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

‘We are the makers - IoT’ Learning Scenario: comunicazione in fibra di vetro con Adafruit CPX Express

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

Il seguente lavoro è stato sviluppato e testato in un ambiente scolastico con circa 16 studenti di 13-14 anni nell'anno scolastico 2019/2020. Esso riflette l'esperienza di studenti talentuosi e curiosi che hanno realizzato da soli i loro script di programmazione dopo un'unità didattica di tecnologia di rete. Questo documento dovrebbe essere una raccomandazione come punto di partenza.

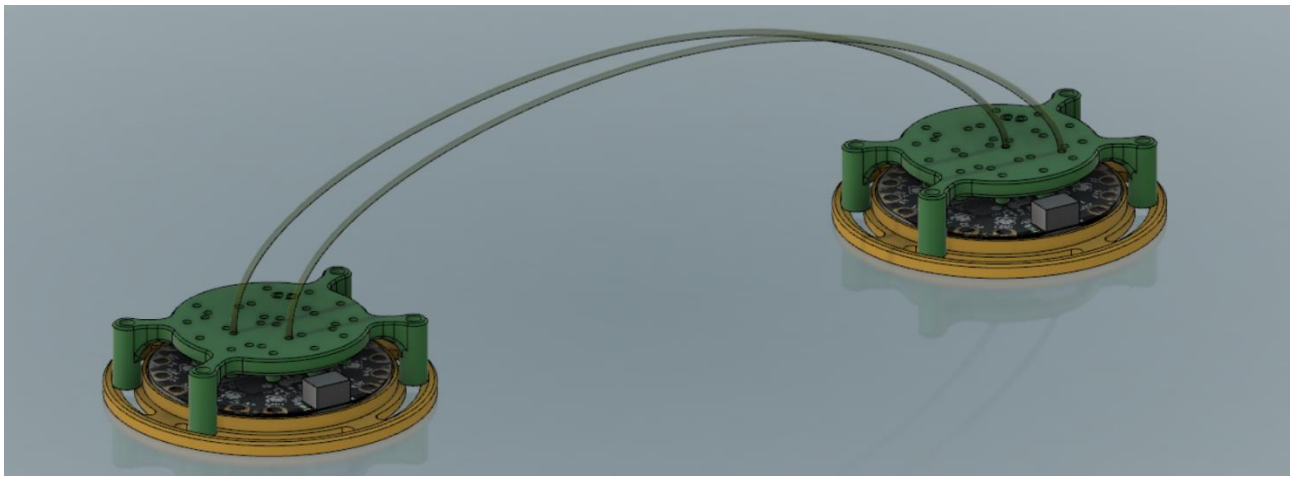
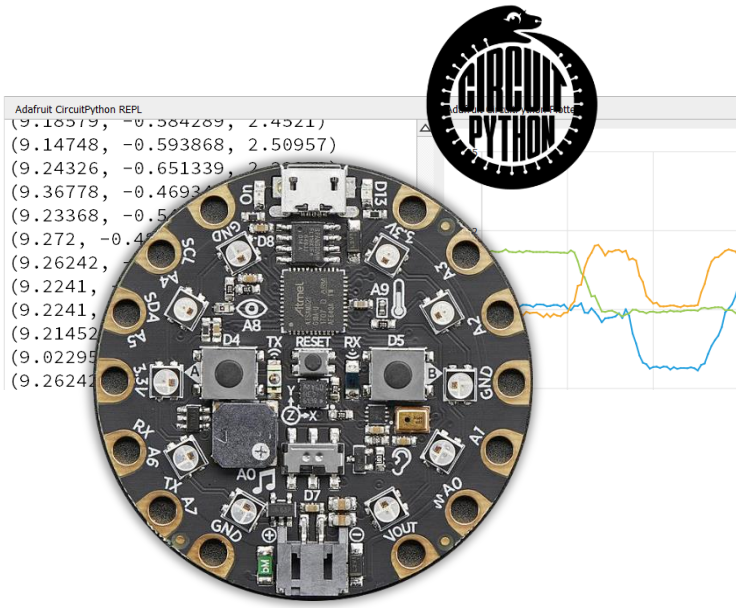


Figure 1: *Prototipo di una configurazione di comunicazione in fibra di vetro di due microcontrollori*



IoT significa collegamento in rete di dispositivi controllati da computer. La comunicazione in rete significa spesso che gli apparecchi si scambiano le informazioni senza fili, cioè via radio. In effetti, sono inclusi anche i classici mezzi di trasmissione come il collegamento in fibra ottica.
Il vantaggio didattico dell'utilizzo della fibra ottica per l'introduzione alla tecnologia della comunicazione è la visibilità dello scambio di informazioni: si possono vedere gli impulsi di luce con cui vengono scambiati i bit di informazione. Un semplice principio fisico, ovvero la riflessione totale, è sufficiente per comprendere la moderna comunicazione in fibra ottica.

