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

'We are the makers – Scenariu didactic IoT: Comunicarea prin fibră optică folosind Adafruit CPX Express

Autor: Thomas Jörg, Johannes-Kepler-Gymnasium Weil der Stadt

Această activitate a fost dezvoltată și testată în clasă cu circa 16 elevi cu vârste între 13 și 14 ani, în anul școlar 2019-2020. Ea reflectă experiența avută cu elevi talentați care au creat singuri scripturile pornind de la o unitate de predare privind tehnologiile de comunicare. Această lucrare se dorește o recomandare, un punct de plecare.

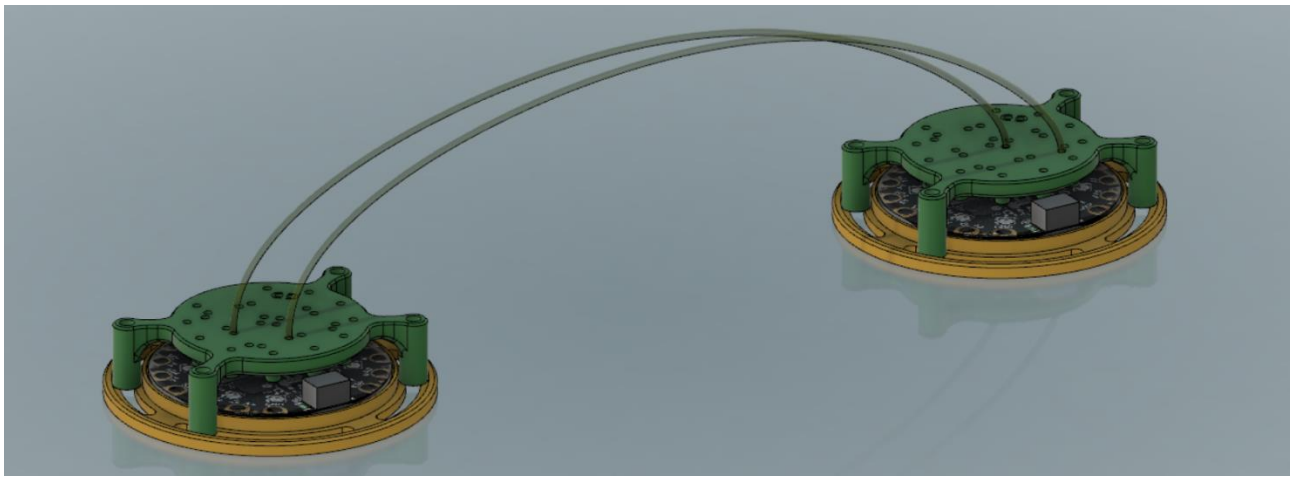
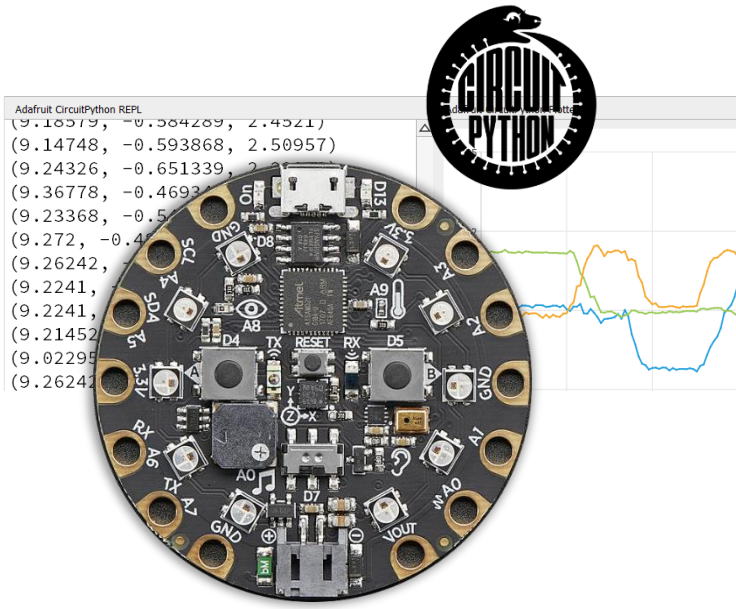


Figura 1: Prototipul unui dispozitiv de comunicare prin fibră optică dintre două microcontrolere

IoT înseamnă crearea de rețele de dispozitive, controlate de calculator. Comunicarea în rețea înseamnă adesea că dispozitivele schimbă informații între ele cu sau fără fir. Fibra optică este un mediu foarte des utilizat. Avantajul didactic al utilizării fibrei optice pentru introducerea în tehnologia de comunicare este vizibilitatea schimbului de informații: se pot vedea impulsurile luminoase cu care sunt transmise biții de informație. Înțelegerea modului de comunicare prin fibră optică necesită înțelegerea unui principiu fizic simplu, și anume reflexia totală.



