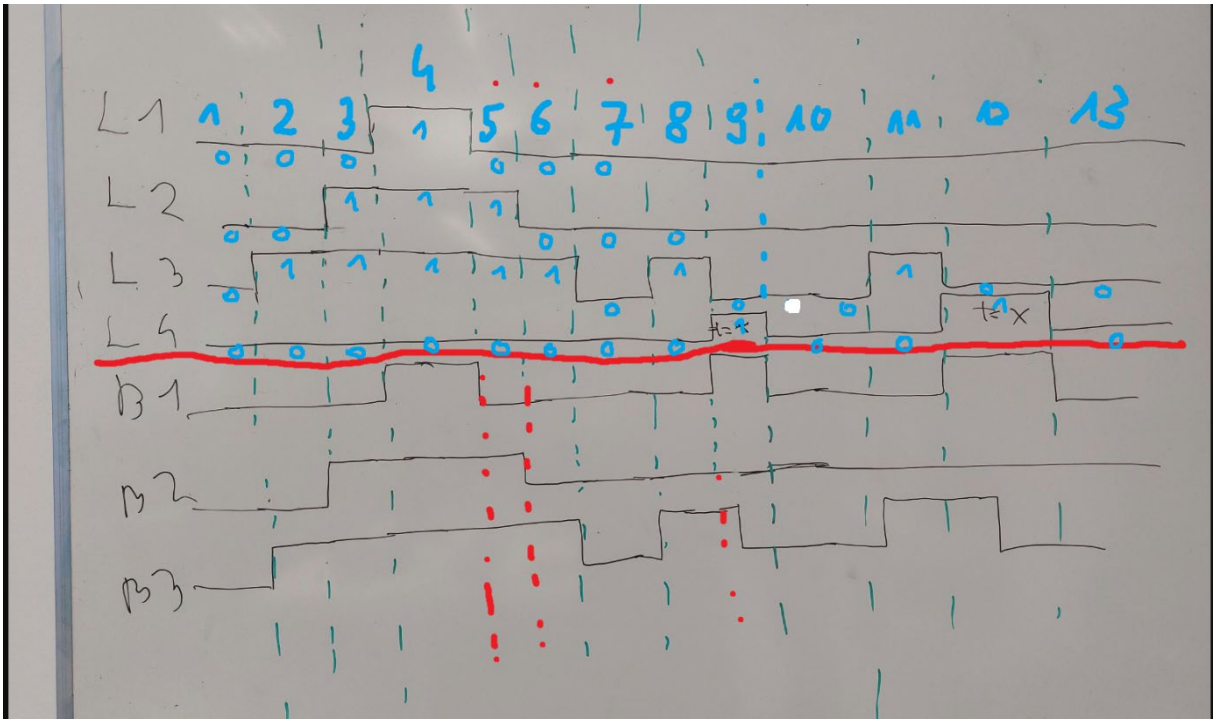
Zadanie

Na podstawie narysowanego przebiegu czasowego dla diód i przycisków napisz program na maline, by sterował diodami wedle sygnałów od przycisków, zgodnie z przebiegiem.



Rozwiązanie

0 – stan NISKI (LOW)

1 – stan WYSOKI(HIGH)

DLA L1	1	2	3	4	5	6	7	8	9	10	11		
L1	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
B1	0	0	0	1	0	0	0	0	1	0	0	1	0
B2	0	0	1	1	1	0	0	0	0	0	0	0	0
B3	0	1	1	1	1	1	0	1	0	0	1	0	0

DLA L2	1	2	3	4	5	6	7	8	9	10	11		
L2	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
B1	0	0	0	1	0	0	0	0	1	0	0	1	0
B2	0	0	1	1	1	0	0	0	0	0	0	0	0
B3	0	1	1	1	1	1	0	1	0	0	1	0	0

DLA L3	1	2	3	4	5	6	7	8	9	10	11	12	13
L3	**0**	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**
B1	0	0	0	1	0	0	0	0	1	0	0	1	0
B2	0	0	1	1	1	0	0	0	0	0	0	0	0
B3	0	1	1	1	1	1	0	1	0	0	1	0	0

DLA L4	1	2	3	4	5	6	7	8	9	10	11	12	13
L3	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**
B1	0	0	0	1	0	0	0	0	1	0	0	1	0
B2	0	0	1	1	1	0	0	0	0	0	0	0	0
B3	0	1	1	1	1	1	0	1	0	0	1	0	0

L1:

Widać, że dioda L1 włącza się tylko wtedy, gdy pojawia się stan WYSOKI na przyciskach B1, B2 i B3.

If (B1= HIGH && B2 = HIGH && B3 = HIGH) {

Włącz diode L1

}

Else {

Wyłącz diode L1

}

L2:

Dioda włącza się tylko wtedy, gdy pojawia się stan wysoki na przycisku B2.

If (B2 = HIGH) {

L2 = HIGH

}

Else{

L2 = LOW

}

L3:

Dioda włącza się tylko wtedy, gdy stan na przycisku B3 jest wysoki.

If (B3 = HIGH) {

L3 = HIGH

}

Else {

L3 = LOW

}

L4:

Dioda włącza się tylko wtedy, gdy przycisk B1 ma stan wysoki a B2 i B3 stan niski.

