

# Спасителен план. за растенията през Ваканцията (СПРВ)

<Автор> Андреас Майер

<Автор> Соня Ван дер Бил



**<Инфо>**

**<Ключови гуми>** автоматизирано поливане на растения с помощта на микроконтролер за управление на водна помпа

**<Дисциплини>** компютърни науки, естествени науки, технологии

**<Възраст на учениците>** 10–14

**<Хардуер>** компютър (по един за всеки ученик, ако е възможно), Calliope mini<sup>[1]</sup> със сензор за влажност и температура (един за групата)

**<Езици>** Scratch<sup>[2]</sup> (online или offline), редактор за Calliope<sup>[3]</sup> (online)

**<Ниво на програмиране>** начално

**<Резюме>**

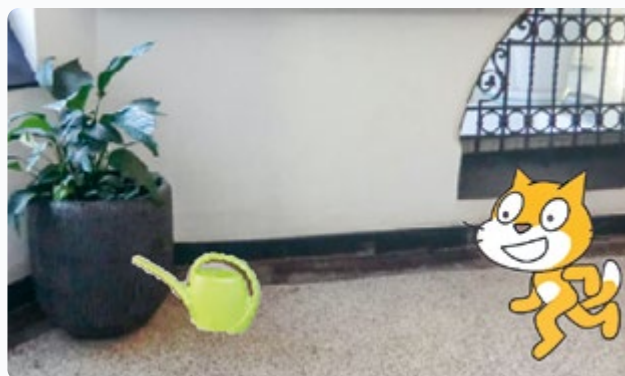
Саксийните растения в училищните сгради често умират по време на лятната ваканция, защото никой не се грижи за тях. За да решим този проблем, имаме нужда от спасителен план за растенията през ваканцията. В тази част ще разработим виртуален и реален проект за спасение на училищните растения.

**<Въведение>**

Проектът е подходящ за всички STEM дисциплини, тъй като изискваното ниво за програмиране е начално. В първата част, поливането на растенията през равни интервали, се реализира във виртуални условия. Това ще обхване различни контролни структури (условни конструкции) в компютърното моделиране, като обектно ориентиране, цикли и условия, както и използването на променливи. Учениците ще направят своята първа лесна програма, като ще използват разработен онлайн урок в „Първи стъпки със Scratch“<sup>[4]</sup> (продължителност: 45 минути).

Обектно-ориентираното програмиране (ООП) е вид програмиране, което дава на обектите свойства (атрибути) и възможности (методи). В нашия случай тези обекти са котка, наречена „Спрайт“, лейка и сцена. Всеки обект принадлежи към съответен клас. Всички обекти от един и същи клас имат еднакви свойства и възможности. Можете да тълкувате класа като проект, а обекта като представител, т.е. конкретният изпълнител на проекта. В Scratch фигурите са представители на клас „Спрайт“ или накратко „спрайтове“. Котката, наречена „Спрайт“, е екземпляр от класа „Спрайт“.

Един от атрибутите на спрайт е неговият костюм; в този случай това е образът на котка. Друг екземпляр от класа „Спрайт“ може да бъде изобразен като човек и да бъде наречен Гюнтер. Котката „Спрайт“ и човекът „Гюнтер“ са и обекти (примери) от класа „Спрайт“ или накратко: и двата са спрайтове. В Scratch има само два класа: сцена и спрайтовете. В този проект имаме два спрайта (котката и лейката) и сцената (Ⓢ1).



Ⓢ 1: Виртуален план за спасяване на живота на растенията през ваканцията

Към втората част на проекта може да се пристъпи след завършване на първата част, а може да се изпълни и самостоятелно, ако учениците познават споменатите по-горе структури за управление и вече са придобили начален опит в програмирането с Calliope mini<sup>[1]</sup>. Вместо променливи величини ще се използват сензори за микроконтролер, които ще управляват клапана на водната помпа.