1. Titolo dello scenario	**Comunicazione di rete dei dispositivi IoT: Programmazione della fibra di vetro**
2. Gruppo target	▪ 14 – 16 anni
3. Durata	▪ Minimo 7 settimane con 2-3 lezioni a settimana
4. Elementi di apprendimento discussi durante la lezione	▪ Sensori e attuatori nello scambio di informazioni tra dispositivi digitali ▪ Principio della comunicazione di rete basata su protocollo e su pacchetto ▪ Applicazione della riflessione totale per il trasporto di impulsi di luce. ▪ Programmazione di microcontrollori a base di pitone in piccoli gruppi di due studenti ciascuno
5. Risultati attesi dell'apprendimento	▪ Come funziona un sistema IoT? ▪ Come strutturare e implementare la comunicazione di rete? ▪ Perché avete bisogno di un protocollo di comunicazione? ▪ Come funziona la comunicazione digitale a pacchetti?
6. Metodologie	▪ In questo scenario gli studenti costruiranno, costruiranno e programmeranno da soli una comunicazione seriale tra due dispositivi a microcontrollore da zero. Gli studenti useranno anche il Serial Monitor e il Serial plotter per visualizzare e tracciare i dati.
7. Ambiente	▪ Un set di cavi in fibra ottica di classe per collegare i microcontrollori ▪ Un set di classe di microcontrollori CPX Adafruit, ▪ Questi controllori CPX sono programmati con Circuit python ▪ Ogni studente riceve un portatile con un editor Mu preinstallato per il suo microcontrollore CPX ▪ Ogni CPX è costruito in un involucro stampato in 3D, che garantisce l'allineamento preciso delle fibre di vetro. ▪ Ogni studente scrive un registro del suo lavoro di progetto

8. Strumenti, materiali e risorse

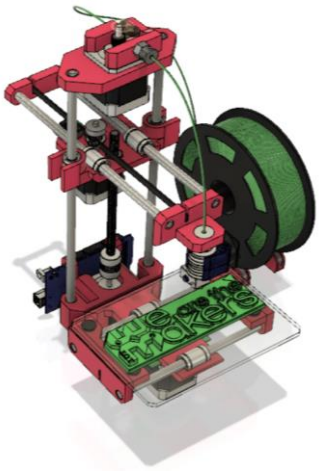


Figure 2: Un modello di stampante 3D

3D-Drucker
Sono necessarie circa 2-3 stampanti 3D, poiché gli studenti stamperanno CPX-housings. Naturalmente, è possibile per gli studenti costruire parti di macchine da soli

Componenti stampati 3d:

Come punto di partenza, tutte le parti necessarie sono fornite in .stl-formato e come file Autodesk Fusion 360:

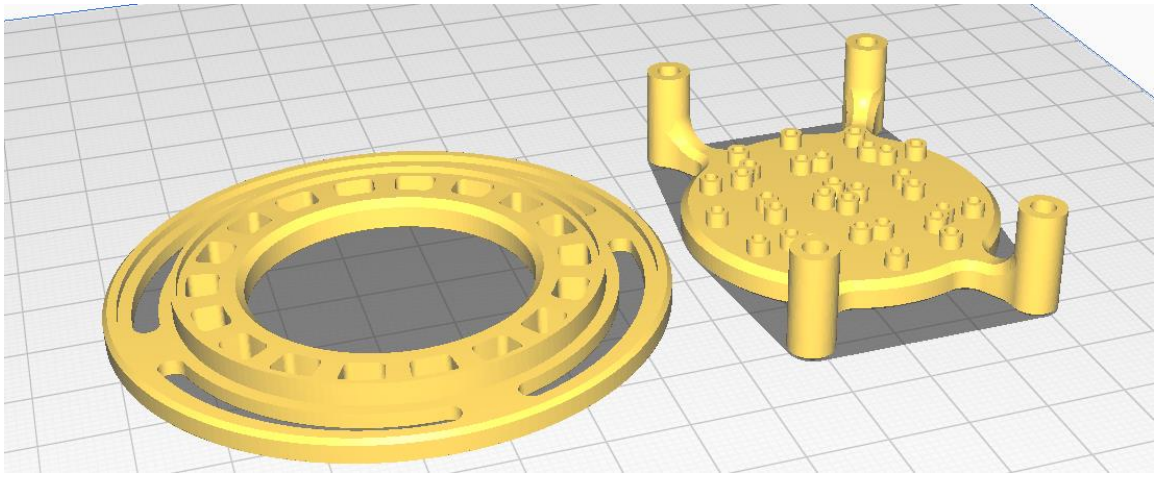
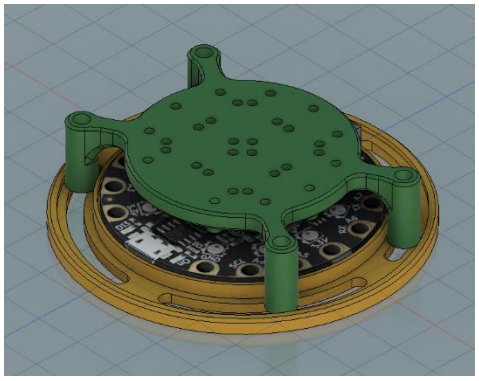


