

**STEMKIT**  
4SCHOOLS

# UTILIZAREA SENZORILOR IR ÎN SISTEME DE ALARMĂ

PLAN DE LECTIE 1



Co-funded by the  
Erasmus+ Programme  
of the European Union

This project has been funded with support from the European Commission.

This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



## Cuprins

1.	Folosirea IR senzori în sistemele de alarmă.....	2
1.1	Informatii Generale .....	2
1.1.1	Scurta descriere.....	2
1.1.2	Obiectivele invatarii .....	2
1.1.3	Links catre curriculum .....	2
1.1.4	Materiale solicitate .....	3
1.1.5	Durata .....	3
1.2	Planul de Lectie .....	4
1.2.1	Introducere.....	4
1.2.2	Pregatire .....	4
1.2.3	Investigatie.....	9
1.2.4	Concluzii .....	10
1.2.5	Follow-up - exercitiu (optional) .....	10
1.3	Referinte sau Resurse .....	10



# 1. Folosirea IR senzori în sistemele de alarmă

## 1.1 Informatii Generale

### 1.1.1 Scurta descriere

Această lecție introduce utilizarea senzorului optic reflectorizant TCRT5000 cu ieșire cu tranzistor pentru a proiecta un circuit simplu care va acționa ca o alarmă pentru ferestrele care ar fi putut fi deschise de o persoană neautorizată. Poziționarea unui astfel de senzor poate fi introdusă în reglare reală lângă rama ferestrei datorită razei de funcționare a senzorului de la 0,2 mm la 15 mm.

### 1.1.2 Obiectivele invatarii

Principalele obiective de învățare ale acestui plan de lecție sau ale activității educaționale sunt:

- Înțelegerea conceptului și a conținutului de hardware pentru a configura un cod în Scratch.
- Familiarizarea cu activitățile de construire a consolei pentru a îmbunătăți experimentarea la subiecte legate de STEM.
- Familiarizarea cu citirile din pini GPIO.
- Înțelegerea etapelor de codare din Scratch.
- Proiectarea codării în Scratch.
- Utilizarea de senzori, emițător infraroșu, fototranzistor și alte elemente pentru a experimenta cu consola STEMKIT.
- Efectuarea codării pe o consolă STEMKIT pe exemplul senzorilor IR din sistemele de alarmă.
- Efectuarea asamblării de bază a senzorilor pe o placă de calcul
- Experimentarea conectării senzorilor la consola Raspberry Pi și STEMKIT.
- Autonomie în crearea unui circuit simplu care poate servi drept demonstrație a unei instalații de securitate într-o clădire.
- Autonomie în introducerea conceptelor de codificare în mediul clasei.

### 1.1.3 Links catre curriculum

Domeniile, subdomeniile, subiectele / subiectele la care acest plan de lecție poate fi legat sunt:



Erasmus+

2019-1-FR01-KA201-062281



STEMKIT  
4SCHOOLS

Știință (Fizică / Chimie / Biologie / Geologie): tensiune, putere, circuite, declanșatoare de alarmă, metodă științifică, investigație, experimentare, analiză și interpretare a rezultatelor

Informatică / Informatică: unitate de procesare și periferice, interfețe, limbaj de programare și structuri principale, codare

Tehnologie: electronică, hardware și software open source, senzori, semnal digital, calculatoare cu o singură placă, consolă

### 1.1.4 Materiale solicitate

Pentru a realiza acest plan de lecție, este necesară consola STEMKIT cu Raspberry Pi împreună cu următoarele elemente:

- 3 senzori TCRT5000
- 1 x buzzer cu generator
- 2 x fire jumper /Female-to-Female jumper wires
- 5 x fire jumper /Male-to-Female jumper wires
- 3 rezistențe de 10k $\Omega$
- Rezistențe de 3 x 330 $\Omega$
- 1 x panou de testare /breadboard

### 1.1.5 Durata

Durata acestui plan de lecție este estimată la aproximativ 45-60 de minute, adică o oră de clasă.



## 1.2 Plan de Lecție

Planul de lecție este împărțit în patru etape, care sunt introducerea, pregătirea, investigarea și încheierea. Ca o continuare, există și un exercițiu opțional la sfârșit.

### 1.2.1 Introducere

TCRT5000 este un senzor compus din două elemente într-o singură incintă. Emițătorul emite lungimea de undă de 950 nm care urmează să fie recepționată de fototranzistor care funcționează ca detector. Cu cât obiectul este mai aproape de senzor, cu atât va fi mai mare citirea tensiunii de la fototranzistor. Deoarece Raspberry Pi oferă pini GPIO, această citire poate fi apoi urmată pentru a verifica dacă obstacolul (cadru ferestrei) este aproape de senzor și, ca urmare, dacă fereastra este închisă.