**<Какво правят учениците/учителите>**

Всички необходими материали и работни листове са на разположение за изтегляне.<sup>[5]</sup>

**<Част 1: Виртуално реговно поливане на растенията>**  
**Стъпка 1: Програмиране на програма само с една променлива, време; запознаване цикли с едно условие (продължителност: 180 минути.)**

След анализиране на проблема (Как може виртуално да се спаси животът на растенията през ваканцията?), учениците ще бъдат помолени да помислят за основната структура на такава програма. Те трябва да си водят бележки под формата на рецепта (алгоритъм) и да тестват идеите си взаимно. След това те ще трябва да постигнат съгласие за общата структура на програмата (виж Работен лист 1<sup>[5]</sup>).

Този етап ще помогне на учениците да мислят за основната структура на програмата, която искат да програ-



мират. Ключовите думи „списък с инструкции (команди)“, „цикъл“ и „условие“ са извлечени от контекста.

Сега учениците ще реализират програмата в Scratch<sup>[2]</sup>, като сглобят отделните компоненти<sup>[5]</sup>, обединени в работна програма. Те ще научат повече за Scratch в процеса на работа и по-специално следното:

- ↳ ориентация на обекта (всяка фигура има собствен скрипт, дори и сцената);
- ↳ структура (Как изглежда структурата блоковете за условие или цикъл в Scratch?);
- ↳ сценарий / костюм / и звуци могат да бъдат зададени на всяка отделна фигура.

Освен това, учениците ще научат повече за основната структура на програмата за поливане на растения, докато обмислят как най-добре да подредят отделните части от кода, за да може програмата да работи. Особено важно е да решите кои условия трябва да бъдат вътре в цикъла за повторение (⌚ 2 и 3); това може да стане чрез опит и грешка.

```

when I receive Please start walking!
  go to x: 0 y: 0
  repeat 8
    move 10 steps
    switch costume to costume2
    wait 0.5 secs
    move 10 steps
    switch costume to costume1
    wait 0.5 secs
  broadcast Arrived!
    
```

⌚ 2

```

when I receive Please start walking!
  broadcast Arrived!
  switch costume to costume2
  repeat 8
    wait 0.5 secs
    move 10 steps
  switch costume to costume1
  move 10 steps
  go to x: 0 y: 0
    
```

⌚ 3

Въз основа на програмата, дадена в предходната част, учениците вече ще могат да създадат своя собствена програма, в която котката „Спрайт“ се придвижва към растенията и ги полива, според променливата – време. Единственият файл, който те ще получат, е със стартовата сцена.<sup>[5]</sup>

Учениците ще бъдат насърчавани да изследват езика за програмиране самостоятелно, за да изпробват собствените си идеи и да бъдат креативни. Важно е да могат да използват необходимия език за програмиране с увереност (в съответствие с нивото на знания); по този начин ще им бъде по-приятно.

**Стъпка 2: Напишете програма, която включва променливите „ниво на водата“ и „температура“ (необходимо време около 270 минути).**

Като въведение към втората стъпка, учениците трябва да помислят за други фактори, които определят колко често трябва да се полива едно стайно растение. „Стайната температура“ и „нивото на водата“ в контейнера за растения със сигурност ще играят роля тук, като променливи величини. (виж Работен лист 2<sup>[5]</sup>).

Учениците ще получат работна програма, в която програмирането на котката „Спрайт“ и напояването могат да бъдат почти същите, както преди.

Въпреки това ще има нов код за етапа, който контролира променливата „ниво на водата“ на мястото на предишния таймер. Котката ще полива растението само веднъж. Учениците ще бъдат насърчавани да мислят за това, как променливата „ниво на водата“ може да бъде определена и как тя да контролира дейността на котката. От друга страна, те ще имат задачата да разрешат проблема, така че растението да се полива само веднъж. Има наличен помощен файл, ако е необходимо.<sup>[5]</sup>

Използвайки само една променлива, програмата ще бъде ясно структурирана. Ще е твърде предизвикателно за начинаещ програмист да координира няколко променливи наведнъж.

Използваната структура на цикъла е по-сложна от преди, тъй като е вързана към състояние (ниво на водата) (⌚4). Учениците ще трябва да помислят внимателно какво трябва да се повтаря, колко често и при какви условия.