Figura 3: STL-File.Tempo di stampa per entrambe le parti circa 1 ora

Le due parti sono dimensionate in modo tale che un CPX con circa 0,5 mm di gioco possa essere inserito nella parte inferiore. La parte superiore e quella inferiore sono collegate con viti M3: La lunghezza consigliata della vite è di 25mm. Le parti possono essere bloccate con dadi ad alette, in modo che il microcontrollore possa essere rapidamente rimosso e rimosso.

Figura 5: Vite M3 e dado ad alette

Figura 4: Anteprima del file di Fusion



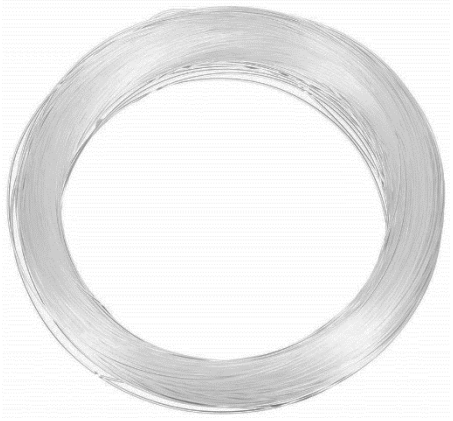


Figura 6: 100 metri di fibra di vetro

L'involucro è progettato per collegare facilmente i cavi in fibra ottica e posizionarli in modo ottimale sopra i LED e il sensore di luminosità.
Come fibra ottica viene utilizzato un cavo PMMA da 1,5 mm. Si ottengono queste fibre in rotoli di circa 100 metri di lunghezza per circa 20 euro. Ogni gruppo di studenti che vuole collegare due CPX richiede 2 per 1 metro di cavo. Un cavo come linea di trasmissione e un cavo come linea di ricezione. Con un tale ruolo, diverse classi possono essere fornite con fibre ottiche a basso costo.

Il diametro di 1,5 mm garantisce che gli studenti non possano farsi male così facilmente. Inoltre, le fibre di questo spessore sono robuste contro la piegatura o la deformazione.

La parte superiore dell'involucro stampato in 3D ha molti fori, che sono intesi come guida per questi cavi. Il foro pre-costruito ha un diametro di 1,8 mm.

Figura 7: Trapano a mano

Se, a causa di una pressione rapida e di scarsa qualità, il cavo non può essere spinto attraverso le aperture dell'involucro, può essere rielaborato con un piccolo trapano a mano. Questi trapani a mano sono guidati a mano e quindi innocui; di solito vengono forniti con una buona attrezzatura di base su vari trapani.

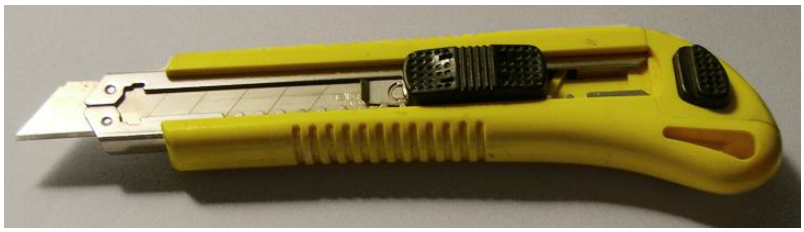


Figure 3: Cutter

Per tagliare le fibre di vetro, non si dovrebbero usare pinze o forbici perché comprimono la fibra di vetro. È necessario un taglio liscio dal quale la luce possa uscire. Pertanto, si consiglia di utilizzare un coltello per tappeti.

L'Adafruit CPX

In questo lavoro utilizziamo il microcontrollore a base di Python "Circuit Python Express" della società "Adafruit Industries". Questa scheda è un modo conveniente per utilizzare un microcontrollore insieme a molti attuatori e sensori prefabbricati e integrati in classe.

Il CPX ha dimostrato di essere robusto ed affidabile in molteplici applicazioni didattiche. Inoltre, Adafruit offre anche molte informazioni sul microcontrollore. Oltre a numerosi programmi campione, c'è anche una buona documentazione.

