

קלט פלט עם ראסברי פיי פיקו

שפת התוכנה פייתון והגרסה שלה למיקרו בקרים - מיקרו פייתון - הן שפות מונחית עצמים. כמעט כל דבר בפייתון הוא מחלקה - class - וכאשר יוצרים instance (מופע, דוגמה) מקבלים אובייקט. למחלקה יש מתודות (פונקציה השייכת למחלקה). המתודה (הנקראת גם פונקציית חבר - member function) משמשת לאינטראקציה עם האובייקט או לבקרה עליו .

במאמר זה נקרא בחלק מהמקומות לכרטיס הראספברי פיי פיקו בקצרה פיקו.

שים לב : במידה והתוכנית רצה בלולאה אין סופית ורוצים לעצור את הריצה שלה יש להקיש על המקשים : CTRL ו C יחד .

1. פלט עם ראספברי פיי פיקו

בדוגמאות שנתאר בהמשך נייבא מהמודול machine את המחלקה Pin . הסבר נרחב יותר על המודול machine והמחלקה Pin ניתן למצוא בקישור :

<http://www.arikporat.com/raspberry%20pi%20pico/Micropython%20introduction.pdf>

בסעיף כאן נשתמש ב 3 מתודות (פונקציות חבר) השייכות למחלקה Pin להוצאת 0 או 1 להדק כלשהו והן :

on() , off() ו toggle() .

: דוגמה מספר 1

הדוגמה הראשונה שנתאר היא הדלקת ה led שבכרטיס הפיקו. ה led מתחברת להדק 25 . כל לחיצה על RUN בתוכנת ה Thonny תהפוך את מצב ה led. לחיצה ראשונה על RUN תדליק את ה led. הלחיצה הבאה תכבה אותה, הבאה תדליק וכך הלאה. התוכנית נראית כך :

```
from machine import Pin
```

```
led = Pin(25, Pin.OUT)
```

```
led.toggle()
```

בשורה הראשונה ייבאנו מהמודול machine את המחלקה Pin . (במילה machine שפרושה בעברית מכונה מתכוונים למיקרו בקר שבכרטיס. המודול מטפל במיקרו בקר ובחומרה שבכרטיס).

בשורה הבאה יצרנו אובייקט שקראנו לו led והשתמשנו במחלקה Pin כדי להגדיר שהדק 25 בכרטיס הפיקו הוא הדק יציאה - OUT .

בשורה האחרונה הפעלנו על האובייקט את המתודה `toggle()` . במתודה רשומות פקודות ההופכות את מצב הדק 25 שבה נמצאת הLED בכרטיס הפיקו.

ואכן בהרצה נראה שכל לחיצה על RUN בתוכנת ה Thonny תחליף את מצב הLED.

דוגמה מספר 2 : בדוגמה זו נהבה את הLED. הLED תידלק ותכבה באופן מחזורי ללא צורך ללחוץ על לחצן RUN. התוכנית נראית כך :

```
from machine import Pin
import time
led = Pin(25, Pin.OUT)
while True:
    led.high()      # led.on() אפשר גם לרשום
    time.sleep_ms(250)
    led.low()       # led.off() אפשר גם לרשום
    time.sleep_ms(250)
```

ההבדלים בין דוגמה מספר 1 לדוגמה מספר 2 הם :

בשורה שנייה ייבאנו את המודול `time` .

בשורה 4 עשינו לולאת `while` המבצעת את השורות שבאות אחריה בלולאה אין סופית .

בשורה החמישית הדלקנו את הLED ובורה השישית ביצענו השהייה של 250 מילי שניות .

בשורה השביעית כיבינו את הLED ובשורה השמינית עשינו השהייה של 250 מילי שניות.

סיכום : הLED תהיה 250 מילי שניות ב ON ו 250 מילי שניות היא תהיה ב off . זמן המחזור (on+off) שווה ל 500 מילי שניות כלומר התדר הוא 2 הרץ. הווה אומר שבהרצת התוכנית הLED תהבהב בקצב של 2 פעמים בשנייה.

נלחץ על RUN ב Thonny ונקבל 2 הבהובים של הLED בשנייה.

