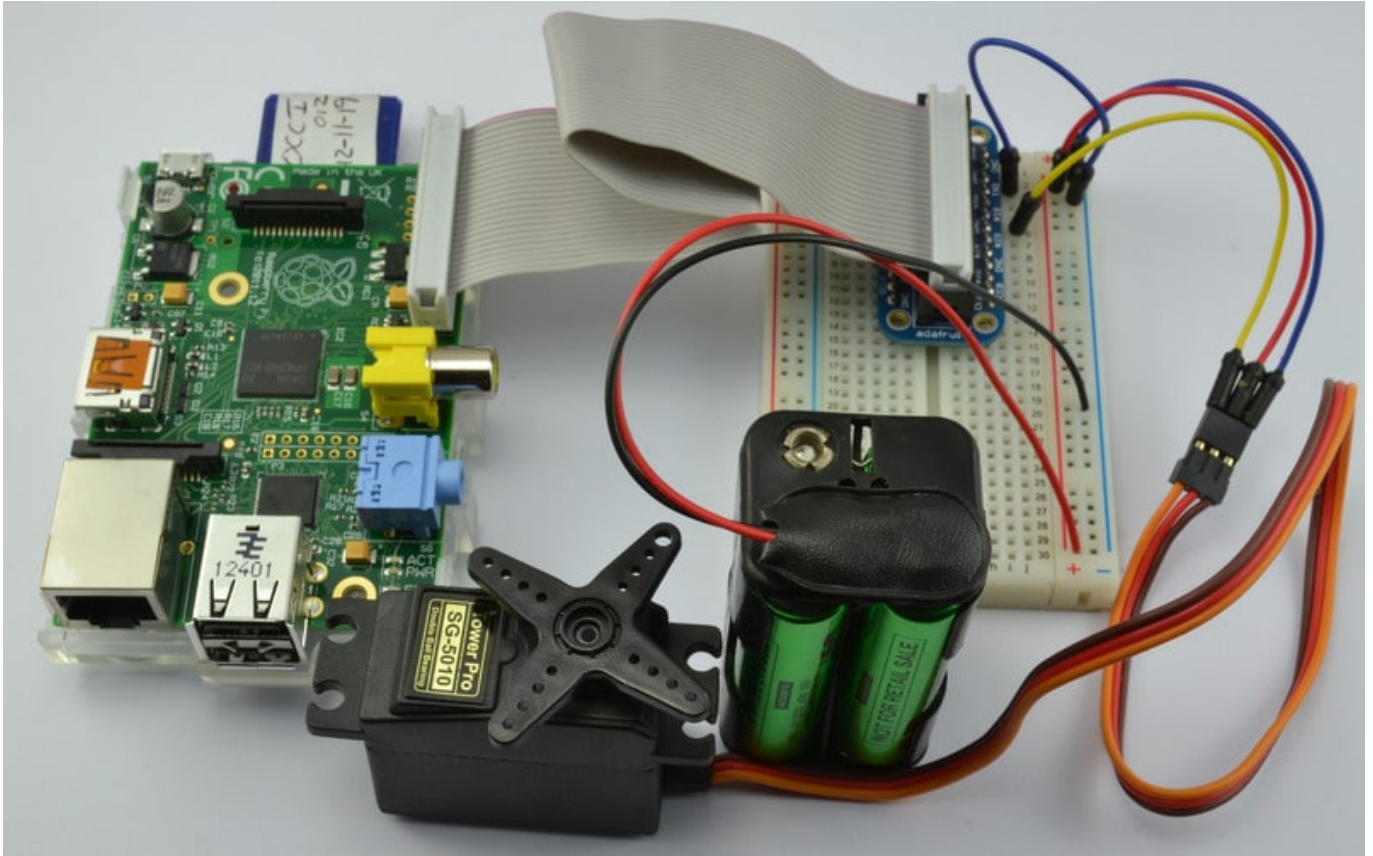


راسبيري باي - الدرس الثامن - التحكم بمحرك السيرفو

في هذا الدرس سوف نشرح كيف نستخدم محرك سيرفو والتحكم به بالبايثون (Python).