Număr proiect: 2017-1-DE03-KA201-035615

1. Titlu	**Dispozitive IoT de comunicare în rețea: Programarea fibrei optice**
2. Grup țintă	▪ 14 - 16 ani
3. Durată	▪ Minim 7 săptămâni cu câte 2-3 lecții e săptămână
4. Nevoile de învățare	▪ Senzorii și actuatoare în schimbul de informații dintre dispozitivele digitale ▪ Principiul comunicării în rețea bazată pe protocoale și pachete ▪ Aplicarea conceptului de reflexie totală pentru transportul pulsurilor de lumină ▪ Programarea în Python a microcontrolerelor, în grupuri mici sau perechi de elevi
5. Rezultatele învățării	▪ Cum funcționează un sistem IoT? ▪ Cum se structurează și implementează comunicarea în rețea? ▪ De ce e nevoie de un protocol de comunicații? ▪ Cum funcționează comunicarea digitală bazată pe pachete?
6. Metodologie	▪ În această activitate elevii vor construi și programa, de la zero, o comunicare serială între două dispozitive microcontrolere. Elevii vor folosi Serial Monitor și Serial plotter pentru a vizualiza și reprezenta grafic datele.
7. Aranjamente	▪ Cabluri de fibră optică pentru a conecta microcontrolerele. ▪ Microcontrolere Adafruit CP, ▪ Aceste controlere CPX sunt programate cu Circuit python ▪ Fiecare elev primește un laptop cu un editor Mu instalat pentru microcontrolerul CPX ▪ Fiecare CPX e introdus într-o carcasă tipărită 3D care asigură o aliniere precisă a fibrelor de sticlă ▪ Fiecare elev va scrie un jurnal al activităților lor din cadrul proiectului

8. Unelte/ Materiale/ Resurse

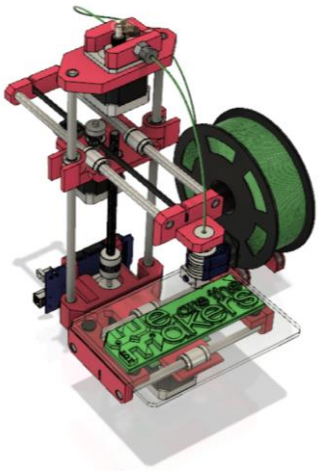


Figura 2: Modelul unei imprimante 3D

Imprimante 3D

Sunt necesare circa 2-3 imprimante 3D întrucât elevii își vor tipări singuri carcasele pentru CPX. Desigur, elevii ar putea să își proiecteze singuri componentele.

Componente tipărite 3d:

Ca punct de plecare, toate părțile necesare sunt date în format .stl și ca fișiere Autodesk Fusion 360.

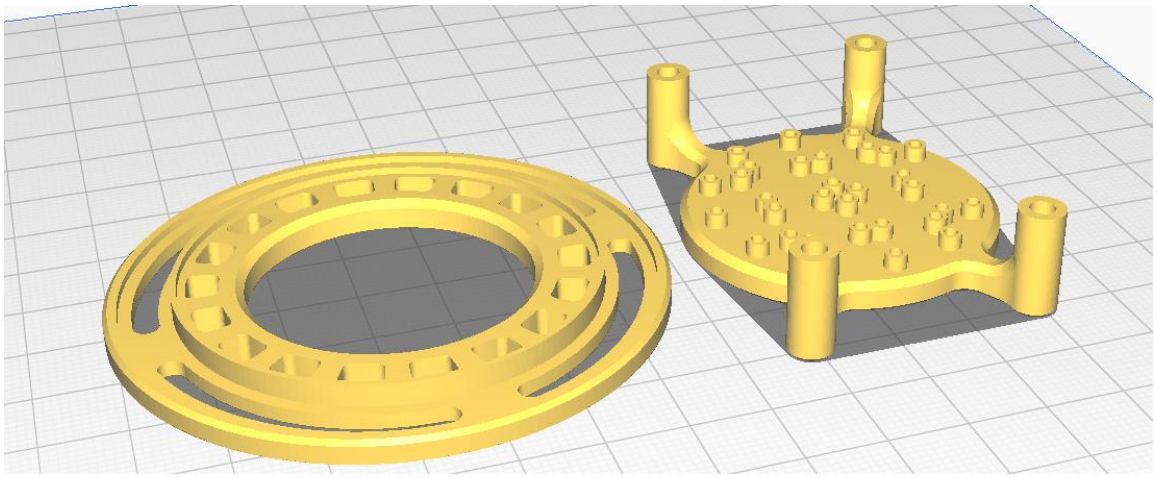
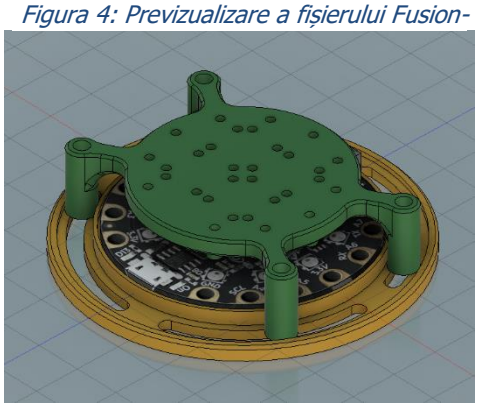


Figura 3: Fișiere STL – timpul de tipărire pentru ambele componente – circa 1h

Cele două părți sunt dimensionate astfel încât un CPX poate fi inserat în partea inferioară, cu un spațiu lateral de circa 0.5mm. Părțile superioară și inferioară sunt conectate cu șuruburi M3: Lungimea recomandată a șurubului este de 25mm. Piesele pot fi blocate cu piulițe-fluture, astfel încât microcontrolerul să poată fi rapid introdus și scos.

