

Научна магическа кутия

<Автор> Люк Ивара

<Автор> Марко Николини



<Инфо>

<Ключови гуми> микроконтролер, датчик, сензор, преобразувател, изходно устройство, сигнал, физична величина, цикъл, разклонение, последователност, процес, калибриране, вход, изход, четене, писане, аналогов, цифров, линейност, преобразуване, дъска, щифт, запояване, интерфейс човек-машина

<Дисциплини> физика, електроника, математика, ICT, логика, биология

<Възраст на учениците> 14–18

<Хардуер> Arduino UNO^[1] с Arduino DUE^[2] и/или TI-Nspire CX CAS с TI-Innovator Hub

<Език> C++ (използване Arduino IDE^[3]) и/или TI-Basic

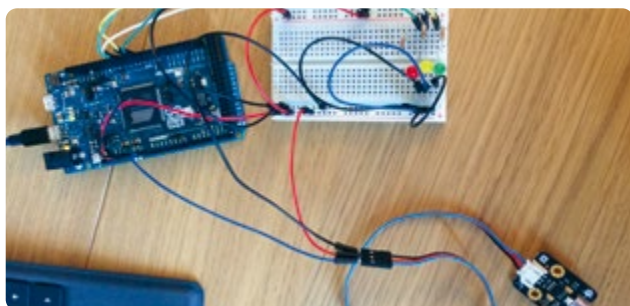
<Ниво на програмиране> средно, с аудио част за напредналите ученици

Списък на абривиатурите, специалните термини и съкращенията, използвани в този раздел, е достъпен онлайн.^[4]

<Резюме>

Учениците ще научат как да програмират самостоятелно изградена хардуерно-софтуерна среда (базирана на Arduino) и готов джобен компютър (калкулатор TI-Nspire CX CAS с разширението му, TI-Innovator Hub). И двете се използват като устройства за събиране и преобразуване на данни от сензори за лесно управление, четене, конвертиране и изпълнение.

Използвайки платформата Arduino, учениците ще програмират сензори, които регистрират информация за физични параметри и я подават, като входни сигнали към Arduino, там те задействат преобразувателя, който реагира на получения входен сигнал и поражда изходен сигнал. След като микроконтролерът е обработил входния сигнал, той го изпраща към правилния изход за изпълнение от изходни устройства (виж ☺1).



☺ 1: Платка Arduino

TI-Innovator Hub е кутия „готова за употреба“, която дава възможност на учениците да научат основите на програмирането. Трябва да бъде включена в калкулатор TI-Nspire CX CAS. Той има добър I/O интерфейс, който включва сензор за светлина, два светодиода и вграден зумер, който издава звук с определена честота (вижте ☺2).



☺ 2: TI-Nspire u TI-Innovator Hub

<Въведение>

Този раздел запознава учениците със света на програмирането за решаване на физични проблеми, въз основа на откриване и измерване на физични величини, обработка на данните, реакция и вземане на решение за извършване на последващо действие от изпълнителните устройства.

Кодът обикновено се основава на безкраен цикъл (машината обикновено е „жива“, докато е захранвана и трябва да работи през цялото време), където трите действия на измерване, обработка и действие се извършват в този ред.

Учениците ще научат, че могат да напишат всеки код с помощта на:

1. последователни инструкции
2. цикли (докато ... прави; повтаряй ... докато)
3. разклонения (ако...тогава...още)

както е посочено в теоремата на Бьом-Якопини (виж „Допълнителна информация“^[4]).



3: Блок схема

Втората цел на тази част е да запознае учениците с микроконтролерите. Те ще се научат да настройват входни датчици и изходни изпълнителни устройства, използвайки цифрови или аналогови портове, да напишат проста програма, която чете вход, обработва данни и извежда изход. Те могат да избират звукови или светлинни сигнали, които да бъдат използвани като „аларми“, за да издават предупреждение въз основа на входния сигнал, прочетен от сензорите.

Учениците ще използват Arduino интегрираната развойна среда^[1] (Arduino IDE^[3]) за програмиране в C++, с готови за употреба функционални библиотеки, които правят процеса на програмиране по-лесен и бърз.

Трябва да имате на разположение и прототипна платка – Breadboard (вижте „Допълнителна информация“^[4]), за да могат учениците да изработват своите прототипи и по-лесно да свързват пиновете на сензорите към Arduino I/Os, 5V захранване и GND пин.

Структурата на всяка програма в блок схема е показана на 3.

Моля, обърнете внимание, че инструкцията „докато (TRUE)“ е трик да кажете на всеки процесор да повтаря включените инструкции за неопределено време (стига микроконтролерът да се захранва).