If (B1 = HIGH && B2 = LOW && B3 = LOW) {

L4 = HIGH

}

Else {

L4 = LOW

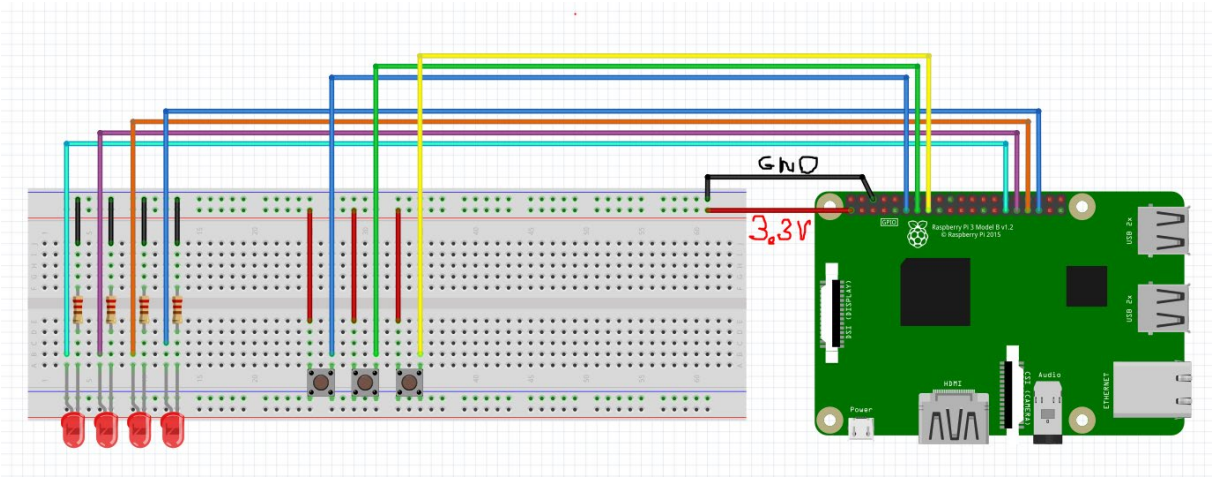
}

Program w pythonie:

```
import RPi.GPIO as GPIO
import time


# Konfiguracja pinów
BUTTONS = [17, 27, 22]  # Piny GPIO dla przycisków: B1, B2, B3
LEDS = [5, 6, 13, 19]   # Piny GPIO dla diod: L1, L2, L3, L4


# Ustawienie trybu numeracji i konfiguracja pinów
GPIO.setmode(GPIO.BCM)
for button in BUTTONS:
    GPIO.setup(button, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)


for led in LEDS:
    GPIO.setup(led, GPIO.OUT)
    GPIO.output(led, GPIO.LOW)


try:
    while True:
        # Odczyt stanów przycisków
        B1 = GPIO.input(BUTTONS[0])
        B2 = GPIO.input(BUTTONS[1])
        B3 = GPIO.input(BUTTONS[2])
```

```
        if B1 == GPIO.HIGH and B2 == GPIO.HIGH and B3 == GPIO.HIGH: #Logika L1
            GPIO.output(LEDS[0], GPIO.HIGH)
        else:
            GPIO.output(LEDS[0], GPIO.LOW)

        # Logika dla L2
        if B2 == GPIO.HIGH:
            GPIO.output(LEDS[1], GPIO.HIGH)
        else:
            GPIO.output(LEDS[1], GPIO.LOW)

        # Logika dla L3
        if B3 == GPIO.HIGH:
            GPIO.output(LEDS[2], GPIO.HIGH)
        else:
            GPIO.output(LEDS[2], GPIO.LOW)

        # Logika dla L4
        if B3 == GPIO.LOW and B2 == GPIO.LOW and B1 == GPIO.HIGH:
            GPIO.output(LEDS[3], GPIO.HIGH)
            time.sleep(1) #dodane, żeby dioda od razu nie gasła
        else:
            GPIO.output(LEDS[3], GPIO.LOW)
    # Krótkie opóźnienie, aby uniknąć błędów odczytu
        time.sleep(0.1) #w sekundach

except KeyboardInterrupt:
    print("Przerwanie programu")
finally:
    GPIO.cleanup()
```

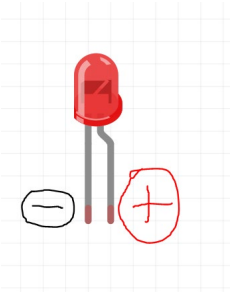
Układ:



Ważne – w kodzie można dodać linię

```
GPIO.setup(button, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

Wtedy nie trzeba dawać rezystorów pull down albo pull up do przycisków na płytce



dłuższa nóżka(plus), krótsza nóżka(minus)

Jak to zaprogramować?

- Ogarnąć wtyczkę do ssh do vscode i jeżeli będzie vsc na lapku to można się jakoś bawić.
- Tak jak na labach:
    1. Podłączamy raspberry pi do zasilania i czekamy chwilę, żeby połączyło się z siecią
    2. W cmd: ssh pi@192.168.50.XXX – ostatnie 3 cyfry są naklejone na maline
    3. Hasło: raspberry (chyba, że zmienią)
    4. Odpalamy nano
    5. Tworzymy plik z kodem i zapisujemy .py
    6. Odpalamy: python3 nazwa_programu.py

Dodatki:

https://pinout.xyz/ - strona do sprawdzania pinoutu Raspberry

Wgrywanie kodu tak jak na labach – edytor tekstu nano

Otworzenie już zapisanego wcześniej pliku: nano nazwa_pliku.py

Ctrl + O – zapisz plik

Ctrl + X – wyjdz z edytora

Wszystkie skróty są wyświetlane w dolnej części edytora

Uruchomienie programu:

python3 nazwa_pliku.py

Zatrzymanie programu:

CTRL + C