Figura 5: Șurubuiri M3 și piulițe-fluture

Figura 4: Previzualizare a fișierului Fusion-



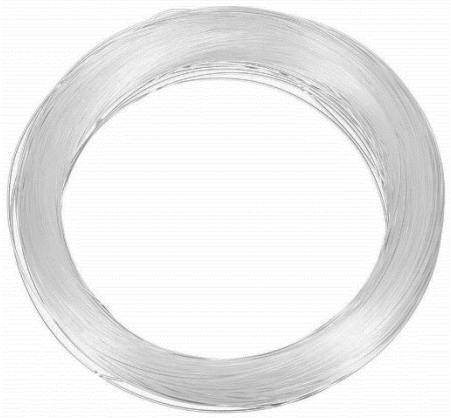


Figura 6: 100 metri de fibră de sticlă

Carcasa este proiectată să conecteze cu ușurință cablurile de fibră optică și să le poziționeze optim deasupra LED-urilor și a senzorului de luminozitate. Ca fibră optică este utilizat un cablu PMMA de 1,5 mm. Aceste fibre se găsesc de obicei în role de aproximativ 100 de metri în lungime, la un preț de aproximativ 20 de euro. Fiecare grup de elevi care dorește să conecteze două CPX-uri are nevoie de două bucăți de fibră, de 1 metru fiecare: una ca linie de transmisie și una ca linie de primire. Cu o astfel de rolă de fibră, e de ajuns pentru a implementa lecția la mai multe clase.

Diametrul de 1.5mm asigură faptul că elevii nu se vor răni cu ușurință. În plus, fibrele de o asemenea grosime sunt rezistente la îndoire și deformare.

Partea superioară a carcasei imprimate 3D are multe găuri, care au rol de ghidare a acestor cabluri. Găurile au un diametru de 1.8mm.

Figura 7: Hand drill

În cazul în care firul nu intră în găuri, acestea pot fi lărgite cu un burghiu mic, de mână. Aceste burghie manuale sunt ghidate manual și, prin urmare, inofensive.

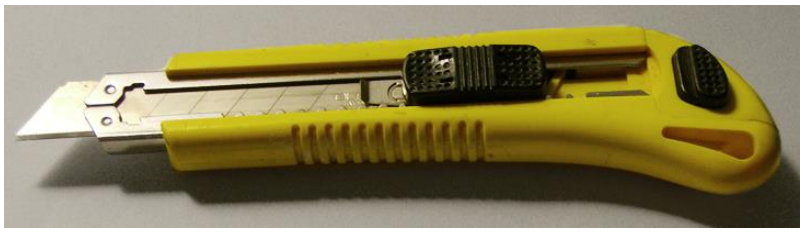


Figura 8: Cutter

Pentru a tăia fibrele de sticlă, nu ar trebui să utilizați clești sau foarfece, deoarece strânge fibra de sticla. Este necesară o tăietură netedă din care lumina poate ieși. Prin urmare, se recomandă utilizarea unui cutter.

Adafruit CPX

În cadrul acestei activități folosim microcontrolerul bazat pe Python "Circuit Python Express" produs de compania "Adafruit Industries". Această placă e convenabilă deoarece include și o serie de senzori și actuatoare.

CPX s-a dovedit a fi robust și fiabil în mai multe aplicații de predare. În plus, Adafruit oferă, de asemenea, o mulțime de informații despre microcontroler. În plus față de numeroase programe exemplu, există, de asemenea, o documentație bună.

