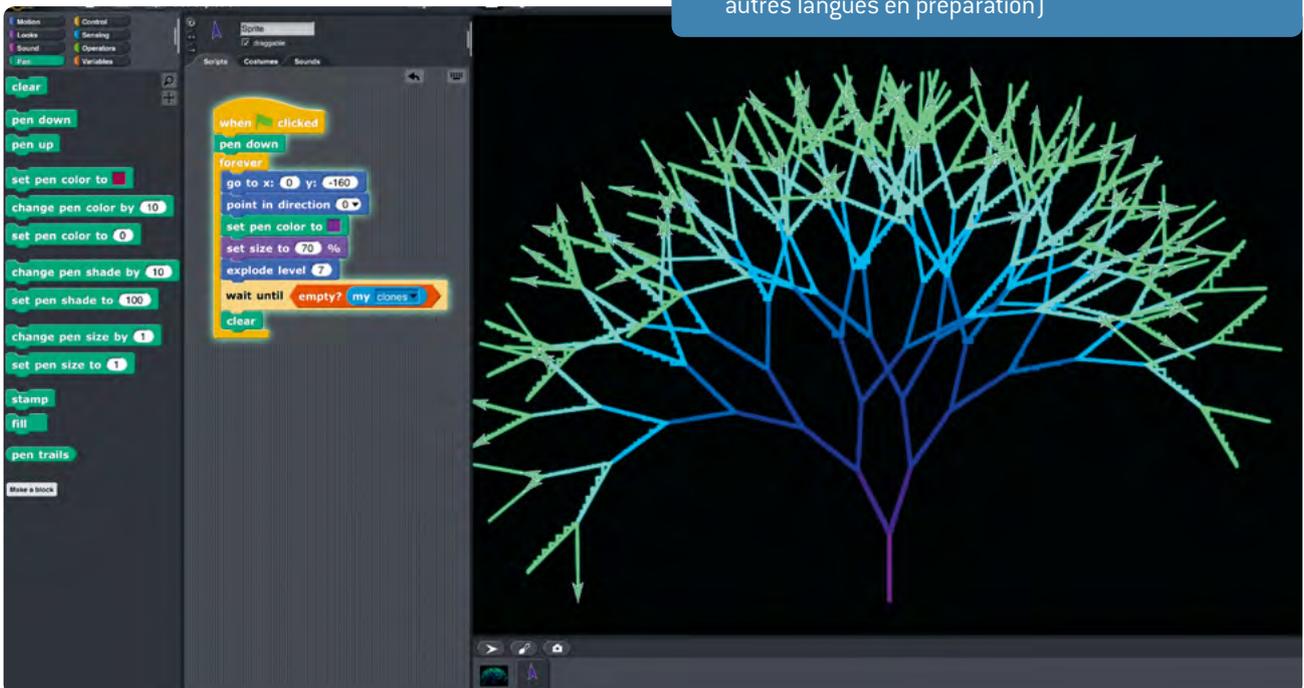
Snap!-Build your own Blocks est un langage de programmation visuel basé sur des blocs. Il invite les apprenants à donner vie à leurs idées tout en apprenant à connaître l'informatique de façon expérimentale et amusante. Ce qui est fascinant à propos de *Snap!* est son aspiration à offrir un niveau d'accès aisé tout en ne réduisant pas son expressivité. Il permet aux programmeurs débutants et expérimentés de s'immerger dans des concepts CS avancés tels que des structures de données arbitraires, des fonctions d'ordre supérieur et même des structures de contrôle personnalisées, d'une manière visuellement attrayante et compréhensible.

Snap! est développé chez SAP en collaboration avec des chercheurs de UC Berkeley. Aujourd'hui, il est disponible dans plus de 40 langues et utilisé pour l'éducation CS dans autant de pays. *Snap!* est open source et fonctionne dans tous les navigateurs web modernes.



<Vous saviez que Snap!...>

- ↳ est l'un des 100 premiers langages de programmation de l'index TIOBE
- ↳ est utilisé pour l'intelligence artificielle par des chercheurs de l'Université d'Oxford
- ↳ est utilisé par les concepteurs d'impression 3D, de broderie et de robotique.
- ↳ est facile à enseigner, par exemple avec le cours gratuit sur open.sap.com/courses/snap1 (en anglais, autres langues en préparation)



<Participer à Meet and Code>

Obtenez le soutien dont vous avez besoin pour organiser et promouvoir le codage !

