الوحدة 1 | الدرس 5

لعبة الاستجابة السريعة

وصف الدرس

الغرض العام من هذا الدرس هو أن يتعلم الطلبة كيفية برمجة الأزرار للتحكم في سير البرنامج. سيُنشئ الطلبة برنامجًا لتشغيل ثنائي ضوئي (LED) عند الضغط على الزر، و سيقومون أيضًا بإنشاء "لعبة الاستجابة السريعة" بثنائي ضوئي واحد ومفتاحين.

ما سيتعلمه الطالب

> توصيل وبرمجة مفتاح للتحكم بدارة كهربائية.

نتاجات التعلم

> كيفية توصيل وبرمجة مفتاح للتحكم بدارة كهربائية.

المصطلحات

اللغة العربية	اللغة الإنجليزية
كابل شريطي	Ribbon cable
لوحة جانبية	Breakout board

التحديات المتوقعة

> قد يواجه الطلبة صعوبة في فهم الاختلاف بين استخدام **Raspberry Pi** مع لوحة جانبية **(Breakout)** وبدونها، حيث يعتقد الكثيرون أن استخدامها غير ضروري. اشرح لهم بأن الدارتين متماثلتان في عملهما، ولكن من الأفضل استخدام اللوحة الجانبية، حيث تنقل جميع المنافذ من **Raspberry Pi** إلى لوحة تجارب **Breadboard**، مما يتيح الاستغناء عن توصيل الكابلات بـ **Raspberry Pi**، حيث يمكنك توصيل جميع الأجزاء الإلكترونية مباشرة بلوحة التجارب.

> يجب على الطلبة توخي الحذر عند توصيل أطراف اللوحة الجانبية بالثقوب الصحيحة في لوحة التجارب. اشرح لهم تأثير توصيل اللوحة الجانبية بلوحة التجارب.

> يجب التنويه للطلبة بضرورة الضغط برفق على الزر أثناء وضع السيقان داخل الثقوب خلال توصيل الزر بلوحة التجارب، وذلك لتجنب كسر سيقان ذلك الزر.

التمهيد

> قدِّم الغرض من الدرس بتحفيز اهتمام الطلبة في إنشاء لعبة الاستجابة السريعة.

> ابدأ بطرح بعض الأسئلة على الطلبة على مثل:

- هل سبق لكم ممارسة لعبة تتطلب القيام برد الفعل السريع لتحقيق الفوز؟
- هل سبق لكم إنشاء لعبة؟
- هل يمكنكم التنبؤ بسبب الحاجة إلى زرين في اللعبة التي سيتم إنشاؤها في هذا الدرس؟

التلميحات الخاصة بالتنفيذ

> اطلب من الطلبة تفحُّص إحدى اللوحات الجانبية قبل توصيلها. ذكِّر الطلبة بمعاينة الصورة التي تمثل نظام **BCM** من الدرس الرابع صفحة 61 في كتاب الطالب. اشرح لهم أن هناك تسمية خاصة لكل منفذ موجود في اللوحة الجانبية.

> بعد أن يقوم الطلبة بتوصيل اللوحة الجانبية بلوحة التجارب، وضِّح للطلبة تأثير ذلك على طريقة توصيل العناصر والمكونات المختلفة.

> اعرض بعض الأمثلة على الطلبة، حيث يمكنك مثلاً إنشاء دارة بسيطة عن طريق أخذ كابل واحد وتوصيل حافة واحدة بساق ثنائي ضوئي الطويلة والأخرى بأحد منافذ **GPIO**.

> أثناء توصيل الزر بلوحة التجارب، اشرح للطلبة أن منافذ **GPIO** في **Raspberry Pi** تعمل كأزرار تُخرج جهدًا قدره 3.3 فولت عند الضبط على **HIGH** (مرتفع)، ولا تُخرج أي جهد عند ضبطها على **LOW** (منخفض). وضِّح للطلبة أن الزر هو عبارة عن دارة بسيطة يمكنه أن يأخذ إحدى القيمتين: 0 (منخفض)، و 1 (مرتفع). تمثل هاتان القيمتان مدخلات/مخرجات منافذ **GPIO**. تكون قيمة الإدخال **LOW** (منخفض) أي (0 فولت) عندما لا يتم الضغط على الزر، وتكون **HIGH** (مرتفع) أي (3.3 فولت) عند الضغط عليه.