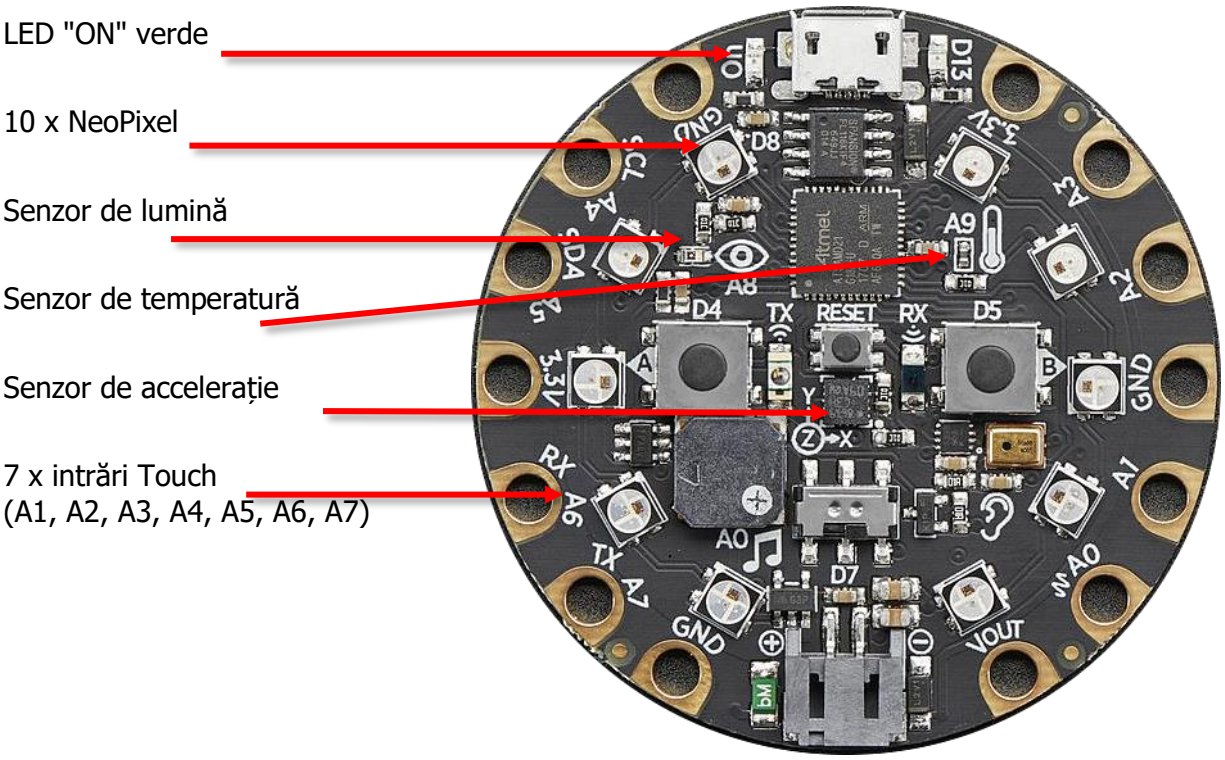


Figura 9: Câteva caracteristic ale plăcii CPX

La următoarea adresă se găsesc informații utile pentru lucrul cu CPPX și o serie de exemple de programe: https://iludis.de/?page_id=291

Echipamente necesare

Calculatoarele cu care lucrează elevii ar trebui să aibă instalat următoarele soft-uri:

- Autodesk Fusion 360 (sau oricare software de modelare 3D, ex. Wings3D)
- CURA ,
- Conexiune a internet pentru descărcarea bibliotecilor necesare
- **Editorul Mu (https://codewith.mu/)**

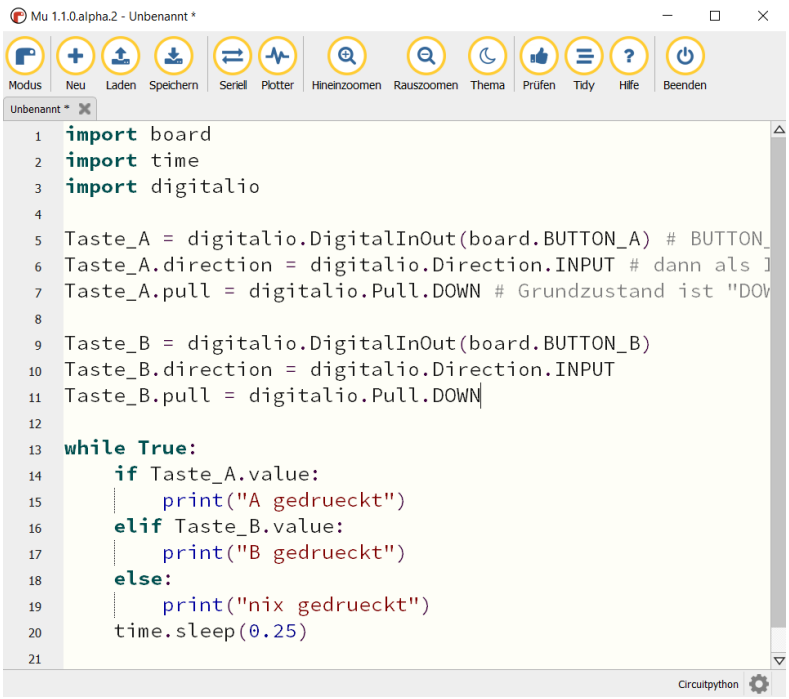


Figura 10: Captură după editorul Mu

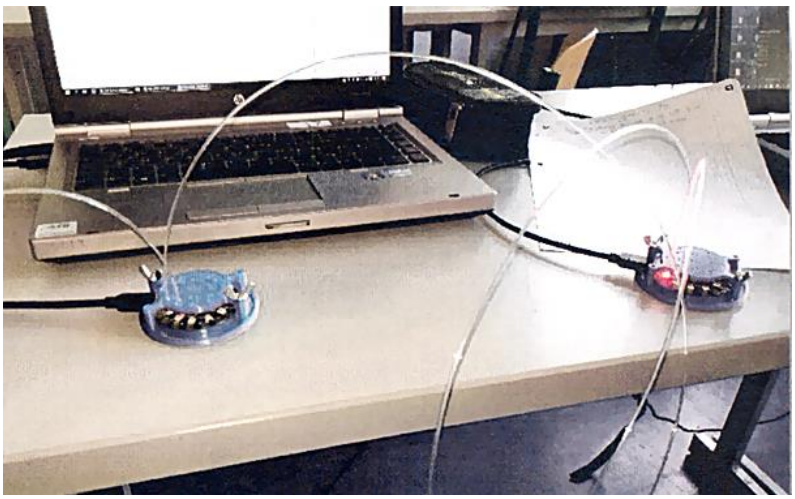


Figura 11: Fotografie a unui aranjamet realizat de elevi

9. Descrierea pa cu pas c activității/conținutului

Lecțiile 1 & 2 (90min): Introducere în IoT

Elevilor li se va prezenta conceptul de IoT prin exemple: roboți de aspirare ce pot fi controlați de la distanță prin intermediul unei aplicații, stații meteo bazate pe web, agricultura inteligentă și la sfârșit aplicațiile de sănătate. Elevii trebuie să examineze modul în care funcționează aceste dispozitive și ce componente sunt necesare pentru crearea acestora: un sistem bazat pe microcontroler controlează și coordonează o serie de senzori și actuatoare. În plus comunică și coordonează cu alte sisteme asemănătoare, de obicei prin rețele de comunicare wireless. Componente necesare: senzori, actuatoare, dispozitive de comunicare. Se va discuta despre posibilități, amenințări și limitări: unde ar trebui și unde nu ar trebui folosit IoT.