Третата основна цел е да научите как да преобразувате регистрираните физични параметри в други такива (например интензитет на светлината в звук), готови за предаване във външната среда, като следвате тези стъпки:

1. Физичният сигнал (светлина, звук, сила, енергия ...) се получава от сензор и се превръща в електрически сигнал.
2. Електрическият сигнал се преобразува в число, достъпно за процесора.
3. Числото се обработва и превръща от процесора в друго число и след това се използва за извършване на действие с преобразувател (преобразувател – изпълняващо устройство, което превръщаща входния сигнал (електрически, оптичен, механичен, пневматичен и др.) в изходен сигнал (светлина, звук, сила, енергия, движение), въздействащ върху управлявания обект).
4. Преобразователят превръща числата в електрически сигнали, които са готови за извеждане.
5. Накрая електрическият сигнал се трансформира във физичен сигнал (например звук, светлина).

В стъпки 1 и 5 сигналите трябва да бъдат преобразувани от една форма в друга. В тези фази линейността на трансформацията или „близката до линейността“ зависимост е изключително важна (вижте „Допълнителна информация“^[4]).

Вижте „Допълнителна информация“^[4] за подробно обяснение на последния ред от блок схемата („Мигащ бордов светодиод“).

Входният сигнал обикновено се нарича „стимул“ и идва от средата, където сензорите са поставени за получаване на данни. Процесорът и кодът са проектирани да „реагират“ на стимула с математически/логически операции (изпълнявани от инструкциите на кода, обработени от микроконтролера) и да извеждат този „отговор“ в околната среда. В нашите дейности средата е пространството около Arduino, което е в състояние да „вижда“, „чува“ и „чувства сили“, благодарение на сензорите.

Работата с TI-Innovator Hub позволява на учениците да се съсредоточат повече върху програмната част на

своята работа, тъй като микроконтролерите и сензорите вече са настроени и готови за употреба.

<Какво правят учениците/учителите>

Препоръчваме ви да започнете с брейнсторминг, за да събирате прости идеи от всички ученици относно сензорите и автоматичното управление на устройствата. Събирането на тези идеи, натрупването на практически опит със сензорите и автоматичното управление на устройствата, сравняването на резултатите с техните предишни (може би неправилни) идеи е добър начин да помогнете на учениците да разберат истински процеса.

Можете да подготвите формуляр с въпроси като:

- ↳ Знаете ли как термостат контролира температурата в помещението?
- ↳ За какво е звуковата предупредителна система за паркиране? Как реагира водачът, когато системата излъчва звук?
- ↳ Имате ли индукционен/стъклокерамичен котлон в кухнята си? Какво означава светец светодиод?

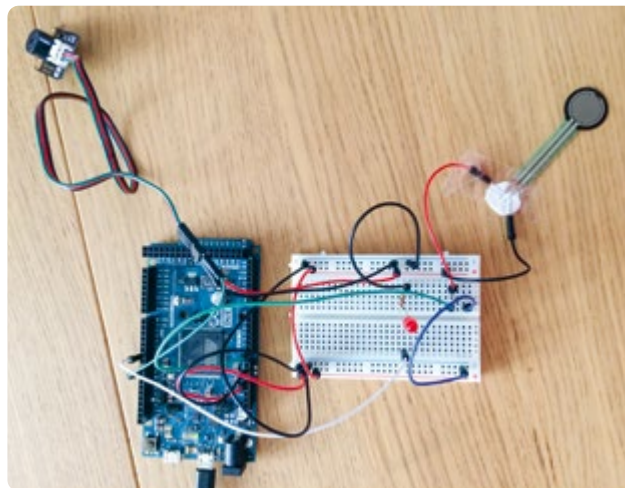
Вижте „ПОК и списък с въпроси“^[4] за вариант на пълен списък с въпроси и препратки към методологията на преподаване на ПОК (Предупреждение, опит и корекция).

<Теоретична част Arduino^[4]>

Преподавателят ще въведе C++ програмирането^[3] със структурата и основните инструкции, така че учениците да могат да напишат обикновен цикъл, използвайки инструкциите *analogRead*, *digitalRead*, *analogWrite*, *digitalWrite*, *if...then...else*, *loop*, *while*.

Хардуерна част

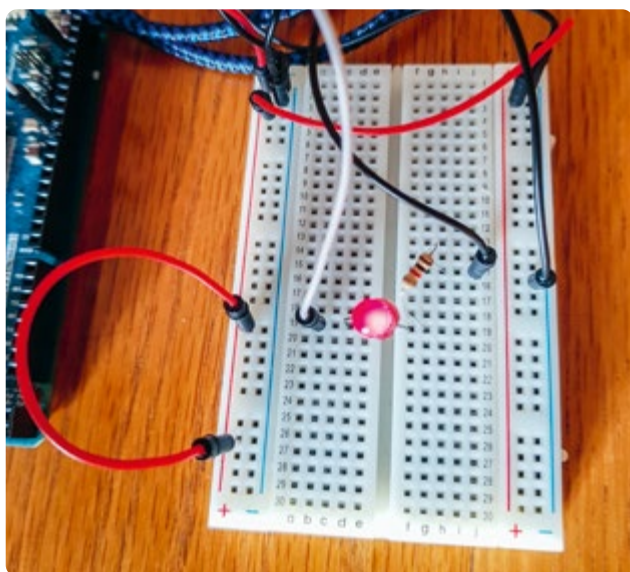
Преподавателят ще представи устройството на микроконтролера, показвайки микропроцесора, аналоговите I/O пинове (връзки) и цифровите входно / изходни пинове (връзки).