În cadrul acestei lecții, Scratch va fi folosit pentru a demonstra codul de probă care poate fi utilizat pentru a urmări acest circuit simplu.

### 1.2.2 Pregătirea

Faza de pregătire necesită efectuarea unui ansamblu de bază al senzorilor pe o placă de calcul și configurarea codului în Scratch. Să începem mai întâi cu panoul de verificare.

Plasați trei senzori TCRT5000 pe o placă goală, conectând emițătorul fototranzistorului și catodul emițătorului cu infraroșu la șina de la sol. După aceea, conectați colectorul fototranzistorului utilizând un rezistor de 10 k $\Omega$  la șina pozitivă de pe o placă de măsurare. De asemenea, conectați anodul emițătorului cu infraroșu la șina pozitivă utilizând un rezistor de 330 $\Omega$ . Repetați-l pentru restul de doi senzori TCRT5000.

Acum este timpul să conectați senzorii la Raspberry Pi. Conectați firul jumper de o lungime adecvată la fiecare senzor TCRT5000. Ar trebui să se conecteze la șina unde colectorul fototranzistorului este conectat și alimentat de o șină de tensiune pozitivă utilizând un rezistor de 10k $\Omega$ . În această etapă ar trebui să existe trei fire jumper - câte unul pentru fiecare senzor TCRT5000. Aceste fire jumper ar trebui să fie conectate la pinii GPIO de pe Raspberry Pi marcate ca 35, 33 și 31 (sau în GPIO: 19, 13 și 6). Acum este și momentul să conectați buzzerul direct la Raspberry Pi. Puteți face acest lucru atașând pinul GND al buzzerului la pinul 39 și firul său pozitiv la pinul 38 (sau GPIO: 26). În cele din urmă, panoul trebuie să primească puterea de la Raspberry Pi. În acest scop, puteți utiliza șina + 5V de la Raspberry Pi (pinul 4) și GND (pinul 6). Utilizați fire jumper pentru a porni panoul. Configurarea hardware-ului este finalizată, așa că acum putem trece la Scratch.



În interiorul Scratch, puteți selecta orice fundal care are cel puțin trei ferestre vizibile. Pentru această lecție vom folosi fundalul urban1. Setează-l pentru întreaga scenă. În același timp, trebuie să adăugați trei obiecte care își vor schimba costumele odată ce alarma este declanșată. Aici sugestia este de a utiliza un sprite care combină butonul 4-a și butonul 5-b. Codul va schimba costumele pe baza citirii de la senzori. Deoarece trebuie să aveți trei sprite (câte unul pentru fiecare fereastră), vă rugăm să le duplicați pentru a avea următoarea configurație (spritul din mijloc are celălalt costum activ pentru a arăta diferența):



**Imag. 1. Scratch - mediul cu toate elementele necesare poziționate pe ecran**

*Sursa: STEMKIT4Schools project*

Înainte de a merge la codul principal, permiteți-ne să lucrăm la cele trei sprite care sunt plasate pe ferestre.

Pentru a simula alarma, vom folosi mesajele de difuzare ale Scratch pentru a reacționa în mod corespunzător la citirile de la senzori. Când treceți la fila Scripturi, va trebui să adăugați două reacții pentru mesajele primite. Să presupunem că pentru primul senzor vom emite mesaje window1-deschis și window1-închis. Când mesajul primit este deschis la fereastra1, atunci trebuie să declanșăm alarma și să schimbăm costumul dintr-o bifă verde în cruce roșie. În mod similar, atunci când fereastra este închisă, trebuie să avem din nou costumul verde. Exemplul de cod este prezentat mai jos:



**Imag. 2. Script pentru sprite care reacționează la citirile de la pinii GPIO**  
Sursa: STEMKIT4Schools project

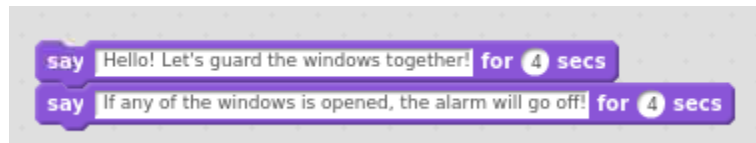
Replicați această configurație pentru restul de două sprite plasate peste ferestre și nu uitați să schimbați titlurile mesajelor (*window2-open*, *window2-closed*, *window3-open*, *window3-closed*).

Codul principal este locul în care totul devine interesant. După ce selectați pisica, comutați la fila Scripturi și începeți să adăugați codul. Primul lucru pe care am dori să-l configurăm este să anunțăm Scratch că pinii GPIO 19, 13 și 6 ar trebui să fie citiți ca pini de intrare.