Lecțiile 2 & 3 (90min): Introducere în teoria protocoalelor de comunicație

Elevii vor juca un joc de rol: clasa privește și dezvoltă idei. Se aplic următoarele reguli:

- Doi elevi, "transmițător" și "receptor", stau împreună pe scaune, poziționate spate în spate, astfel încât să nu se poată vedea reciproc, iar între cele două scaune, se plasează un scaun gol.
- Transmițătorul ar trebui să trimită receptorului două cuvinte complet diferite, care sunt rostite invers cu un înțeles complet diferit: 4 litere: LIVE/EVIL și STAR/ RATS | 3 litere: GOD/ DOG și RAW/WAR
- La fel ca limba vorbită, transmiterea de cuvinte permite o singură secvență de caractere individuale în succesiune, astfel încât toate literele trebuie transmise individual. Prin urmare, literele celor două cuvinte sunt scrise pe foi individuale.
- Aceste foi pot fi transmise unul câte unul de la transmițător la receptor, prin plasarea foii de către transmițător pe scaunul liber și ridicarea foii de către receptor din acel loc – fără ca cei doi să se vadă unul pe celălalt.
- Singura comunicare directă permisă între cei doi este un singur ton (ex. un „beep”) care e permis tuturor.
- Dacă o comunicare eșuează, transmisiunea e întreruptă, se stabilește o nouă regulă și elevii o iau de la capăt.

Este logic ca elevii să realizeze că transmiterea datelor trebuie făcută prin intermediul unor norme convenite de comun acord, și anume un protocol:

Elevii trebuie să se pună de acord asupra unui semnal de pornire. Trebuie să existe o pauză în timp după fiecare literă pentru ca literele să ajungă în ordinea corectă; acest lucru poate fi convenit prin cronometrare sau prin "bipuri". În plus, trebuie să se convină ordinea în care sunt schimbate scrisorile. În caz contrar, cuvintele vor avea un sens diferit, neintenționat. Și în cele din urmă, comunicarea trebuie oprită.

Lecțiile 4 & 5 & 6 (120min): Principiile comunicației seriale

Principiile multiplexării și demultiplexării: biții de date sunt transmiși unul câte unul, începând de la MSB ("cel mai semnificativ bit") la LSB ("cel mai puțin semnificativ bit") într-o ordine specificată, printr-un singur cablu de date.

Se face o distincție între transmiterea sincronă și cea asincronă a datelor: În cazul sincron, mai simplu, receptorul – care funcționează ca un master – specifică frecvența comună a ceasului cu care declanșează transmisia individuală de biți la emițător (slave). După un număr prestabilit de biți transmiși, transferul de date se termină.

În cazul asincron, trebuie să se convină, în prealabil, asupra unui ciclu de timp pentru ambele dispozitive implicate în comunicare, deși acești timpi pot diferi doar cu foarte puțin unii de alții.

Figura 12: Emile Baudot

Lecțiile 7 (45min):

Istoria transmisiei de date: Émile Baudot

Folosind codul baudot pe 5 biți, elementele de bază ale transmiterii datelor sunt prezentate folosind un exemplu practic. Dacă doriți să transferați numai litere mari, aveți nevoie de 26 de caractere diferite și, eventual, de 2-3 semne de punctuație, cum ar fi spațiul sau semnul de întrebare. Pentru a codifica aceste simboluri diferite cu biți (sunt mai puțin de 32 de caractere distincte), e nevoie de 5 biți. O posibilă codificare ar fi, de exemplu:

Symbol	A	B	C	D	E	F	G	H	I	J	K
Bitcode	00001	00010	00011	00101	00110	00111	01000	01001	01010	01011	01100

Symbol	L	M	N	O	P	Q	R	S	T	U	V
Bitcode	01101	01110	01111	10000	10001	10010	10011	10100	10101	10110	10111

Symbol	W	X	Y	Z	LEER	START	STOP	.	
Bitcode	11000	11001	11010	11011	11100	11101	11110	11111	00000

Elevii ar trebui să propună un sistem pentru a codifica litere cu biți – ideal ar fi ca ei să propus ceva asemănător cu cel din tabelul de mai sus.

Biții sunt transmiși cu o anumită viteză. Aceasta se numește rată baud, numită după inginerul francez Emile Baudot. La acest moment în lecție se poate discuta despre biografia lui Baudot.