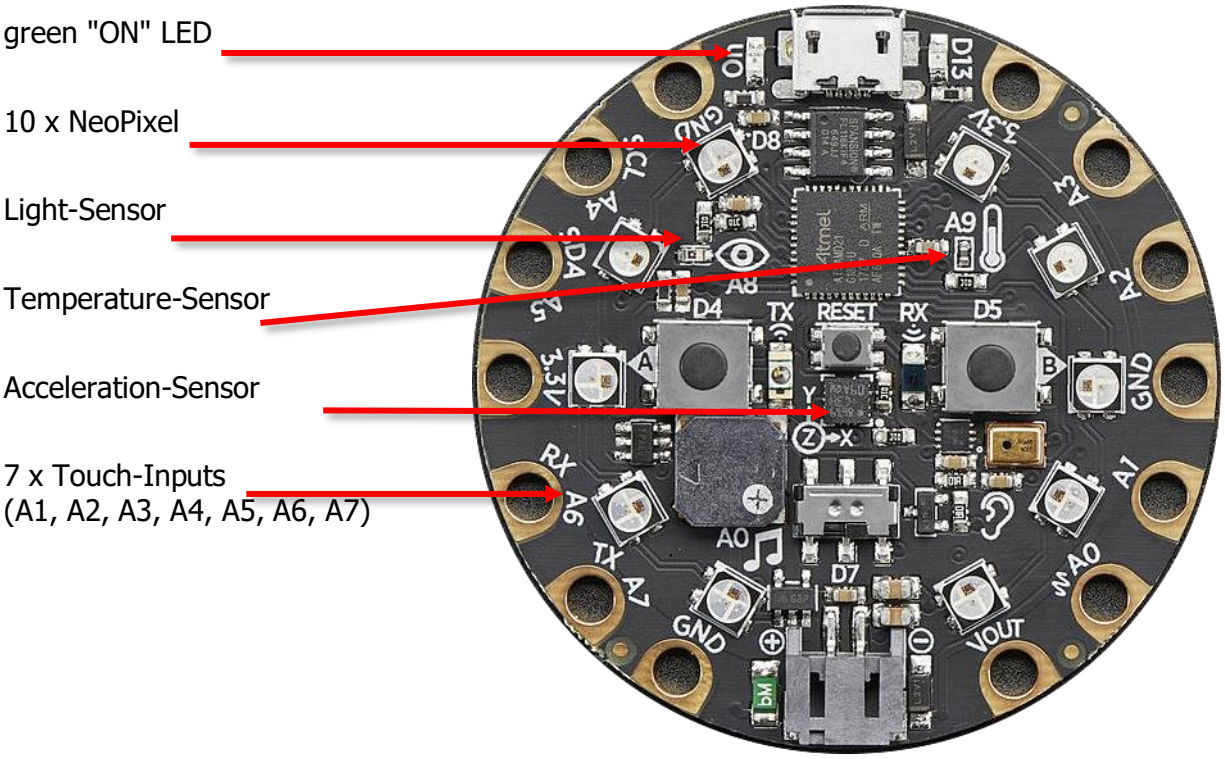


Figure 4: some features of the CPX

Un tutorial su come introdurre la programmazione del CPX insieme a molti programmi di esempio può essere trovato qui:
https://iludis.de/?page_id=291

Attrezzatura necessaria

I computer con cui gli studenti lavorano dovrebbero avere installato il seguente software:

- Autodesk Fusion 360 (o qualsiasi altro software di modellazione 3D, ad esempio Wings3D)
- Software di slicing CURA,
- Una connessione internet per scaricare le biblioteche
- **Mu Editor (https://codewith.mu/)**

