

# STEMKIT

## 4SCHOOLS

# REALIZAREA UNUI DETONATOR ÎN MINECRAFT

## PLANUL 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.	Realizarea unui detonator în Minecraft.....	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 .....	2
1.1.5	Durata .....	3
1.2	Planul de Lectie .....	4
1.2.1	Introducere.....	4
1.2.2	Pregatire .....	4
1.2.3	Executie .....	6
1.2.4	Concluzii .....	7
1.3	Referinte.....	8
1.4	Appendix .....	8



# 1. Realizarea unui detonator în Minecraft

## 1.1 Informatii generale

### 1.1.1 Scurta descriere

acest plan de lecție, vom construi un buton mare detonator roșu. Când apăsăm butonul va apărea o numărătoare inversă pe un afișaj pe 7 segmente, astfel încât jucătorul să aibă timp să fugă de explozie, iar apoi va apărea un imens crater în punctul în care jucătorul nostru stătea când am apăsat butonul. Acest lucru va fi destul de la îndemână pentru aventura noastră în Minecraft ca o modalitate rapidă de a goli spațiul pe măsură ce ne deplasăm și construim construcții. Acest plan de lecție introduce un alt tip de funcție GPIO, numită intrare, care detectează când este apăsat un buton.

### 1.1.2 Obiectivele invatarii

Principalele obiective de învățare ale acestui plan de lecție sunt:

- utilizarea unui afișaj pe 7 segmente pentru a interacționa cu Minecraft Pi.
- familiarizarea cu circuitele și programele pentru a interacționa cu pinii GPIO ai Raspberry Pi
- înțelegerea structurilor de bază ale limbajului de programare Python (și anume, utilizați *while loop*, *if statements*, *definition of function* - program sintaxa, scopul și funcția buclei *while*, instrucțiunile *if*, definiția funcției etc.

### 1.1.3 Links catre curriculum

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

- 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, circuite, calculatoare cu o singură placă

### 1.1.4 Materiale solicitate



Erasmus+

2019-1-FR01-KA201-062281



STEMKIT  
4SCHOOLS

Pentru acest plan de lecție (și pentru fiecare grup de studenți), pe lângă consola STEMKIT și Raspberry Pi, vom avea nevoie de:

- 1 x 40P T-Cobbler Plus GPIO Breakout Board
- 1 x cablu panglică curcubeu GPIO 40P
- Afișaj 1 x 7 segmente
- 11 x fire jumper de la bărbat la bărbat
- 1 x Breadboard
- 1 x buton
- Rezistor 1 x 10K Ohm
- Rezistențe de 8 x 220 Ohm

### 1.1.5 Durata

Durata acestui plan de lecție este estimată la aproximativ 80-100 de minute, adică două ore de clasă.

## 1.2 Planul Lecției

Planul lecției este împărțit în patru etape, care sunt introducerea, pregătirea, execuția și încheierea.

### 1.2.1 Introducere

În acest proiect, vom conecta un buton pe care îl putem apăsa pentru a declanșa un detonator. În scopuri realiste, alegem un buton roșu mare, astfel încât seamănă cu un buton detonator real pe care îl vedem în filme și jocuri video.

Butonul pe care urmează să îl folosim este un buton care nu se blochează, apasă pentru a face. Aceasta înseamnă că butonul nu face în mod normal un circuit electric și doar atunci când îl apăsăm se creează un circuit completat. Butonul nu se lipește și astfel, de îndată ce ne scoatem degetul, circuitul este rupt din nou.

Pentru a viziona un video tutorial despre cum să construim și să jucăm jocul detonator, vizitați următorul site web <https://www.wiley.com/WileyCDA/Section/id-823690.html> și alege Adventure 5 video.

### 1.2.2 Pregătire

Primul lucru pe care trebuie să-l facem este să realizăm un circuit și să conectăm senzorul la pini GPIO ai Raspberry Pi. Înainte de a continua, oprim Raspberry Pi și îl deconectăm. Pentru circuitul nostru, vom avea nevoie de afișajul nostru pe 7 segmente, o placă de rezoluție, rezistențe, fire jumper și un buton. În plus, folosim o placă de descompunere GPIO T-Cobbler 40P și un cablu cu panglică curcubeu 40P pentru a ne ajuta să configurăm circuitul pe panou în loc să conectăm firele jumper direct pe Raspberry. Circuitul complet este în Figura 1 care urmează.