Lecțiile 8 & 9 (90min): Repetare Reflexia totală

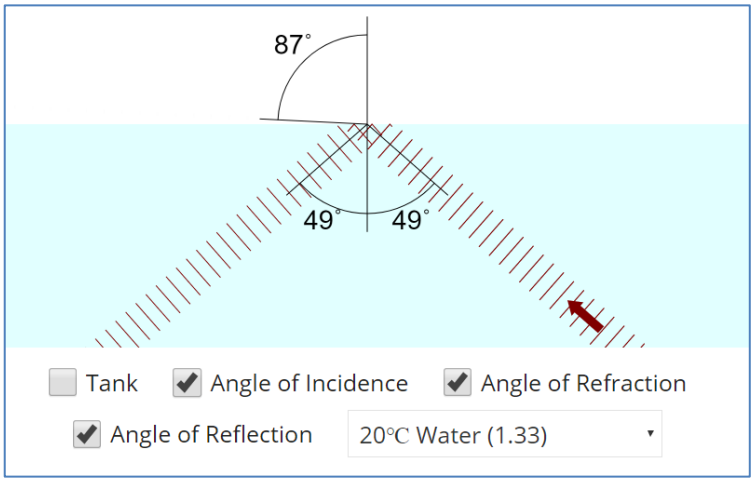
Refexia totală

Se folosesc simulările „Refracția luminii": https://javalab.org/en/light_refraction_en/
Și „Reflexia internă totală": https://javalab.org/en/total_internal_reflection_en/

Pentru trecerea de la apă la aer, "Fasciculul luminos este frânt în timpul tranziției de la mediul optic mai dens (apă) la mediul optic mai puțin dens (aer)"

În cazul în care unghiul de incidență al fasciculului de lumină în apă depășește un așa-numit "**unghi critic**", atunci unghiul de refracție în aerul mediu ar trebui să fie mai mare de 90° – și acest lucru este imposibil! Prin urmare, raza de lumină nu are de ales decât să rămână în apă.

1) unghiuri critice la trecerea din diferite medii în aer
https://javalab.org/en/light_refraction_en/



Determinați unghiurile critice – exe. unghiurile de incursiune la care fasciculul de lumină NU este scos din mediu:

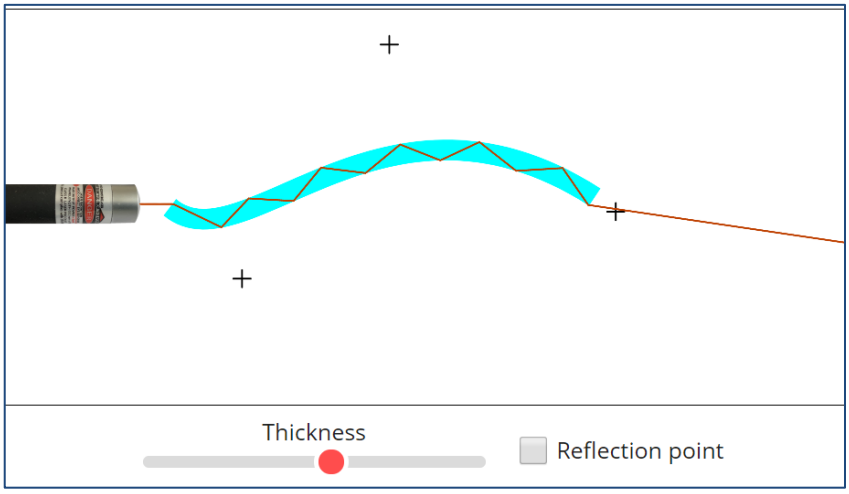
Medium	Unghiul critic, deasupra este reflexia totală
Apă	49°
Diamant	
Safir	
Sticlă	

2) Interpretați următoarele două imagini:

Interpretați zonele selectate din cele două imagini cu propriile cuvinte. apar aceste reflecții? Și ce vedeți mai exact în Zona 3?

3) Aplicații ale reflexiei totale

https://javalab.org/en/total_internal_reflection_en/



Pe partea stângă este un laser care alimentează fasciculul său de lumină într-o așa-numită fibră de sticlă. Fibra optică este utilizată pentru a transporta informații cu lumină. Pe de o parte a fibrei optice, semnalelor de la calculator sunt iradiat cu ajutorul fasciculului laser și, pe de altă parte, cu ajutorul unui senzor de lumină semnalele luminoase sunt primite și transmise la computerul receptor. Explicați principiul de funcționare al fibrei optice, răspunzând la următoarele întrebări:

1. De ce raza de lumină rămâne în fibră?
2. Există situații în care fasciculul de lumină se scurge din fibră prea devreme?
3. Cum depinde raza de lumină din fibră de grosimea fibrei?