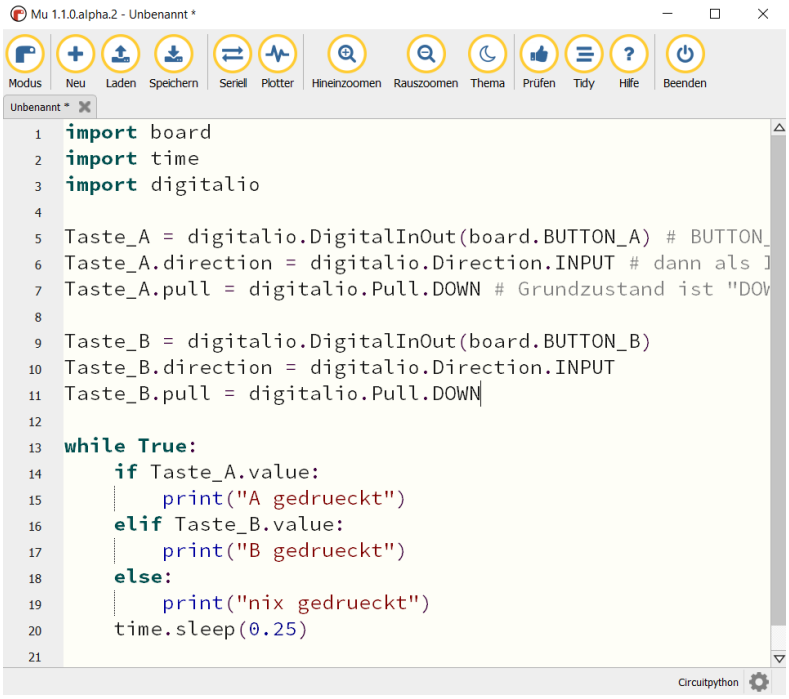


Figura 10: Screenshot Mu Editor

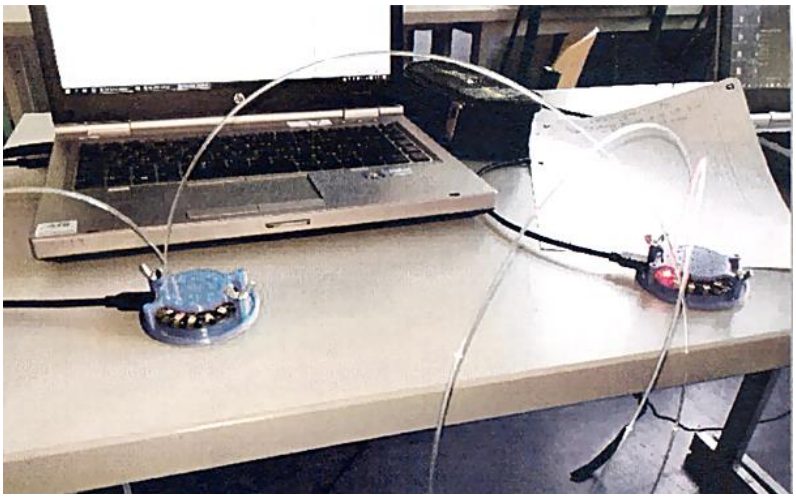


Figura 11: Fotografia dell'allestimento degli studenti

9. Piano di lezioni: Descrizione passo dopo passo dell'attività/contenuto

Lezioni 1 & 2 (90min): Introduzione all' IoT

Gli studenti saranno introdotti con esempi di IoT: Robot sottovuoto con app remote, stazioni meteorologiche basate su internet, smart farming e, non ultime, applicazioni per la salute. Gli studenti dovrebbero esaminare come funzionano questi dispositivi e quali componenti sono necessari: un sistema basato su microcontrollore controlla e coordina i sensori e gli attori collegati. Inoltre comunica e coordina con altri sistemi di tipo simile spesso tramite reti di comunicazione wireless. Parti necessarie: Sensori, attori, dispositivi di comunicazione. Occorre discutere delle possibilità e delle minacce e anche delle limitazioni: dove ha senso e dove non ha senso l'Internet degli oggetti?

Lezioni 2 & 3 (90min): Introduzione alla teoria dei protocolli di comunicazione

Gli studenti della classe fanno un gioco di ruolo: la classe guarda e sviluppa idee. Si applicano le seguenti regole:

- Due studenti "trasmettitore" e "ricevitore" si siedono insieme su sedie, che stanno in piedi schiena contro schiena, in modo che entrambi non possano vedersi tra i due sta una terza sedia vuota.
- Il trasmettitore dovrebbe inviare al ricevitore due parole completamente diverse, che vengono pronunciate all'indietro con un significato completamente diverso, come le coppie di parole:
- (4 lettere: "LIVE"/EVIL" e "STAR"/ "RATS" | 3 lettere: "DIO"/ "CANE" e "RAW/WAR")
- Come per la lingua parlata, la trasmissione delle parole permette una sola sequenza di singoli caratteri in successione, quindi tutte le lettere devono essere trasmesse singolarmente. Pertanto, le lettere delle due parole sono scritte su singole note.
- Queste note possono essere trasferite una ad una dal mittente al destinatario solo mettendo una nota sulla sedia e il destinatario che la prende, senza che le due parti possano vedersi.
- L'unica comunicazione diretta consentita tra le due parti è un unico tono (ad es. "bip") che tutti possono dare.
- Se una comunicazione va male, la trasmissione del messaggio viene interrotta, viene stabilita una nuova regola e gli studenti ricominciano da capo.

Ha senso che gli studenti si rendano conto che la comunicazione deve avvenire attraverso regole concordate di comune accordo, cioè un protocollo:

Gli studenti devono concordare un segnale di inizio. Dopo ogni lettera deve esserci una pausa di tempo perché i caratteri arrivino nell'ordine corretto; questo può essere concordato con un orologio o con un "bip". Inoltre, deve essere concordato l'ordine in cui le lettere vengono scambiate. In caso contrario, le parole avranno un significato diverso e non voluto. Infine, la comunicazione deve essere interrotta.