Iată 4 pași pentru configurarea circuitului nostru:

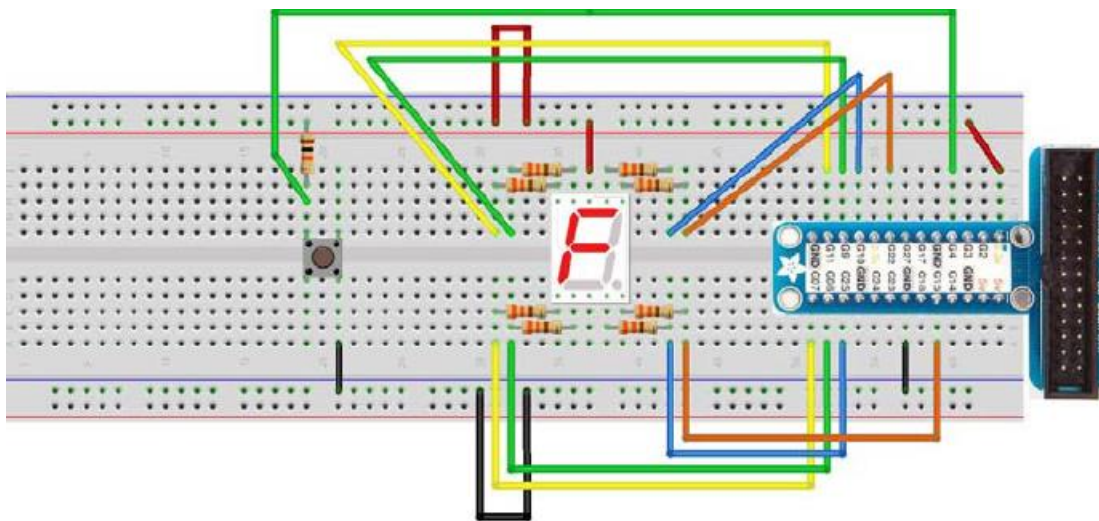
1. Apăsăm butonul într-un spațiu liber de pe panou. Butonul pe care l-am folosit are patru pini pe el și se potrivește frumos în mijlocul panoului. Apăsarea butonului unește știfturile din partea stângă împreună cu știfturile din partea dreaptă, atâta timp cât am instalat butonul cu știfturile afișate în partea de sus și de jos.
2. Avem nevoie de un rezistor pentru a trage butonul de intrare în sus la 3,3 volți. Dacă nu potrivim un rezistor, butonul nostru va genera apăsări false și detonatorul nostru va continua să se oprească tot timpul! Montați rezistența de tracțiune de 10K (cu benzile colorate: maro, negru, portocaliu), astfel încât un picior să fie conectat la pinul stâng al

butonului, iar celălalt picior să fie conectat la șina de alimentare de 3,3 volți din partea de sus a panoul de măsurare.

3. Rulăm un fir de la joncțiunea dintre piciorul rezistorului și butonul și conectăm celălalt capăt al firului la GPIO numărul 4.

4. Rulăm un fir de la pinul din dreapta jos al butonului și conectăm celălalt capăt al firului la șina de alimentare de 0 volți din partea de jos a panoului. Toate butoanele sunt în mod normal conectate la computer în acest fel.

Iată câteva informații suplimentare despre circuit: Când ne-am conectat butonul, am montat un rezistor pe acesta. Acest rezistor se numește rezistor de tracțiune, deoarece trage tensiunea pe pinul butonului „în sus” la 3,3 volți. Când butonul nu este apăsat (și nu se face nicio conexiune), firul care conectează GPIO la acest punct din circuit este acum la 3,3 volți, care este tensiunea pe care pinul GPIO o va interpreta ca un „1” digital. Când apăsăm butonul, mecanica internă a butonului face un circuit între pinii din stânga și din dreapta, iar firul GPIO este „tras puternic în jos” la 0 volți prin șina de alimentare de 0 volți. În această condiție, pinul GPIO va interpreta acest lucru ca un „0” digital. Rezistența are o valoare destul de ridicată (10K ohmi, adică 10.000 Ohmi), deoarece altfel butonul va scurta împreună șina de alimentare de 3,3 volți și șina de alimentare de 0 volți, iar Raspberry ar putea reporni! Acest lucru nu este recomandat, deoarece în unele cazuri poate provoca deteriorarea hardware-ului nostru. 10K este o formă de mână scurtă pe care inginerii o folosesc pentru a se referi la un rezistor de 10.000 Ohm.



**FIGURA 1 DIAGRAMA CIRCUITULUI UNUI BUTON LEGAT DE RASPBERRY PI**



### 1.2.3 Executare

Acum că avem cablul butonului nostru, suntem gata să scriem programul detonator în Python. Acest program va monitoriza starea butonului de fiecare dată în jurul buclei de joc și, atunci când detectează un „0” digital pe pinul GPIO (cu alte cuvinte, când butonul este apăsat), acesta va conta de la 5 la 0 în jos afișează pe 7 segmente și apoi aruncă o cutie masivă în lumea Minecraft.

Începem un nou program alegând File → New File /Fișier File Fișier nou și îl salvăm ca detonator.py.