Lecțiile 10 & 11 (90min), Introducere în programarea în Python:

CPX e conectat la calculator prin cablu USB și programat cu mediul "mu-Editor". În continuare sunt prezentate câteva scripturi cu care puteți începe lucrul cu CPX: https://iludis.de/?page_id=291

Script 1, "Blink", LED e pornit și oprit:

```
import board
import digitalio
import time

meinPin = digitalio.DigitalInOut(board.D13)
meinPin.direction = digitalio.Direction.OUTPUT

while True:
      meinPin.value = True
      time.sleep(1)
      meinPin.value = False
      time.sleep(1)
```

Script 2, "beep", e emis un ton:

```
import time
from adafruit_circuitplayground.express import cpx

for i in range (1, 5, 1):
    cpx.start_tone(262)
    time.sleep(0.2)
    cpx.stop_tone()
    time.sleep(0.2)
    print(i)
```

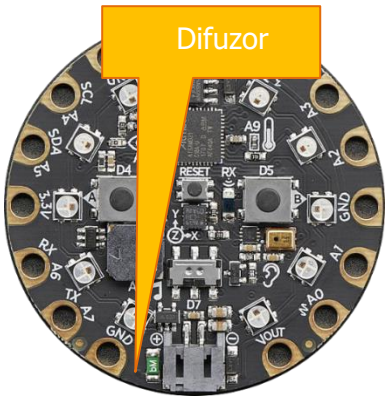


Figura 13: Localizarea difuzorului

Script 3, "Touch", se emite un sunet la apăsare

```
import board
import time
import touchio
from adafruit_circuitplayground.express import cpx

B_A1 = touchio.TouchIn(board.A1)
B_A2 = touchio.TouchIn(board.A2)

while True:
    if B_A1.value == True:
        cpx.start_tone(262)
    else:
        cpx.stop_tone()
    if B_A2.value == True:
        cpx.red_led = True
    else:
        cpx.red_led = False

    print(B_A1.value, B_A2.value)
    time.sleep(0.1)
```

Script 4, Comutare Neopixel

```
import board, time, neopixel
NeopixelListe = neopixel.NeoPixel(board.NEOPIXEL,
10)

for i in range (10):
    NeopixelListe[i] = (0,64,0)
    print(NeopixelListe[i])
    time.sleep(0.1)
print(NeopixelListe)
```

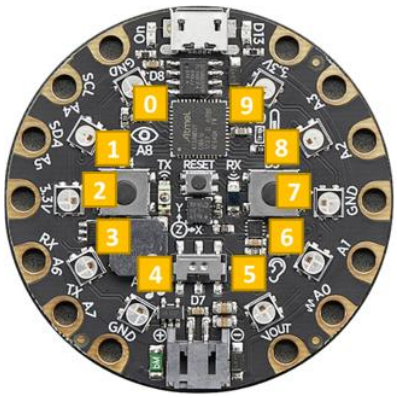


Figura 14: Numerotarea neopixel-ilor

Script 5, citire de la senzorul de lumină

```
import board
import time
import analogio

licht = analogio.AnalogIn(board.LIGHT)

while True:
    print((licht.value,))
    time.sleep(0.1)
```

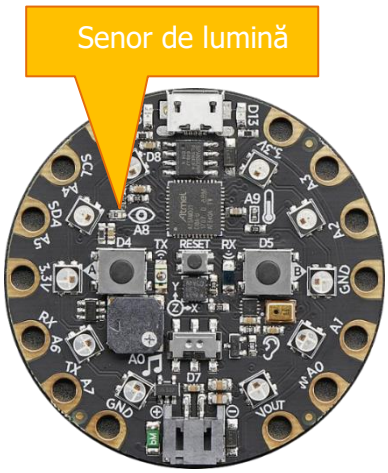


Figura 15: Localizarea senzorului de lumină

Lecțiile 12 & 13 (90 min): Exemple de comunicare sincronă

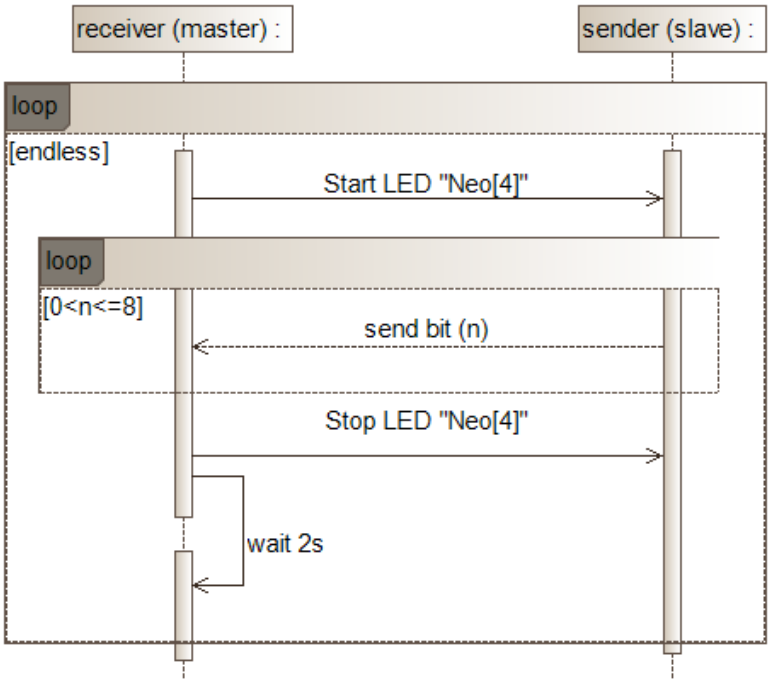


Figura 16: Diagrama de secvență UML a comunicației sincrone

Cod sursă receptor	Cod sursă transmițător

`import board`
`import time`
`import neopixel`
`import analogio`
 	
`light = analogio.AnalogIn(board.LIGHT)`
`Neo=neopixel.NeoPixel(board.NEOPIXEL,10)`
`pulseDuration = 0.05`
 	