Lezioni 4 & 5 & 6 (120min): Principi della comunicazione seriale

Principi di multiplexing e demultiplexing: I bit di dati sono trasmessi uno per uno, a partire dall'MSB ("bit più significativo") all'LSB ("bit meno significativo") in un ordine specifico attraverso un unico cavo dati.

Si distingue tra trasmissione dati sincrona e asincrona: Nel caso più semplice di sincrono, il ricevitore - che funziona come master - specifica la frequenza di clock comune con la quale attiva la trasmissione dei singoli bit sul trasmettitore (slave). Dopo un numero prestabilito di bit trasmessi, il trasferimento dei dati è terminato.

Nel caso asincrono, è necessario concordare in anticipo un tempo di ciclo per entrambi i dispositivi coinvolti nella comunicazione, anche se questi tempi possono differire solo in minima parte l'uno dall'altro.

Figure 5: Emile Baudot

Lezione 7 (45min):

Storia della trasmissione dei dati: Émile Baudot

Utilizzando il codice baudot a 5 bit, le basi della trasmissione dei dati sono sviluppate con l'esempio pratico. Se si desidera trasferire solo lettere maiuscole, sono necessari 26 diversi caratteri di lettera ed eventualmente 2-3 segni di punteggiatura, come lo spazio o il punto interrogativo. Per codificare questi meno di 32 simboli diversi con i bit, sono necessari 5 bit; la codifica possibile sarebbe, ad esempio, la codifica:

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

Gli studenti dovrebbero prendere in considerazione un sistema che possa essere usato per codificare le lettere in modo bitwise-ideativamente, se ne escono con idee simili a quelle della tabella qui sopra.

Poiché i bit vengono trasmessi a una certa velocità, si parla di un cosiddetto baud rate, che prende il nome dall'ingegnere francese Emile Baudot. Qui si può entrare nella biografia di Baudot.

Lezioni 8 & 9 (90min): Ripetizione Riflessione totale

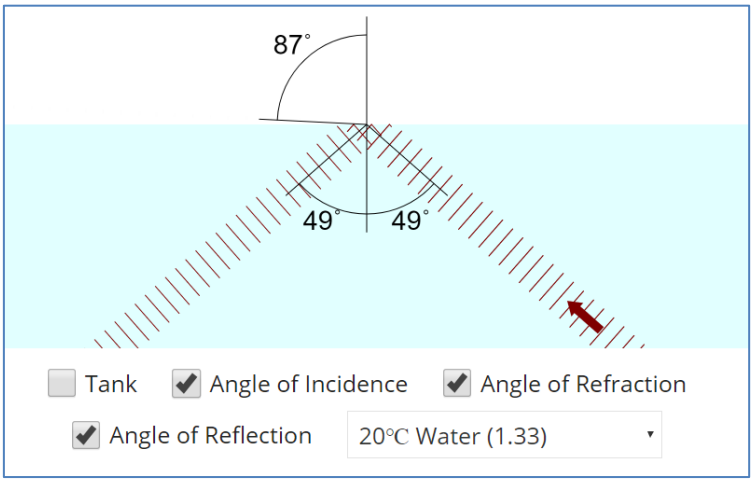
Riflessione totale

Utilizziamo le simulazioni "Rifrazione della luce": https://javalab.org/en/light_refraction_en/
E "Totale Riflessione interna": https://javalab.org/en/total_internal_reflection_en/

Per il passaggio dall'acqua all'aria,
"Il fascio di luce viene interrotto dalla saldatura durante il passaggio dal mezzo otticamente più denso (acqua) al mezzo otticamente meno denso (aria)".

Se l'angolo di incidenza del fascio di luce nell'acqua supera il cosiddetto "angolo critico", allora l'angolo di rifrazione nell'aria media dovrebbe essere maggiore di 90° - e questo è impossibile! Pertanto, il fascio di luce non ha altra scelta se non quella di rimanere nell'acqua.

1) angoli critici quando si passa da diversi supporti all'aria
https://javalab.org/en/light_refraction_en/