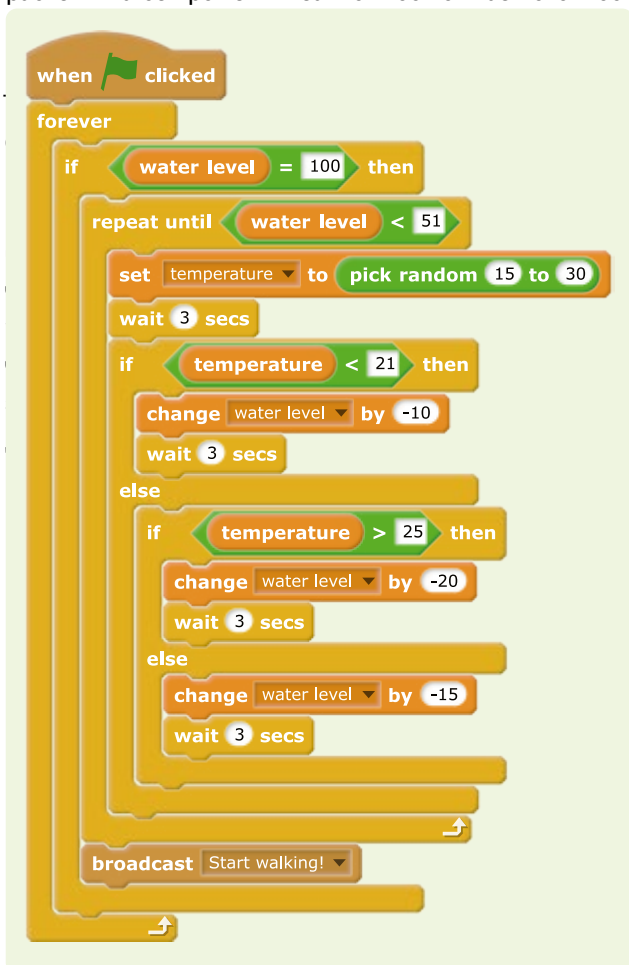


4

Способността за правилно поддръжане на програмата (структуриране) е от съществено значение при изучаването на всеки програмен език, но решавайки дадената задача ще се учи по много по-забавен начин; учениците ще могат да изпробват всичко без негативни последици.

По-бързо възприемащите ще имат възможност и да променят програмата и да изпробват собствените си идеи в края на този работен етап.

В следващата стъпка променливата „температура“ ще играе роля в програмата. Стойността ѝ се определя от генератор на произволни числа, който подава число между 15 °C и 30 °C. Нивото на водата в контейнера за растенията се променя в зависимост от тази стойност



5

Както и в предишната стъпка, на учениците ще бъде предоставена възможност да персонализират получената програма, като работят според техните идеи и възможности за програмиране.

В края на първата част на проекта, където програмата за напояване на растенията през ваканцията е реализирана във виртуални условия, ще се представят много и разнообразни резултати, за да се проверят идеите на учениците и да се докаже тяхната ефективност.