Apoi începem să scriem programul importând modulele necesare:

```
import mcpi.minecraft as minecraft
import mcpi.block as block
import time
import anyio.seg7 as display
```

Apoi configurăm setarea GPIO pentru computerul nostru:

```
import RPi.GPIO as GPIO
BUTTON = 4
LED_PINS = [10, 22, 25, 8, 7, 9, 11, 15] #the order is important
```

Configurăm GPIO pentru buton astfel încât să fie ca intrare și configurăm GPIO-urile afișate:

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUTTON, GPIO.IN)
ON = False # False=common-anode, True=common-cathode
display.setup(GPIO, LED_PINS, ON)
```

Ne conectăm apoi la jocul Minecraft:

```
mc = minecraft.Minecraft.create()
```

Scriem apoi o funcție care aruncă un bloc de TNT pentru a marca unde va exploda bomba, numără în jos până la zero, apoi aruncă un crater mare unde a fost blocul TNT. Următorul cod construiește blocul TNT ușor în partea laterală a jucătorului, deci nu aterizează deasupra lui! Rețineți că craterul are o dimensiune de 20 de blocuri (10 la stânga și 10 la dreapta playerului dvs.) și această dimensiune este stabilită de calculele făcute în `setBlocks ()` aici:

```
def bomb(x, y, z):
    mc.setBlock(x+1, y, z+1, block.TNT.id)
    for t in range(6):
        display.write(str(5-t))
        time.sleep(1)
```

```
mc.postToChat("BANG!")
```

```
mc.setBlocks(x-10, y-5, z-10, x+10, y+10, z+10, block.AIR.id)
```

În cele din urmă, scriem bucla principală a jocului astfel încât să aștepte apăsarea unui buton, apoi declanșăm o bombă. Ar trebui să ne amintim că butonul nostru se conectează la 0 Volți atunci când este apăsat, motiv pentru care trebuie să comparăm GPIO cu `False` pentru a detecta o apăsare pe buton:

```
try:
    while True:
        time.sleep(0.1)
        if GPIO.input(BUTTON) == False:
            pos = mc.player.getTilePos()
            bomb(pos.x, pos.y, pos.z)
finally:
    GPIO.cleanup()
```

Ne salvăm programul și îl rulăm. Apoi, în lumea Minecraft, mergem undeva, apăsăm butonul și alergăm pentru viața noastră. Figura 2 prezintă consecințele exploziei.



**FIGURA 2 UN CRATER ÎN LUMEA MINECRAFT**

Dimensiunea craterului este stabilită de numerele din instrucțiunea `setBlocks()`. Modificând cifrele din această afirmație, putem extinde sau micșora dimensiunea exploziei. Încercați să faceți acest lucru și vedeți ce se întâmplă!

## 1.2.4 Concluzii



Am programat cu succes un buton pentru a simula o explozie în lumea noastră Minecraft. De asemenea, am programat un afișaj pentru a arăta numărătoarea inversă până când bomba merge.

De ce să nu faci chiar mai mult?

Adăugați încă trei butoane în circuitul dvs., în același mod în care ați adăugat primul buton. Apoi mai scrieți câteva coduri Python pentru a face ca aceste butoane să facă lucruri diferite, cum ar fi să postați mesaje aleatorii în chatul Minecraft, să vă teleportați playerul într-o locație secretă, să construiți blocuri de diferite tipuri în care stă jucătorul dvs. sau chiar să construiți o casă întreagă în care sunt în picioare. Următoarele numere GPIO sunt sigure de utilizat pentru cele trei butoane suplimentare.

Button	Raspberry Pi Pin
BUTTON2	14
BUTTON3	23
BUTTON4	24

## 1.3 Referinte

O'Hanlon M. & Whale D., 2015, Adventures in Minecraft, Wiley Publications.

## 1.4 Appendix

Quick Reference Table

Configuring GPIO pins	Reading and writing GPIO pins
<pre>import RPi.GPIO as GPIO # RaspberryPi import anyio.GPIO as GPIO # Arduino GPIO.setmode(GPIO.BCM) GPIO.setup(5, GPIO.OUT) # output GPIO.setup(5, GPIO.IN) # input</pre>	<pre>GPIO.output(5, True) # set on (3.3V) GPIO.output(5, False) # set off (0V) if GPIO.input(6) == False:     print("pressed")</pre>
Safely cleaning up after using the GPIO	Using the 7-segment display module
<pre>try:     do_something() # put code here finally:     GPIO.cleanup()</pre>	<pre>import anyio.seg7 as display LED_PINS = [10,22,25,8,7,9,11,15] ON = False # common-Anode ON = True # common-Cathode display.setup(GPIO, LED_PINS, ON)</pre>
Writing character to the 7-segment display	Other 7-segment display function
<pre>display.write("3") display.write("A") display.write("up")</pre>	<pre>display.setdp(True) # point on display.setdp(False) # point off display.clear() display.pattern([1,1,1,1,1,1,1,1])</pre>