**Imag. 3. Începutul codului - setarea pinilor GPIO relevanți ca intrări**  
Sursa: STEMKIT4Schools project

Apoi, puteți adăuga câteva mesaje înainte de a lansa codul principal.



**Imag. 4. Introducere mesaj executat prin cod**  
Sursa: STEMKIT4Schools project

În pasul următor, vom adăuga o buclă de repetare care va face 30 de cicluri, înțelegând ca „verificări” ale senzorilor.



**Imag. 5. Bucla de repetare principală**  
Sursa: STEMKIT4Schools project

În fiecare buclă, suntem interesați să ajungem la toți cei trei senzori pentru a vedea dacă ferestrele sunt închise sau nu. Structura este foarte simplă, deoarece putem verifica cu o afirmație if dacă citirea de la acei pini GPIO este ridicată și acționează în consecință. Mai jos veți găsi un exemplu pentru numărul PIN-ului GPIO 19.



**Imag. 6. O declarație if care citește valoarea GPIO și transmite mesaje**  
Sursa: STEMKIT4Schools project

Rețineți că singura acțiune pe care o întreprindem aici este de a emite un mesaj window1-deschis sau window1-închis pe baza valorii pinului GPIO 19. Similar dacă ar trebui adăugate blocuri pentru GPIO rămas: 13 și 6. Din nou, nu uitați să schimbați numărul pinului GPIO și, de asemenea, mesajul care trebuie difuzat!

În pasul următor dorim să ne controlăm și soneria. Condiția aici este că, dacă toate ferestrele sunt închise, alarma este silențioasă. Când se deschide cel puțin o fereastră, vom emite sunetul de alarmă și vom lăsa pisica să spună Alarmă! să aibă și o notificare vizuală. Pentru aceasta, vom folosi o instrucțiune if cu trei condiții combinate de o logică sau instrucțiune.





**Imag. 7. O declarație if pentru a declanșa alarma, dacă este necesar**  
Sursa: STEMKIT4Schools project

Chiar înainte de a finaliza bucla curentă, vom întrerupe execuția timp de 1 secundă. Acest lucru definește frecvența verificărilor noastre care trebuie executate o dată pe secundă.



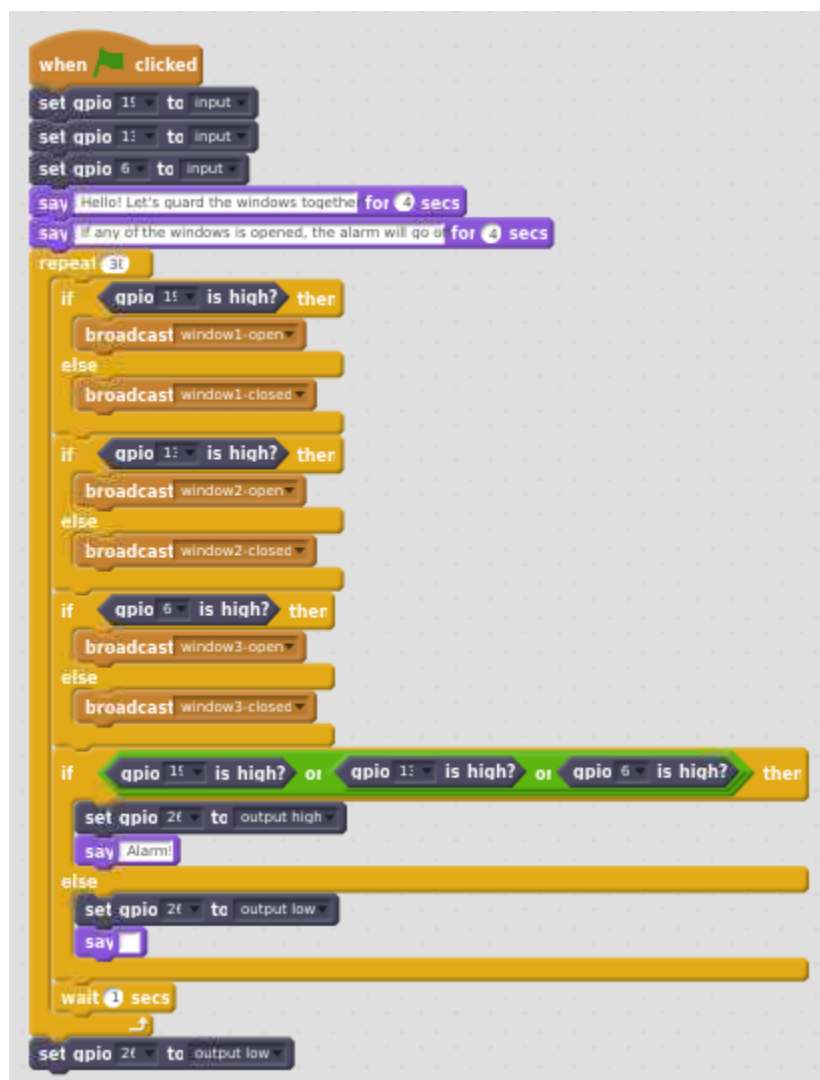
**Imag. 8. Întrerupeți executarea codului timp de 1 secundă chiar înainte de a părăsi bucla**  
Sursa: STEMKIT4Schools project