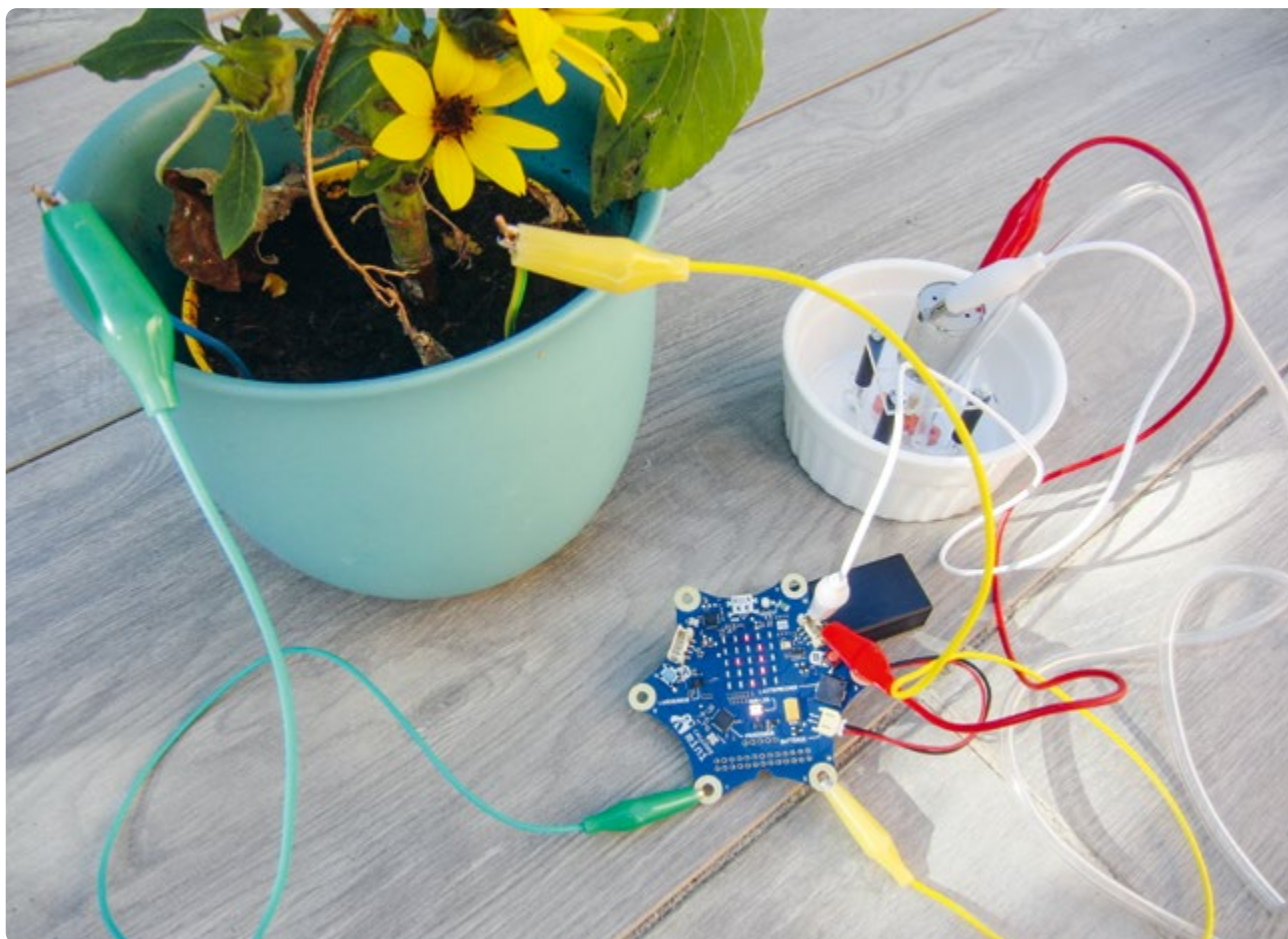
### <Част 2: Реговно поливане на растение, контролирано от микроконтролер>

По време на проекта бързо стана ясно, че някои ученици не са доволни от виртуалното решение на проблема с поливането. Те се интересуваша от начините за поливане на истински растения с помощта на тяхната програма. Програмата за виртуално поливане е относително лесно да се прехвърли в реални условия с помощта на микроконтролер, особено след като много от тези мини-компютри също могат да бъдат програмирани с Scratch<sup>[2]</sup> или подобно приложение. В нашия проект използваме Calliope mini<sup>[4]</sup>, микроконтролер подобен на BBC micro: bit<sup>[6]</sup>, който съдържа и удобни за потребителя „plug-and-play функции“ като сензори и връзки. Въпреки това СПРВ може да се контролира и с всички други микроконтролери, които често се използват в училищата, като LEGO EV3, LEGO NXT, Arduino, Raspberry Pi, Teensy и др.<sup>[5]</sup>. Програмирането на Calliope mini е лесно, тъй като трябва само да го свържете към компютъра чрез USB кабел. Open Roberta lab<sup>[7]</sup>, е онлайн среда за програмиране, която поддържа различни микроконтролери (налични са алтернативни редактори<sup>[5]</sup>), с подходящ интерфейс за програмиране. Интерфейсът на програмиране е достъпен на различни езици и можете да го смените, като кликнете върху иконата на глобус, след като сте избрали вида микроконтролер, в нашия случай на Calliope mini.