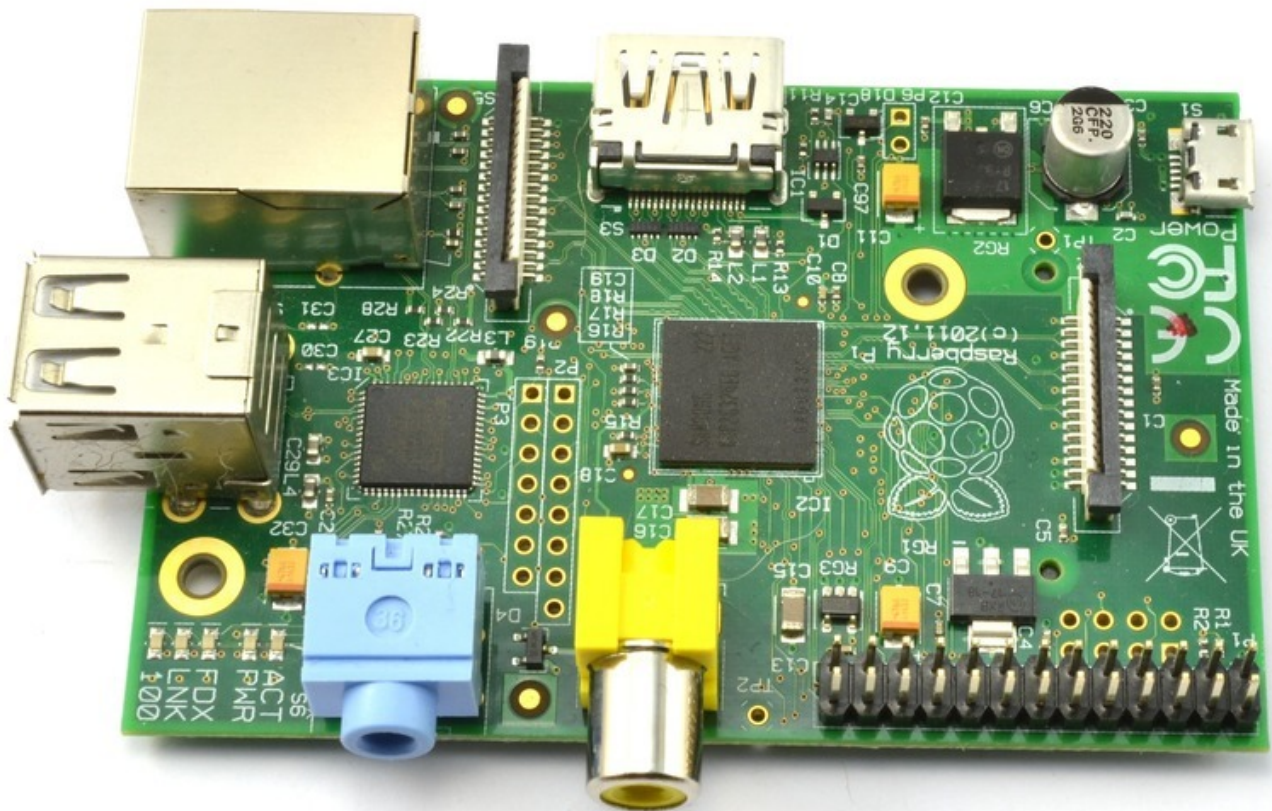


يتم التحكم بمحرك السيرفو عن طريق النبضات الكهربائية من خلال التحكم بطول النبضة في الدورة ، وهذا يتطلب وقت دقيق إلى حدٍ ما ، الازبيري باي تحتوي على أرجل تولد نبضات مباشرة من القطعة نفسها. (راجع الدرس الرابع)

القطع

لتطبيق الدرس تحتاج للقطع التالية:

راسبيري باي



وصلة خارجية لتوصيل أرجل GPIO في لوحة الاختبار



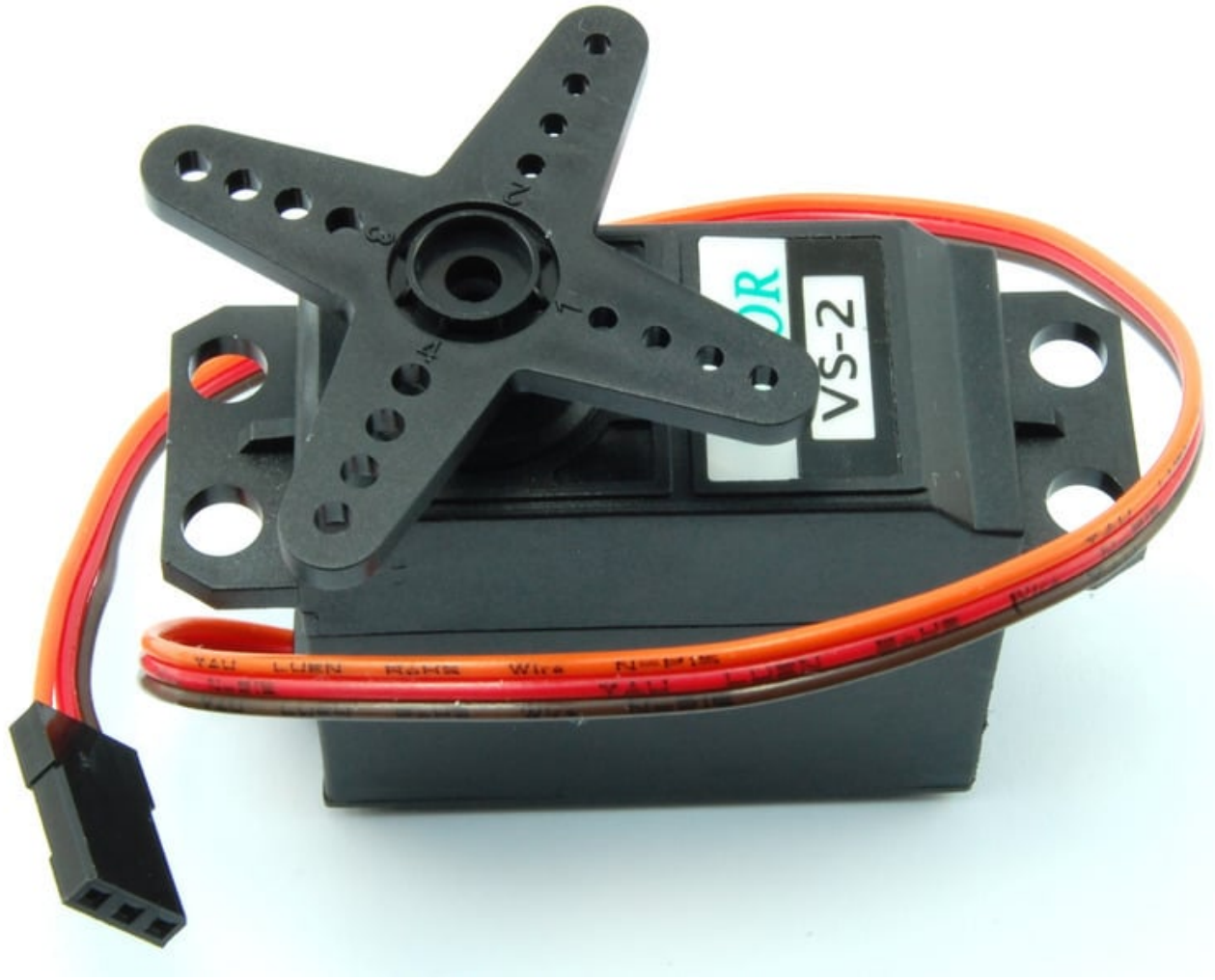
أسلاك توصيل



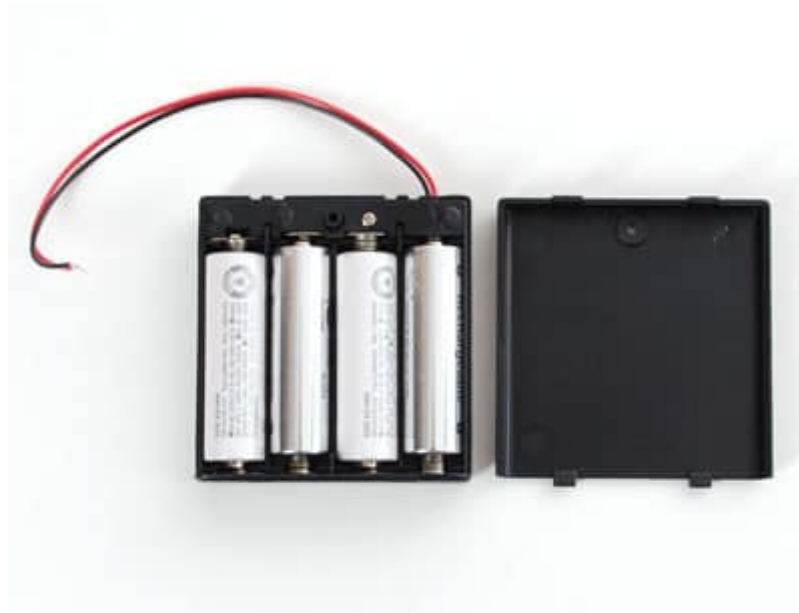
لوحة الاختبار



محرك سيرفو



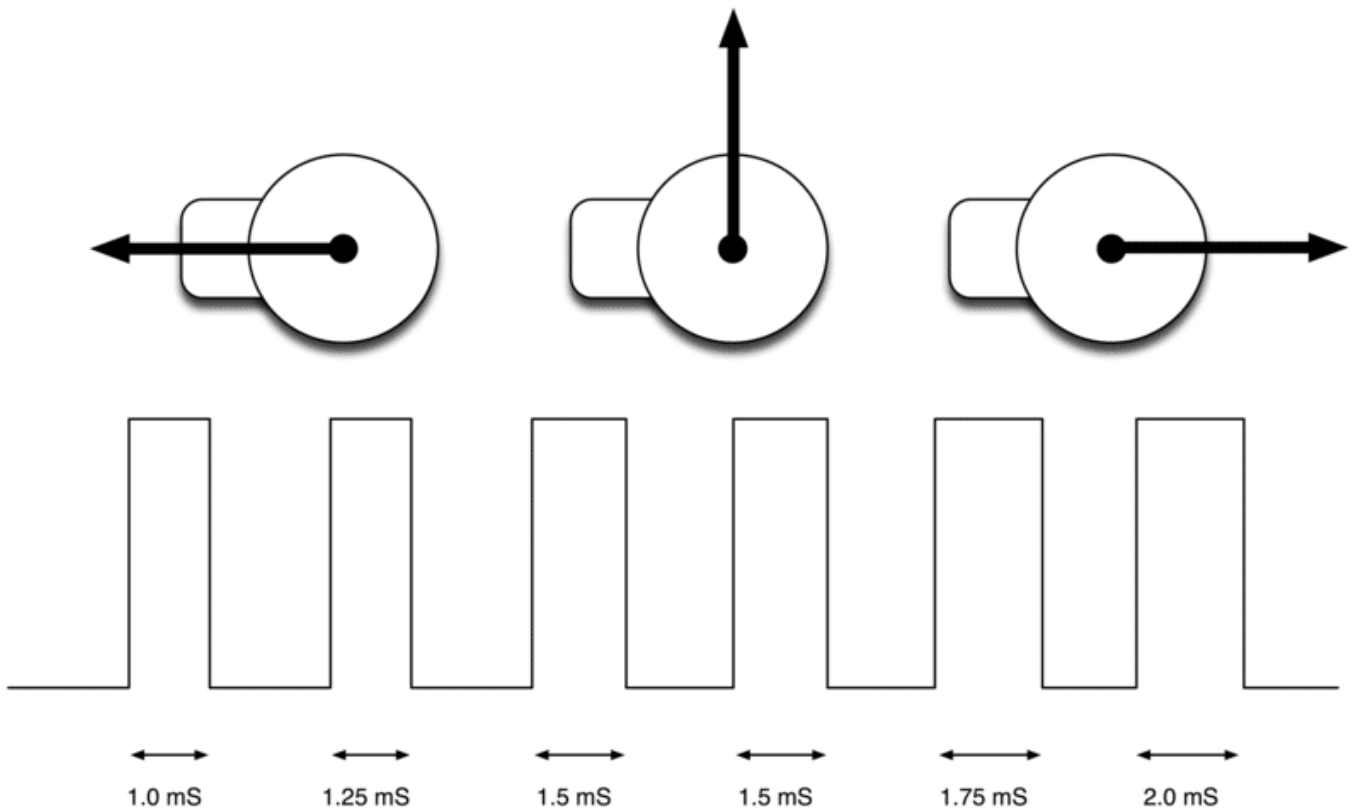
علبة بطاريات + 4 بطاريات AA



محركات السيرفو

التحكم باتجاه حركة محرك السيرفو يكون عن طريق التحكم بطول النبضة أو ما يعرف بد (Duty Cycle) في PWM ، المحرك يتوقع

أن يستقبل نبضة كل 20 ملي ثانية أو كل 50Hz حيث إذا كان طول النبضة 1.5 ملي ثانية فإن المحرك سيتجه للمركز وزاويته 90 درجة ، وإذا قل عن 1.5 ملي ثانية سيتجه للزاوية 0 درجة ، وإذا زاد عن 1.5 سوف يتجه إلى الزاوية 180 درجة.



التحكم بالمحرك باستخدام مكتبة RPi.GPIO

أول خطوة هي تحديد الرجل التي ستستخدم مع PWM وذلك من خلال الدالة PWM كالتالي:

```
p = GPIO.PWM(18, 50)
```

في الحالة السابقة فإن رقم الرجل المراد تفعيل PWM هي 18 وترددها 50Hz والتي تعادل 20 ملي ثانية

لتشغيل أو البدء في إرسال النبضات نستخدم الدالة start:

```
p.start(dc)
```

dc هو قيمة (duty cycle) وهي عبارة عن نسبة مئوية وتحسب كالتالي:

$$dc = L/P$$

حيث L تساوي طول النبضة و P تساوي طول الدورة.