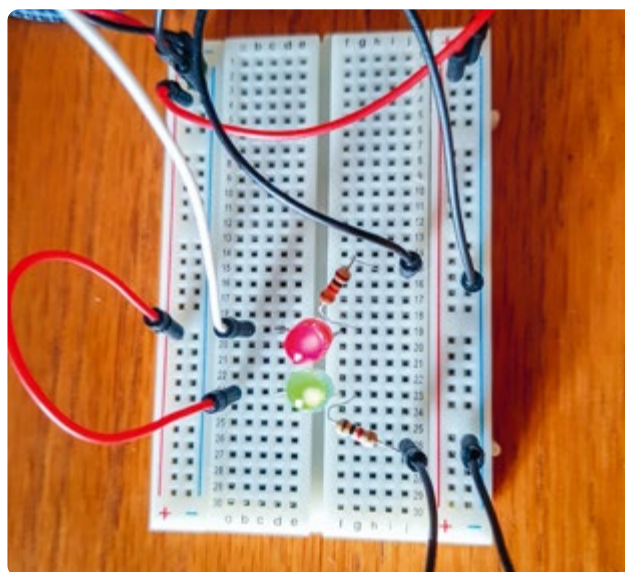
© 4: Arduino, прототипна платка и сензори

Учениците ще научат, че всеки сензор/задвижвано устройство обикновено има множество връзки:

- ↳ към 5V или 3,3V Arduino изход, за да получите захранване;
- ↳ към GND (заземяване) сигнал, за да може да тече ток;
- ↳ към друг цифров или аналогов входен щифт, ако се използва за четене (получаване) на външни данни;
- ↳ към друг цифров или аналогов изходен пин, ако се използва за предприемане на действия, генериращи изходен сигнал, например за излъчване на звук или светлина или ако правите нещо друго, което сигнализира за дадено действие (изписване на текст).



© 5: Детайл на прототипната платка



© 6: Прототипна платка с LED

Софтуерна част

Преподавателят ще представи проста програма, която чете датчик и изписва информацията на изпълняващото устройство, което дава ясна връзка между физическите пинове и физическия адрес на платката на микроконтролера и може лесно да се проследи от учениците.

Примери за инструкции, готови за употреба, са на разположение онлайн („Примерна програма 1“^[4]).

Учениците трябва да имат предвид, че процесорът изпълнява инструкциите на кода една след друга и в реда, в който са написани. Само инструкцията 'loop' променя този принцип, тъй като казва на процесора непрекъснато да повтаря инструкциите в скоби, докато микроконтролерът се захранва.

<Практическа част с Arduino>

Учениците ще добият практически умения за работа с микроконтролер, прототипна платка и сензори. Учителят трябва да представи структурата на прототипната платка, като покаже всички налични връзки и как учениците могат да получат 5V, GND и I/O сигнали от Arduino^[4] към прототипната платка. Те ще бъдат подкачени да копират дадения пример за програмиране и да изпробват, тестват и отстраняват грешки от кода със свързания I/Os.

Хардуерна част

Учениците се нуждаят от микроконтролер, сензори и къси кабели (10 см), които позволяват лесни връзки между пиновете на сензора и „дупките“ на дъската. Понякога може да се наложи да започнете допълнителни кабели към сензорите, но много сензори не изискват това.

Учениците ще използват късите кабели, за да свържат 5V захранването и GND сигнал към прототипната платка, а аналоговите/цифровите пинове на Arduino^[4] към някои „дупки“ на платката. Това позволява датчиците лесно да бъдат поставени и да получават необходимите електрически сигнали (виж ⑥6).

Софтуерна част

След като проверят дали връзките между микроконтролера и прототипната платка са правилни, с точното съответствие между логически номер на пина от микроконтролера и сензорния пин върху платката, учениците ще се опитат да напишат предоставения им прост код („Програма пример 1“^[4]).

Алгоритми с Arduino

Подготвили сме няколко преобразувания на сигнали от една физична форма в друга.