Една проста версия на СПРВ може да работи както следва:

1. Влажността на почвата се измерва постоянно.
2. Ако почвата е твърде суха, се изпомпва определено количество вода, докато почвата се овлажни достатъчно.

Следователно микроконтролерът трябва да може да измерва влагата на почвата и да контролира двигателя на водната помпа.



© 6: СПРВ контролиран от Calliople mini

Calliople mini има четири сензора за докосване. Те измерват електрическата проводимост между точките на свързване. Тъй като водата провежда електричество, влажната почва има по-висока проводимост от сухата почва. Можете да използвате два медни проводника, като сензори, които поставяте в саксията за цветя на известно разстояние. След това те се свързват с щипки тип крокодил към мини Calliople с контактите в ъглите (-, P1). Изходната стойност на сензора P1 (аналогов пин) ще бъде между 0 и 1023. Ако проводимостта намалее под определена стойност, помпата, която полива растението, ще се активира (вижте ©6).

Основен компонент на помпата е малък електродвигател, който се свързва към Calliople mini директно или с помощта на допълнителен драйвер на двигателя, в зависимост от необходимото ниво на мощност. В менюто за действие / движение са налични два порта за двигатели (A, B). Количеството на водата в помпата може да се регулира чрез промяна на стойността „скорост в %“.

Разбира се нужно е определено ниво на умения за конструиране на помпата и за изграждането на двигателя. <sup>[5]</sup> Помпата струва само няколко евро.

Докато работят с истинския спасителен план за растенията през ваканцията, учениците ще предложат множество идеи как да го оптимизират, което ще задълбочи знанията им в процеса на работа. Например:

- ↳ Нашите растения нуждаят ли се от много вода или само от малко? Колко голям трябва да е резервоарът?
- ↳ Кое е най-доброто време за поливане на растението? По-добре ли е да поливате растенията много веднъж на ден или пък по-малко, но няколко пъти на ден?
- ↳ На каква дълбочина трябва да се поставят сензорите за влажност в почвата, за да се осигурят оптимални измервания? Какво е идеалното разстояние между сензорите?
- ↳ Колко дълго трае захранването на СПРВ? Може ли да се повиши енергийната ефективност на помпата, така че напояването да е възможно за цялата ваканция в училище?

Както можете да видите, проектът СПРВ предлага на учениците различни подходи от областта на биологията и физиката за бъдещи експерименти или работа по

проекти. Друг интересен вариант би бил да свържете СПРВ към Интернет и да наблюдавате онлайн (Internet of Things). Въпреки, че това би надхвърлило нашето въведение в програмирането с СПРВ, то показва какви резултати могат да се постигнат, като задаваме обикновени въпроси.

#### <Трансфериране към други програмни езици>

Проектът може лесно да бъде прехвърлен на Snap!<sup>[8]</sup>, което е по-нататъшно развитие на Scratch<sup>[2]</sup>. Примерите за програмиране са предоставени онлайн<sup>[5]</sup>. Тези два езика за програмиране са особено подходящи за проект, който е насочен към начинаещи програмисти, тъй като те са лесни за разбиране и насърчават учениците да изпробват идеи, като използват „плъзгане и пускане“.

#### <Заключение>

Ученици, които са нови в програмирането, ще направят първите си стъпки в тази област и ще научат важни основи на език за програмиране при изпълнение на този проект, който представя реална ситуация. Фокусът не е върху изучаването на синтаксиса на програмния език, а върху изпробването на ефекта на различни контролни структури: „Какво работи и защо?“.

Грешките са добре дошли, защото обикновено са лесни за намиране и обяснение и по този начин помагат на учениците да разберат как работи езикът за програмиране.

От една страна, дадената рамка дава на учениците сигурност (който не е сигурен, ще реши само пъзела за сглобяване на предоставените части на програмата), а от друга страна, има достатъчно място за творчество за онези ученици, които изпълняват „задължителните задачи“ бързо.

Някои идеи се появиха в резултат от проекта, напр. да се изгради „истински“ робот за поливане или да се разработи игра за поливане. Във всеки случай учениците придобиха положителен начален опит в програмирането, който се надяваме да има траен и устойчив ефект върху тях.