« Je ne savais pas que programmer était si facile ! » *Meet and Code 2018* a été une véritable révélation pour Philip. Je veux vraiment continuer, a dit Louisa après avoir assisté à un hackathon. Et c'est exactement ce que l'initiative veut montrer ; la programmation est amusante et facile à apprendre.

Au cours de la deuxième année de *Meet and Code*, plus de 52 000 garçons et filles ont participé à plus de 1 100 événements dans 22 pays. Comme toujours, l'action s'est déroulée pendant la semaine du Code de l'UE en octobre.



© Peter Böhmer

Chaque événement audité jugé digne d'être financé a bénéficié d'un capital d'amorçage pouvant aller jusqu'à 500 euros. Toutes sortes d'événements de programmation peuvent être soutenus et n'importe quelle organisation à but non lucratif peut faire une demande avec un projet.

L'initiative vise à poursuivre les succès de l'année dernière en 2019 : de nombreux événements, projets et ateliers auront lieu pendant la semaine du Code de l'UE dans 22 pays. L'objectif est d'initier les enfants et les jeunes de 8 à 24 ans au monde



© Dietrich Bechte1



© Dietrich Bechte1

de la technologie et du codage. Les événements sont conçus pour montrer aux jeunes à quel point le codage peut être amusant et comment il peut aider à donner vie aux idées. En explorant un large éventail de sujets technologiques et numériques et de codage créatif, ils seront encouragés à développer les compétences numériques dont ils ont besoin dans le monde d'aujourd'hui.

L'organisation munichoise Haus des Stiftens gGmbH avec son portail informatique Stifter-helfen et les partenaires nationaux respectifs du réseau TechSoup Europe sont derrière cette initiative. *Meet and Code* est rendu possible grâce à SAP.

En 2019, des prix seront à nouveau décernés pour les idées d'événements les plus créatives et les réalisations les plus originales. Le *Meet and Code Award* sera décerné dans au moins trois catégories, dont l'innovation et la diversité.

Toutes les activités, informations et inscriptions au

www.meet-et-code.org

🐦 Suivez-nous et rejoignez la discussion :

@stifter_helfen @TechSoupEurope

#meetetcode #codeEU #SAP4Good

<Autres ressources>



Les auteurs ont créé des ressources et du matériel supplémentaires pour les unités d'enseignement. Vous pouvez les trouver en ligne en téléchargement gratuit sur

www.science-on-stage.de/coding-materials

<Déroulement du projet Coder dans l'enseignement des sciences>



SCIENCE ON STAGE EUROPE

Science on Stage -

Le réseau européen des professeurs de sciences

- ... est un réseau d'enseignants en sciences, technologie, ingénierie et mathématiques (STEM) de tous les niveaux scolaires.
- ... fournit une plate-forme européenne pour l'échange d'idées pédagogiques.
- ... souligne l'importance de la science et de la technologie dans les écoles et auprès du public.

Le sponsor principal de Science on Stage Deutschland e.V. est la Fédération des Associations d'Employeurs Allemands dans l'Industrie métaux et électriques (GESAMTMETALL), avec son initiative think ING.

Science on Stage France

www.scienceonstage.fr

Science on Stage Europe

www.science-on-stage.eu

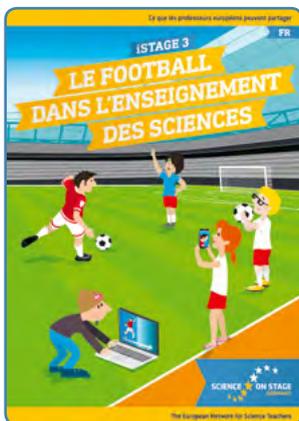
www.facebook.com/scienceonstageeurope

www.twitter.com/ScienceOnStage

Abonnez-vous à notre newsletter

www.science-on-stage.eu/newsletter

<Autres Brochures>



Le football dans l'enseignement des sciences

- ↳ Unités d'enseignement sur les différents aspects des STEM dans le football
- ↳ Chapitres : Biosphère, corps, ballon, grandes données



Smartphones dans l'enseignement des sciences

- ↳ Lignes directrices et expériences pour l'apprentissage basé sur la recherche avec les smartphones



Lilu's House - Compétences linguistiques par l'expérimentation

- ↳ Les élèves du primaire découvrent les phénomènes scientifiques naturels dans la salle de bains, le salon et la cuisine en s'exerçant à parler, à écrire et à lire.



Téléchargement gratuit à l'adresse suivante
www.science-on-stage.eu/teachingmaterials

Un projet de



Sponsor Principal de
Science on Stage Deutschland e. V.



Parrainé par



www.science-on-stage.eu