În cele din urmă, deoarece nu dorim ca sunetul să emită sunet după ce ne-am terminat tot codul, imediat după repetarea buclei, trebuie să adăugăm instrucțiunea pentru a seta pinul GPIO numărul 26 (cel cu soneria) să rămână în cel mai scăzut nivel stare (ceea ce înseamnă că nu există sunet).



**Imag. 9. Setarea GPIO-ului buzzerului la minim după bucla principală**  
Sursa: STEMKIT4Schools project

Rezultatul final care definește codul nostru este prezentat mai jos.



```

when green flag clicked
  set qpio 15 to input
  set qpio 13 to input
  set qpio 6 to input
  say Hello! Let's guard the windows together! for 4 secs
  say If any of the windows is opened, the alarm will go off for 4 secs
  repeat (3)
    if qpio 15 is high? then
      broadcast window1-open
    else
      broadcast window1-closed
    if qpio 13 is high? then
      broadcast window2-open
    else
      broadcast window2-closed
    if qpio 6 is high? then
      broadcast window3-open
    else
      broadcast window3-closed
    if qpio 15 is high? or qpio 13 is high? or qpio 6 is high? then
      set qpio 24 to output high
      say Alarm!
    else
      set qpio 24 to output low
      say 
    wait 1 secs
  set qpio 24 to output low
  
```

*Imag. 10. Codul principal pentru circuitul de probă  
Sursa: STEMKIT4Schools project*

### 1.2.3 Investigare

În sfârșit, putem rula codul nostru! Vă rugăm să urmați următoarele sarcini secundare pentru a afla mai multe despre acest exemplu. Pentru a evita citirile false, asigurați-vă că toate elementele utilizate pentru construirea circuitului (fire jumper sau rezistențe) nu acoperă niciunul dintre senzorii TCRT5000.

*Colectarea datelor*



Pentru prima rundă, asigurați-vă că senzorii nu sunt acoperiți sau, cu alte cuvinte, că nu există elemente care să reflecte fasciculul infraroșu emis și, prin urmare, să transforme pinii GPIO în stare înaltă. În cursul următor, încercați să acoperiți un senzor cu degetul sau orice alt element. Încercați să observați comportamentul atunci când toți cei trei senzori sunt acoperiți, simulând situația în care toate ferestrele sunt închise.

#### *Analiza datelor*

Pe baza datelor observate, puteți spune dacă distanța obstacolului față de senzor este în conformitate cu domeniul de lucru așteptat? Încercați să observați cât de aproape trebuie să fie obstacolul deasupra senzorului pentru a-l face să transforme pinul GPIO în stare înaltă. Comportamentul codului este conform cu așteptările? Sprite-urile de pe ecran se schimbă ca răspuns la senzori specifici care sunt acoperiți / descoperiți? Se declanșează alarma când este deschisă cel puțin o fereastră și pisica spune *Alarmă!*, în această situație?

#### *Prezentarea rezultatelor*

În această etapă suntem invitați să împărtășim rezultatele muncii noastre cu alte grupuri. A funcționat totul bine? Au existat dificultăți în configurarea întregului circuit? Au fost introduse modificări în cod? Dacă da, ce fel de? Au fost citirile cu privire la cât de aproape trebuie să fie obstacolul pentru a transforma portul GPIO al senzorului în stare înaltă consecvent pentru toți senzorii? A fost același caz și în alte grupuri?

### 1.2.4 Concluzii

Am reușit să creăm un circuit foarte simplu care poate servi drept demonstrație a unei instalații de securitate în clădire. În această etapă putem schimba idei cu alte grupuri, ce s-a făcut, în ce mod și în ce ordine și pentru a clarifica orice întrebări care ar putea apărea.

### 1.2.5 Follow-up - exercitiu (optional)

Exercițiul de urmărire poate include un multimetru, care măsoară tensiunea de la senzori în modul în timp ce obstacolul se apropie. Astfel vom putea spune la ce tensiune Raspberry Pi consideră că intrarea este în stare înaltă. De asemenea, putem încerca să schimbăm bucla principală utilizată în cod de la repetare */repeat to forever*.

## 1.3 Referente sau Resurse

Resursele pentru acest plan: <https://www.vishay.com/docs/83760/tcrt5000.pdf>