`while True:`
`  listenBits = [2, 2, 2, 2, 2, 2, 2, 2]`
`  Neo[4] = (0, 255, 0)`
`  for i in range(len(listenBits)):`
`  time.sleep(pulseDuration)`
`  if light.value < 40000:`
`  listenBits [i] = 0`
`  if light.value > 40000:`
`  listenBits [i] = 1`
`  Neo[4] = (0, 0, 0)`
`  print(listenBits)`
`  time.sleep(2)`
	`import board`
	`import time`
	`import neopixel`
	`import analogio`
	
	`light = analogio.AnalogIn(board.LIGHT)`
	`Neo=neopixel.NeoPixel(board.NEOPIXEL,10)`
	`pulseDuration = 0.05`
	
	`while True:`
	`  sendBits = [1, 0, 0, 1, 1, 0, 1, 0]`
	`  if light.value > 40000:`
	`  for i in range(len(sendBits)):`
	`  if sendBits[i] == 1:`
	`  Neo[6] = (255, 0, 0)`
	`  if sendBits[i] == 0:`
	`  Neo[6] = (0, 0, 0)`
	`  time.sleep(pulseDuration)`

Lecțiile 14 & 15 (90 min): Exemple de comunicații asincrone

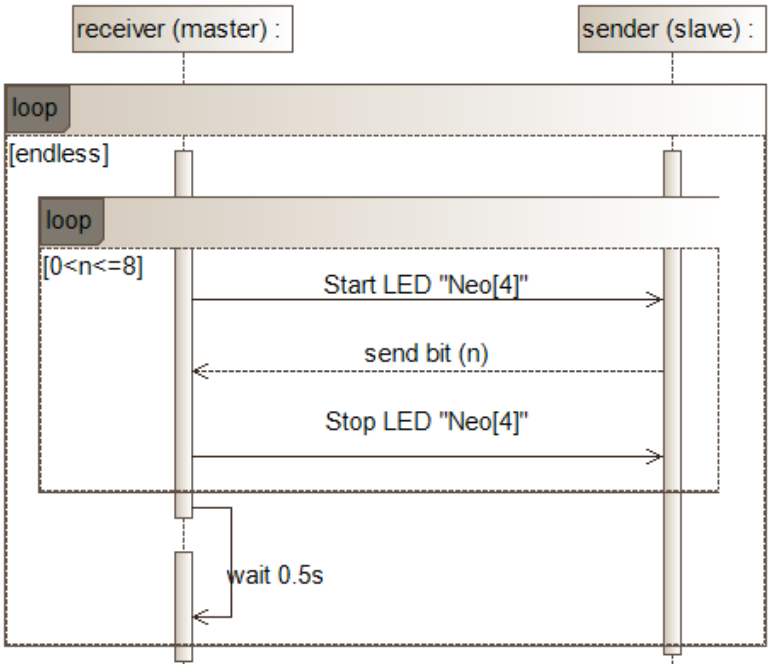


Figura 17: Diagrama de secvență UML a comunicației asincrone

Cod sursă receptor	Cod sursă transmițător

`import board`
`import time`
`import neopixel`
`import analogio`
 	
`light = analogio.AnalogIn(board.LIGHT)`
`Neo=neopixel.NeoPixel(board.NEOPIXEL, 10)`
`pulseDuration = 0.02`
 	
`while True:`
`  listenBits = [2, 2, 2, 2, 2, 2, 2, 2]`
`  for i in range(len(listenBits)):`
`  Neo[4] = (0, 255, 0)`
`  time.sleep(pulseDuration)`
`  if light.value < 40000:`
`  listenBits[i] = 0`
`  if light.value > 40000:`
`  listenBits[i] = 1`
`  time.sleep(pulseDuration)`
`  Neo[4] = (0, 0, 0)`
`  time.sleep(pulseDuration)`
`  print(listenBits)`
`  time.sleep(0.5)`
	`import board`
	
	`import neopixel`
	`import analogio`
	
	`light = analogio.AnalogIn(board.LIGHT)`
	`Neo=neopixel.NeoPixel(board.NEOPIXEL, 10)`
	`sendBits = [1, 0, 0, 1, 1, 0, 1, 0]`
	`i = 0`
	`transmit = True`
	
	`while True:`
	`  print(i)`
	`  if light.value > 40000 and transmit:`
	`  if sendBits[i] == 1:`
	`  Neo[6] = (255, 0, 0)`
	`  if sendBits[i] == 0:`
	`  Neo[6] = (0, 0, 0)`
	`  transmit = False`
	`  if light.value < 40000 and not transmit:`
	`  i = i+1`
	`  i = I % 8`
	`  transmit = True`

10. Feedback	La sfârșitul lecțiilor, elevii ar trebui să dețină cunoștințe bine fundamentate cu privire la modul în care funcționează comunicația serială și ce principii sunt necesare pentru implementarea tehnologiei. Pe parcursul lecțiilor se discută aspecte importante de electronică, optică și construirea de protocoale.
11. Evaluare	Elevii păstrează un jurnal al activităților derulate care poate fi verificat de profesor. Elevii ar putea să prezinte rezultatele experimentelor. În plus, la sfârșit elevi vor fi evaluați cu un test în clasă standard.