Преобразуване на аналогов светлинен сигнал в цифров светлинен сигнал на светодиода (модулиран с PWM функция) и звук: излъчваната звукова честота нараства с интензитета на светлината. Практическо приложение: будилник, помощ за незрящи хора. (Вижте ⑦7)



⑦7: Сензор за светлина

Преобразуване на сила, открита чрез сензор за сила, в цифров светлинен сигнал на светодиода (модулиран с PWM функция) и звук: интензитетът на светлината се увеличава с големината на силата. Практическо приложение: аларма за превишено тегло. (Виж ⑧8)

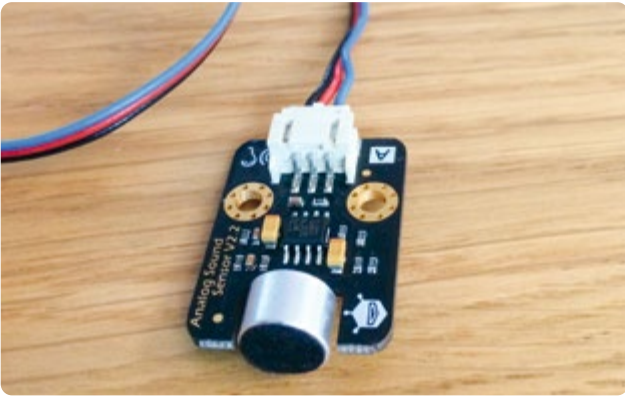


⑧8: Цифров зумер

Преобразуване на външен шумов сигнал в цифров светлинен сигнал на LED. Колкото по-висок е звукът, толкова по-висока ще бъде честотата на светлината. Практическо приложение: контрол на шумовото замърсяване.

Преобразуване на разстояние, измерено със сензор за разстояние, в звук. Техническо приложение: сензори за паркиране на автомобили. (Вижте ⑨9)

Преобразуване на температурен сигнал в звук и светлинен сигнал. Техническо приложение: контрол на температурата във фурната.



© 9: Датчик за разстояние

Преобразуване на концентрацията на влага в почвата в светлинен сигнал. Техническо приложение: напояване и контрол на напояване на растенията. (Вижте ©10)



© 10: Сензор за влажност на почвата

Вижте „Програмен пример 2“^[4] за кода, използван за създаване на тези приложения за Arduino.

Всички тези алгоритми служат, като системи за управление на сигнали и предупреждения, които наблюдават избрана среда и издават предупреждение въз основа на прочетения вход (стимул).

<Теоретична фаза с TI-Innovator Hub>

Хардуерна част

Учителят може да покаже на учениците колко лесно се свързват сензорите.

Софтуерна част

Основното програмиране може да се извърши само на калкулатора; учениците трябва да усвоят инструкциите TI-Basic, предоставени по-горе. Тогава хъбът може да бъде свързан и учениците ще се научат как да комуникират с него, т.е. да използват инструкциите „четене“ и „получаване“, за да получат данни и „настройка“, за да контролират изходите.

<Практическа фаза с TI-Innovator Hub>

Хардуерна и софтуерна част

Учениците ще започнат с основни примери, за да се запознаят с устройството, преди да работят върху собствени проблеми. Те ще се научат как да контролират различните изходи с помощта на малки упражнения, например: контролира цвета на светодиода, кара го да мига, контролира продължителността на мигането и издава звуци с дадена честота. Безкрайният цикъл отново ще бъде основната структура за продължаване на операциите за неопределено време.

Алгоритми с TI-Innovator Hub

Учениците ще решат два отворени проблема: създаване на автоматичен превключвател, който включва светлината само ако интензивността на околната светлина е по-ниска от определен праг, и създаване на будилник, който излъчва звук с все по-голяма честота с увеличаване на околната светлина. Допълнителни разработки са възможни, но ще трябва да се купят допълнителни сензори и да се включат в устройството.

<Закупуване на сензорите>

Информацията за това къде и как да закупите сензорите е достъпна онлайн.^[4]

<Закljučение>

В края на тези дейности забелязахме, че разбирането на нашите ученици за програмирането, общата програмна структура, както и логиката и алгоритмите се подобриха значително.

<Дейности по сътрудничество>

Сътрудничество може да бъде насърчаване чрез самостоятелното предприемачество. Учениците могат да измислят оригинален интерфейс за човешка машина (НМИ). Тази НМИ трябва да чете сигнали от средата (атмосфера, дом, човешко тяло и т.н.) и да реагира, като издава сигнал, който излъчва предупреждение за ситуация или извършва действие. Партньорските училища в чужбина могат да проведат пазарно проучване за търсенето и стойността на устройството в техните страни. Всяко училище, което участва в този обмен, може да измисли устройство или да направи пазарни проучвания за продукта на друго училище. В края на проекта най-популярното устройство може се произведе в малък мащаб от партньорска компания. Предприемачество е високо ценено, тъй като предлага чудесна възможност да преподаваме наука, технологии и финанси заедно.

<Препратки>

* Всички препратки са налични в английската версия.

<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.