Времевата рамка, определена в нашето училище, понякога беше трудна. Имахме само 45 минути на урок да работим по този проект. Самата организационна част на уроците (влизване в компютъра, отваряне на файлове, запазване на файлове, излизане от компютъра) отнемаше 15 минути, така че нямаше достатъчно време за реално програмиране и работа. Можете да поискате

достъп в Scratch, като учител<sup>[2]</sup>, което ще ви позволи да създадете клас и да споделяте материали.

#### <Сътрудничество>

Тъй като СПРВ е въведение в програмирането, има много малко възможности за сътрудничество.

Веднага след като проектът се приложи в реалния живот с помощта на микроконтролерите, сътрудничеството между учениците става много полезно, тъй като степента на трудност на проблема се увеличава при използването на допълнителни компоненти (сензори, помпа). Например, по-големите ученици могат да помогнат за изграждането на помпата. Училища в различни страни могат да работят заедно по проекта СПРВ и да сравняват своите резултати и решения.

Може да се създаде обща база данни за различните растения, за да се адаптира СПРВ към индивидуалните характеристики на различни видове растения. Ако СПРВ е свързан с интернет, училищата могат да приемат растения от друго училище и да поемат отговорността за поливането му.

#### <Препратки>

[1] [www.calliope.cc/en](http://www.calliope.cc/en)

[2] [www.scratch.mit.edu/](http://www.scratch.mit.edu/)

[3] [www.calliope.cc/en/los-geht-s/editor](http://www.calliope.cc/en/los-geht-s/editor)

[4] [https://scratch.mit.edu/projects/editor/?tip\\_bar=getStarted](https://scratch.mit.edu/projects/editor/?tip_bar=getStarted) (29/11/2018)

[5] всички допълнителни материали са достъпни на [www.science-on-stage.de/coding-materials](http://www.science-on-stage.de/coding-materials).

[6] [www.microbit.co.uk/home](http://www.microbit.co.uk/home)

[7] <https://lab.open-roberta.org>

[8] <https://snap.berkeley.edu/>

## <Печат>

### <Публикувано от>

Science on Stage Deutschland e. V.  
Ам Борсигтурм 15  
13507 Берлин, Германия

### <Главен координатор>

↳ **Д-р Йорг Гуцанк**, Гимназия Лайбниц | Международно училище в Дортмунд, Дортмунд, Германия  
Председател на Science on Stage Deutschland e. V.

### <Координатори>

- ↳ **Себастиан Фънк**, Вила Виверсбуш, Велберт – Лангенберг, Германия, Член на съвета на Science on Stage Deutschland e. V.
- ↳ **Жан – Люк Рихтер**, Гимназия Jean-Baptiste Schwilgué, Селестат, Франция, Заместник-председател на Наука на сцената Франция
- ↳ **Бернар Шриек (ret.)**, Гимназия Marien, Верл, Германия

### <Цялостна координация и редактиране>

- ↳ **Даниела Нюман**, Проектен мениджър на Science on Stage Deutschland e. V.
- ↳ **Стефани Шлунк**, Изпълнителен мениджър на Science on Stage Deutschland e. V.
- ↳ **Йохана Шулце**, Заместник изпълнителен мениджър на Science on Stage Deutschland e. V.

### <Корекция и превод>

Мария Петрова, Петър Андреев, Десислава Цокова и Моника Ковачка-Димитрова

### <Дизайн>

WEBERSUPIRAN.berlin

### <Илюстрации>

Рупърт Таке, TricomKommunikation und Verlag GmbH

### <Авторски права>

Всички аспекти на авторското право за изображенията и текстовете, използвани в тази публикации са проверени от авторите, доколкото е възможно.

### <С подкрепата на>

SAP SE

### <За поръчка>

www.science-on-stage.bg  
office@frgi.bg

Това издание е лицензирано от Creative Commons Attribution-ShareAlike 4.0 International License:  
<https://creativecommons.org/licenses/by-sa/4.0/>.



Първо издание 2020

© Science on Stage Deutschland e. V.