> قبل إنشاء الأمر الذي سيوقف تشغيل الثنائي الضوئي بعد وقت يتم تحديده بشكل عشوائي، أشر للطلبة أن هذا الأمر سيضيف تحديًا أكبر إلى لعبتهم لأن الثنائي الضوئي سينطفئ في وقت عشوائي يحدده الطلبة. وضِّح للطلبة أن تعيين أرقام هاتين القيمتين (**A** و **B**) سيوقف تشغيل الثنائي الضوئي لمدة يتم اختيارها عشوائيًا وتقع بين هاتين القيمتين، وفي حالتنا هذه فهي تتراوح بين 5-10 ثوان.

استراتيجيات غلق الدرس

في نهاية الدرس تأكد من تحقيق الطلبة لجميع أهداف الدرس وتقييم معرفتهم من خلال أسئلة على سبيل المثال لا الحصر:

> هل تستطيع أن تتذكر:

- كيف تعمل أزرار **Raspberry Pi**؟
- ما هو الغرض من استخدام الكابل الشريطي؟
- ما هي اللوحة الجانبية (**Breakout**) وكيف يمكن استخدامها في تسهيل عملية التوصيل؟

- كيف يمكن إيقاف تشغيل الثنائي الضوئي (LED) في وقت عشوائي؟

> ذكّر الطلبة بالمصطلحات الهامة وكرّرها معهم.

> يمكنك الاستعانة بتدريبات الكتاب ضمن الاستراتيجيات التي ستستخدمها لغلق الدرس.

التدريبات المقترحة لغلق الدرس

يمكنك استخدام التمرين الثامن ضمن استراتيجية غلق الدرس لتقييم وتعزيز قدرة الطلبة على تطبيق المهارات التي تم تقديمها في هذا الدرس.

الصف الحادي عشر | الفصل الأول | كتاب الطالب | صفحة 109

الفروق الفردية

تمارين إضافية للطلبة ذوي التحصيل المرتفع

> بعد الانتهاء من التمرين الثامن من هذا الدرس، اطلب من الطلبة توصيل ثنائي ضوئي (LED) ثالث بمقاومة في اللوحة الجانبية. اطلب من الطلبة برمجة الثنائي الضوئي (LED) ليتم تشغيله ثم إيقاف تشغيله بشكل عشوائي في مدة تتراوح بين 3-6 ثوانٍ.

```
led2 = LED(13)

#Adding an element of suprise
led.on()
sleep(uniform(3, 6))
led.off()
```

تلميح:

الإجابة المقترحة لهذا التمرين هي: بإضافة ثنائي ضوئي (LED) ثالث وإيقاف تشغيله في وقت عشوائي من 3-6 ثوانٍ.

الإجابات النموذجية للتدريبات:

1

ضع إشارة ✔ أمام الإجابة الصحيحة:

1. يُمكننا إضافة المكتبة في أي موقع داخل المقطع البرمجي لبرنامج بايثون.
   صحيح ○ خطأ ✔

2. توجد دالتان رئيستان خاصتان بالمفاتيح، الأولى هي when_pressed والأخرى when_released.
   صحيح ✔ خطأ ○

3. من المهم إضافة الملاحظات و التعليقات في البرنامج لتذكر كيف يعمل مستقبلاً.
   صحيح ✔ خطأ ○

4. الأمر Uniform يرجع جميع القيم ما بين a وb.
   صحيح ✔ خطأ ○

5. دالة When_released تُنفذ عند تغير حالة المفتاح من غير مفعل (OFF) إلى مفعل (ON).
   صحيح ○ خطأ ✔

2

لماذا يوصى باستخدام "الكابل الشريطي" (ribbon cable) و"اللوحة الجانبية" (breakout board) في الدارات الإلكترونية على لوحة Breadboard؟

يستخدم الكابل الشريطي لتوصيل أطراف GPIO الخاصة بـ Raspberry Pi باللوحة الجانبية (Breadboard) ، ولتوصيل الكابل الشريطي بلوحة Breadboard يتم استخدام اللوحة الجانبية Breakout board.