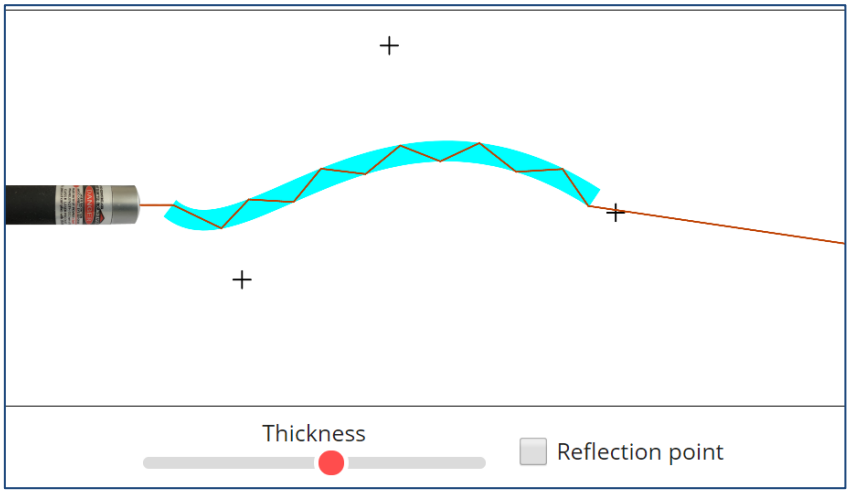
Determinare gli angoli critici - cioè quegli angoli di incursione in cui il fascio di luce NON esce dal mezzo:

Medio	L'angolo critico, sopra è la riflessione totale
Acqua	49°
Diamante	
Zaffiro	
Vetro	

2) Interpreta le due immagini seguenti:

Interpretare le aree selezionate nelle due immagini con parole proprie. Come nascono queste riflessioni? E cosa vedete esattamente nell'Area 3?

3) Applicazioni della riflessione totale
https://javalab.org/en/total_internal_reflection_en/



Sul lato sinistro si trova un laser che alimenta il suo fascio di luce in una cosiddetta fibra di vetro. Le fibre ottiche sono utilizzate per trasportare le informazioni con la luce. Da un lato i segnali del computer a fibre ottiche vengono irradiati per mezzo di un raggio laser e dall'altro lato, per mezzo di un sensore di luce, i segnali di luce vengono ricevuti e trasmessi al computer di destinazione. Spiegare il principio di funzionamento della fibra ottica, rispondendo alle seguenti domande:

1. Perché il raggio di luce rimane nella fibra?
2. Ci sono situazioni in cui il fascio di luce fuoriesce dalla fibra troppo presto?
3. In che modo la linea di luce all'interno della fibra dipende dallo spessore della fibra?

Lezioni 10 & 11 (90min), Introduzione alla programmazione con Python:
il CPX viene collegato al computer tramite cavo USB e programmato con l'ambiente di programmazione "mu-Editor". Ecco alcuni script per abituarsi al CPX. Quelle: https://iludis.de/?page_id=291

Script 1, "Lampeggio", accensione e spegnimento dei LED:

```
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, "bip", gioca a Tone:

```
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)
```

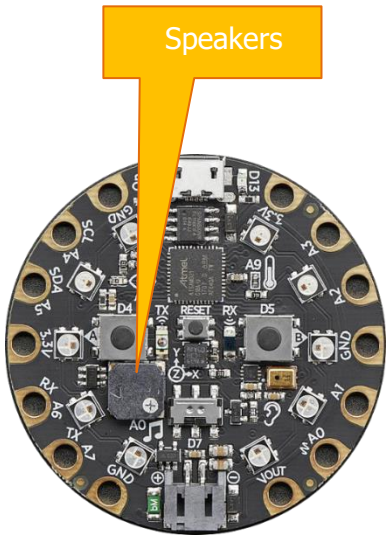


Figura 13: Posizione dell'altoparlante

Script 3, "Touch", Riprodurre il suono quando si preme il tasto Touch:

```
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, Commutazione dei 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)
```

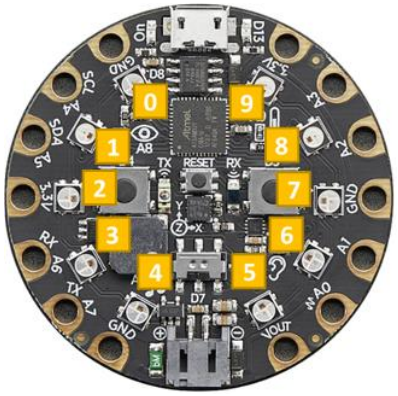


Figura 14: La numerazione dei neopixel

Skript 5, lettura del sensore di luce

```
import board
import time
import analogio

licht = analogio.AnalogIn(board.LIGHT)