דוגמה מספר 3 : בדוגמה זו נבצע לולאה של 10 פעמים בה נדליק את הLED לזמן של 200 מילי שניות ונכבה אותה לזמן של 200 מילי שניות.

```
import time
```

```

from machine import Pin

led = Pin(25, Pin.OUT)

for x in range (10) :

    led.on()

    time.sleep_ms(200)

    led.off()

    time.sleep_ms(200)

```

דוגמה מספר 4 : בדוגמה זו נהבהב את הled בצורה מחזורית . 5 פעמים הled תהבהב 5 פעמים אחר כך נבצע השהייה של 3 שניות ללא הבהוב ונחזור להבהב את הled 5 פעמים וכך הלאה. התוכנית נראית כך :

```

from machine import Pin

import time

led = Pin(25, Pin.OUT)

while True :

    for x in range (10) :

        led.toggle()

        time.sleep_ms(250)

    time.sleep_ms(3000)

```

דוגמה מספר 5 : בדוגמה כאן נהבהב את הled בעזרת טיימר. התוכנית נראית כך :

```

from machine import Pin, Timer # Timer , Pin המחלקות ייבוא
led = Pin(25, Pin.OUT) # הגדרת אובייקט
LED_state = True # True והעברה אליו את הערך # True
tim = Timer() # timer יצירת אובייקט של
def tick(timer): # timer ל הנכנס ערך המקבלת ערך הנכנס ל
    global led, LED_state # המשתנים הגלובאליים

```

```

LED_state = not LED_state # הפיכת מצב המשתנה
led.value(LED_state)      # מוציאים להדק את הערך שבמשתנה ( פעם 0 ופעם 1 לחילופין)
# tick אתחול האובייקט לתדר 2 הרץ , הטיימר סופר במחזוריות ובכל פסיקה של הטיימר נפעיל את הפונקציה
tim.init(freq=2, mode=Timer.PERIODIC, callback=tick)

```

2. קלט עם ראספברי פיי

בדוגמאות שנתאר בהמשך נייבא מהמודול machine את המחלקה Pin . הסבר נרחב יותר על המודול machine והמחלקה Pin ניתן למצוא בקישור :

<http://www.arikporat.com/raspberry%20pi%20pico/Micropython%20introduction.pdf>

דוגמה מספר 1 : קריאת המצב של מפסק והדפסה למסך ה Shell. בדוגמה זו נחבר מפסק אל הדק 4 של הפיקו. הצד השני של המפסק מחובר לאדמה. המפסק מסוג Normally Open – N.O – בדרך כלל פתוח. כאשר נלחץ על המפסק האדמה תעבור דרך המפסק אל ההדק ובהדק יהיה '0'. נגדיר את הדק 4 כקלט עם נגד משיכה למעלה. במצב זה כאשר המפסק פתוח יש בהדק '1' ("גבוה") וכאשר לוחצים על המפסק יש בהדק '0' ("נמוך"). נבדוק באיזה מצב המפסק ונדפיס ל Shell של ה Thonny את המילה up אם המפסק פתוח ובהדק יש גבוה. התוכנית נראית כך :

```

from machine import Pin

button = Pin(4, Pin.IN, Pin.PULL_UP)

while True:

    if (button.value() == 0):

        buttonStatus = "down"

    else:

        buttonStatus = "up"

    print( "Button Status: " + buttonStatus)

```

בשורה הראשונה ייבאנו את המחלקה Pin שבמודול machine .

בשורה השנייה יצרנו אובייקט בשם button והשתמשנו במחלקה Pin כדי להגדיר שהדק 4 בכרטיס הפיקו הוא הדק קלט עם נגד משיכה למעלה. בשורות הבאות עשינו לולאה אין סופית שבה בודקים את מצב המפסק ובהתאמה מדפיסים את מצבו למסך ה Shell .

הרץ את התוכנית ובדוק מה קורה כאשר המפסק לא לחוץ (נקבל up) וכאשר הוא לחוץ (נקבל down).

דוגמה מספר 2 : קריאת המצב של מפסק והבהוב או כיבוי של הLED בהתאמה . בדוגמה זו נהבהב את הLED שבהדק 25 בכרטיס הפיקו בהתאם למצב המפסק. אם המפסק לא לחוץ הLED כבוי . אם המפסק לחוץ הLED מהבהב בקצב של פעמיים בשנייה . התוכנית נראית כך :

```
from machine import Pin
from time import sleep

led = Pin(25, Pin.OUT)
button = Pin(4, Pin.IN, Pin.PULL_UP)

def led_toggle(): # הגדרת פונקציה
    led.value(not led.value())

while True:
    if not button.value():
        led_toggle() # קריאה לפונקציה
        sleep(.25) # (on+off=0.5 sec) 0.25 שניות
```