3

نستخدم كابلات التوصيل (jumper wires) لربط عناصر الدارات معًا. لماذا يعتبر من المهم استخدام ألوان مختلفة لتلك الكابلات؟

يمكنك استخدام ألوان مختلفة للكابلات لتمييز التوصيلات المختلفة التي تجريها على الدارة ولمساعدتك على تذكرها.

4

في البرنامج الذي أنشأناه سابقًا، قم باستبدال الدالة when_pressed بالدالة when_ released، ثم قم بتشغيل البرنامج. ما الذي تلاحظ تغيره في طريقة عمل البرنامج؟

سيتم تشغيل الثنائي الضوئي (LED) بمجرد بدء البرنامج، ثم سيتم إيقاف تشغيله بشكل عشوائي بين 5-10 ثوان. إذا لم يترك أي من اللاعبين الزر فلن يكون لدينا فائز، حتى يقوم أحدهما بتحرير الزر. وباختصار، سيكون لدينا فائز عند تحرير الزر وليس عند الضغط عليه.

تلميح:

حث الطلبة على عمل بعض التغييرات على البرنامج ثم اختباره وملاحظة تأثير تلك التغييرات.

5 افتح Python في Raspberry Pi وأنشئ برنامجًا يضيء LED عندما يتم الضغط على مفتاح، بينما يتم ايقاف تشغيل LED عند تحرير المفتاح من الضغط.

```
from gpiozero import LED,Button
from time import sleep

led = LED(24)
button=Button(21)
led.off()

def pressed(button):
    if button .pin.number==21:
        led.on()
def released(button):
    if button .pin.number==21:
        led.off()
button.when_pressed=pressed
button.when_released=released
```

تلميح:

هذه هي الإجابة المقترحة لهذا التمرين.

6

افتح البرنامج الذي أنشأناه سابقًا في هذا الدرس، ثم قم بإضافة LED جديد إلى لوحة Breadboard. قم بعمل التغييرات اللازمة على البرنامج بحيث تتم إضاءة LED الأيسر إذا ضغط اللاعب على الجهة اليسرى المفتاح أولاً، وسوى ذلك تتم إضاءة LED الأيمن.

```
from gpiozero import LED, Button
from time import sleep
from random import uniform
from sys import exit

#Get player names
left_name = input("left player name is ")
right_name = input("right player name is ")

left_led = LED(16)
right_led = LED(20)
led = LED(24)
left_button = Button(21)
right_button = Button(23)

#Adding an element of suprise
led.on()
sleep(uniform(5, 10))
led.off()

#Detecting the buttons
def pressed(button):
    if button.pin.number == 21:
        print(right_name + " won the game")
        left_led.on()
        sleep(1)
    else:
        print(left_name + " won the game")
        right_led.on()
        sleep(1)
    exit()

left_button.when_pressed = pressed
right_button.when_pressed = pressed
```

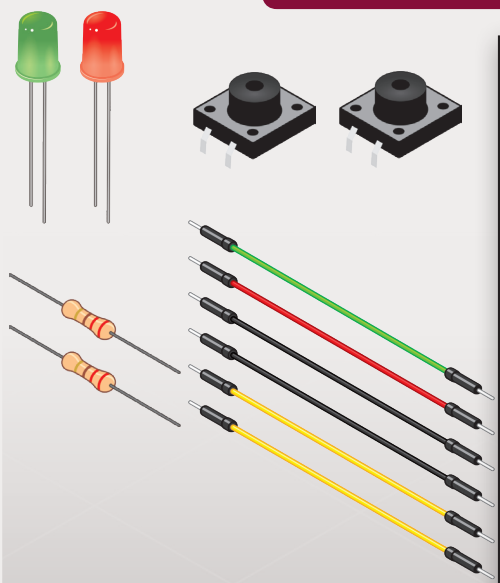
تلميح:

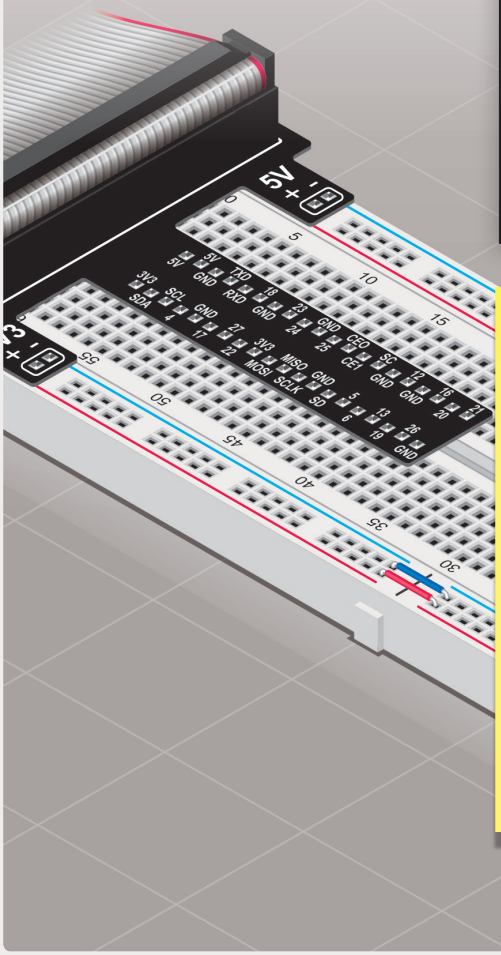
هذه هي الإجابة المقترحة لهذا التمرين.

7

وصِّل المكونات على لوحة Breadboard. ثم سجل أسماء المنافذ التي استخدمتها لكل مكون في الأماكن الصحيحة في الجدول.

المكون	الموضع			
طرف ثنائي LED الأحمر الطويل	I30			
طرف ثنائي LED الأحمر القصير	I25			
طرف ثنائي LED الأخضر الطويل	I20			
طرف ثنائي LED الأخضر القصير	I15			
المقاومة 1	GND 29	**F25**		
المقاومة 2	GND 38	**F15**		
الكابل الأحمر	H44	**J30**		
الكابل الأخضر	H43	**J20**		
الكابل الأصفر 1	C18	**A59**		
الكابل الأصفر 2	C17	**A27**		
الكابل الأسود 1	GND 48	**A57**		
الكابل الأسود 2	GND 21	**A25**		
المفتاح 1	**F3**	**F1**	**C57**	**C59**
المفتاح 2	**F33**	**F35**	**C25**	**C27**





تلميح:

حث الطلبة على إنشاء التوصيلات التي يحتاجون إليها لإنشاء هذا المشروع. يمكنهم أيضًا استخدام قلم رصاص لرسم المكونات على صورة لوحة التجارب في كتاب الطالب. إن الثقوب التي تم توصيل كل مكون من المكونات في الجدول أعلاه هي ثقوب إرشادية، حيث يمكن للطلبة استخدام هذه الثقوب أو أية ثقوب أخرى في لوحة التجارب لإكمال هذا التمرين. يرمز GND في الجدول أعلاه إلى الخط الأرضي.

8 املأ الصناديق بالأطراف الصحيحة. ثم افتح Raspbian واكتب المقطع البرمجي لاختباره.

> استخدم التوصيلات التي أنشأتها في التدريب السابق لملء الصناديق بأرقام أطراف GPIO.
> افتح raspbian واكتب المقطع البرمجي واختبره.

```
from gpiozero import LED, Button
from time import sleep
from sys import exit

led = LED( 21 )
led1 = LED( 26 )
left_button = Button( 20 )
right_button = Button( 21 )

#Detecting the buttons
def pressed(button):
    if button.pin.number == 20 :
        print('Left button is pressed')
        led.on()
        sleep(1)
        led.off()
        sleep(1)
    else:
        print('right button is pressed')
        led1.on()
        sleep(1)
        led1.off()
        sleep(1)
    exit()

left_button.when_pressed = pressed
right_button.when_pressed = pressed
```