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

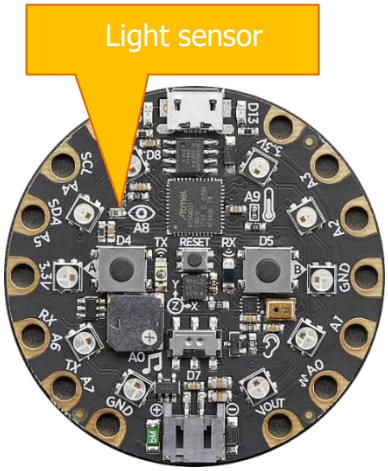


Figura 15: Posizione del sensore di luce

Lezioni 12 & 13 (90 min): Esempio di comunicazione sincrona

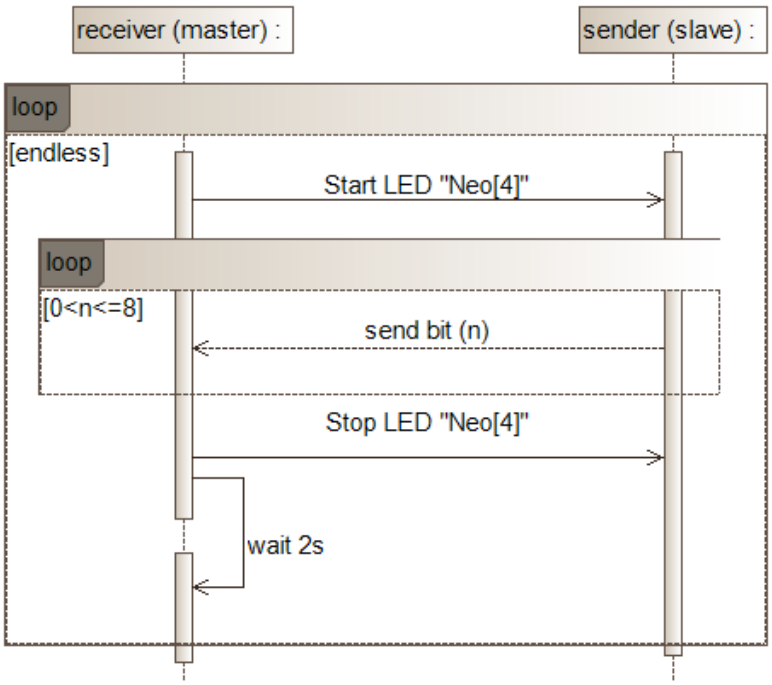


Figura 16: diagramma UML-Sequence della comunicazione sincrona

codice sorgente del ricevitore

```
import board
import time
import neopixel
import analogio

light = analogio.AnalogIn(board.LIGHT)
Neo=neopixel.NeoPixel(board.NEOPIXEL,1
0)
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)
```

codice sorgente del mittente

```
import board
import time
import neopixel
import analogio

light = analogio.AnalogIn(board.LIGHT)
Neo=neopixel.NeoPixel(board.NEOPIXEL,1
0)
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)
```

Lezioni 14 & 15 (90 min): Esempio di comunicazione asincrona

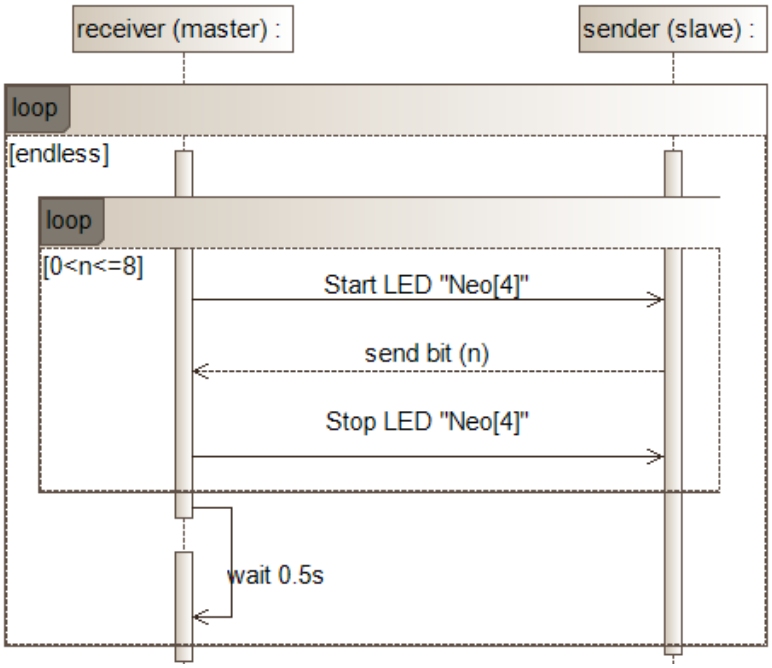


Figura 17: diagramma UML-Sequence della comunicazione asincrona

codice sorgente del ricevitore	codice sorgente del mittente

`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	Alla fine del modulo, gli studenti dovrebbero aver sviluppato una comprensione più profonda di come funziona la comunicazione seriale e quali principi informatici sono necessari per implementare questa tecnologia. Durante le lezioni vengono discussi importanti aspetti dell'elettronica, dell'ottica e della costruzione del protocollo.
11. Valutazioni	Gli studenti tengono il loro diario del lavoro, che può essere rivisto dall'insegnante. Gli studenti possono anche presentare i risultati dei loro esperimenti. Inoltre, alla fine delle lezioni deve essere effettuato un test standard in classe.