لتغيير قيمة dc نستخدم الدالة ChangeDutyCycle:

```
p.ChangeDutyCycle(dc)
```

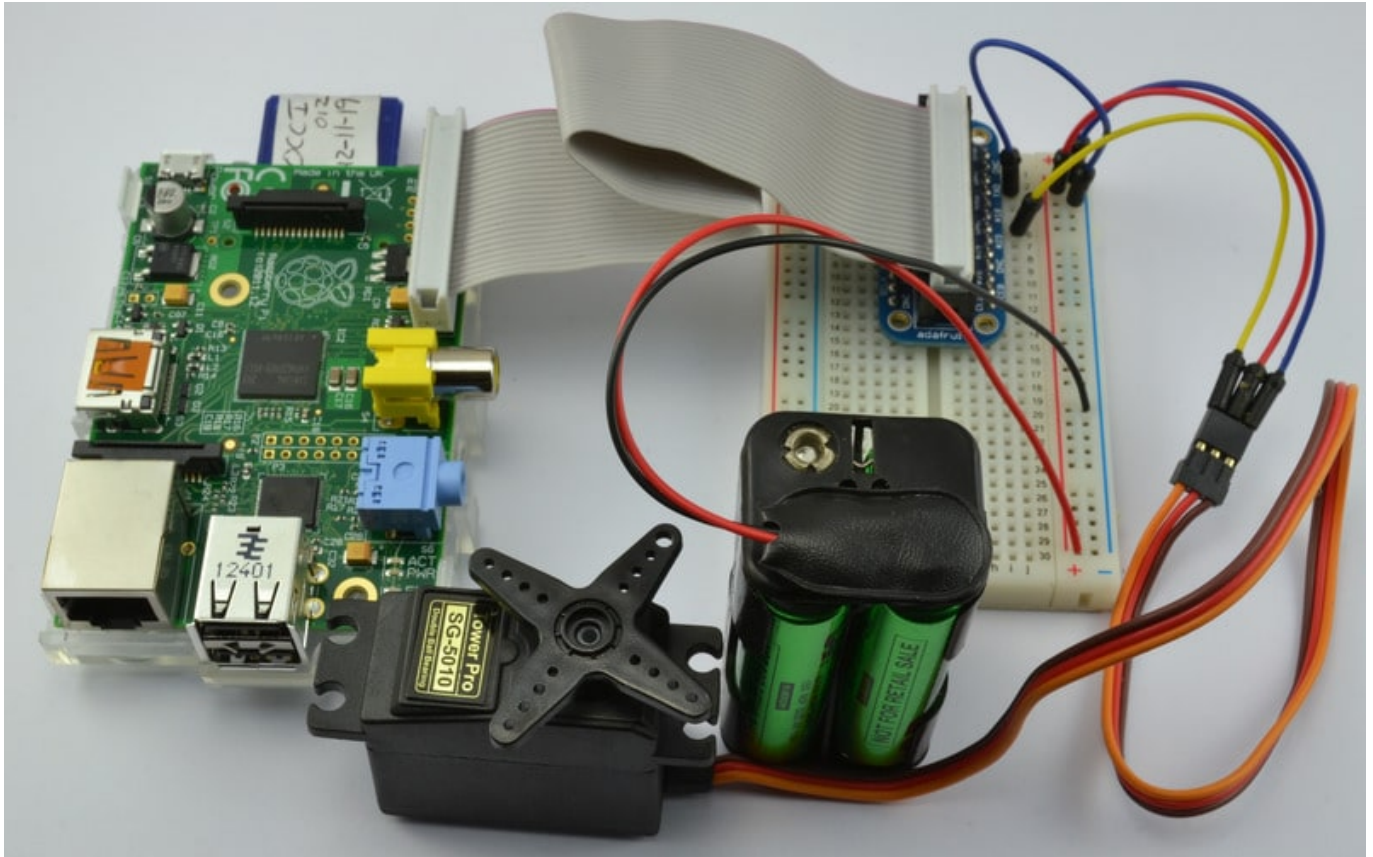
```
p.stop()
```

أولا التوصيل:

السلك الأصفر متصل مع الرجل رقم (GPIO 18)

والسلك الأحمر من البطارية متصل مع السلك الأحمر من المحرك

6/8



ثانيًا الكود البرمجي:

البرنامج التالي سيتحكم بالمحرك من خلال تحريكه للزاوية 90 درجة و ينتظر ثانية واحدة ، ثم يحركه إلى الزاوية 0 درجة و ينتظر ثانية واحدة ، وفي الأخير يحركه إلى الزاوية 180 درجة ، ثم يستمر بتكرار نفس الخطوات حتى تتم مقاطعته.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)

GPIO.setup(18, GPIO.OUT)

p = GPIO.PWM(18, 50)

p.start(7.5)

try:

    while True:

        p.ChangeDutyCycle(7.5)          # turn towards 90 degree
        time.sleep(1)                  # sleep 1 second
        p.ChangeDutyCycle(2.5)          # turn towards 0 degree
        time.sleep(1)                  # sleep 1 second
        p.ChangeDutyCycle(12.5)         # turn towards 180 degree
        time.sleep(1)                  # sleep 1 second

except KeyboardInterrupt:
```

```
p.stop()  
GPIO.cleanup()
```

ملاحظة: dc تم حسابها كالتالي:

$$dc = (0.5/20) \times 100 = 2.5\%$$

$$dc = (1.5/20) \times 100 = 7.5\%$$

$$dc = (2.5/20) \times 100 = 12.5\%$$