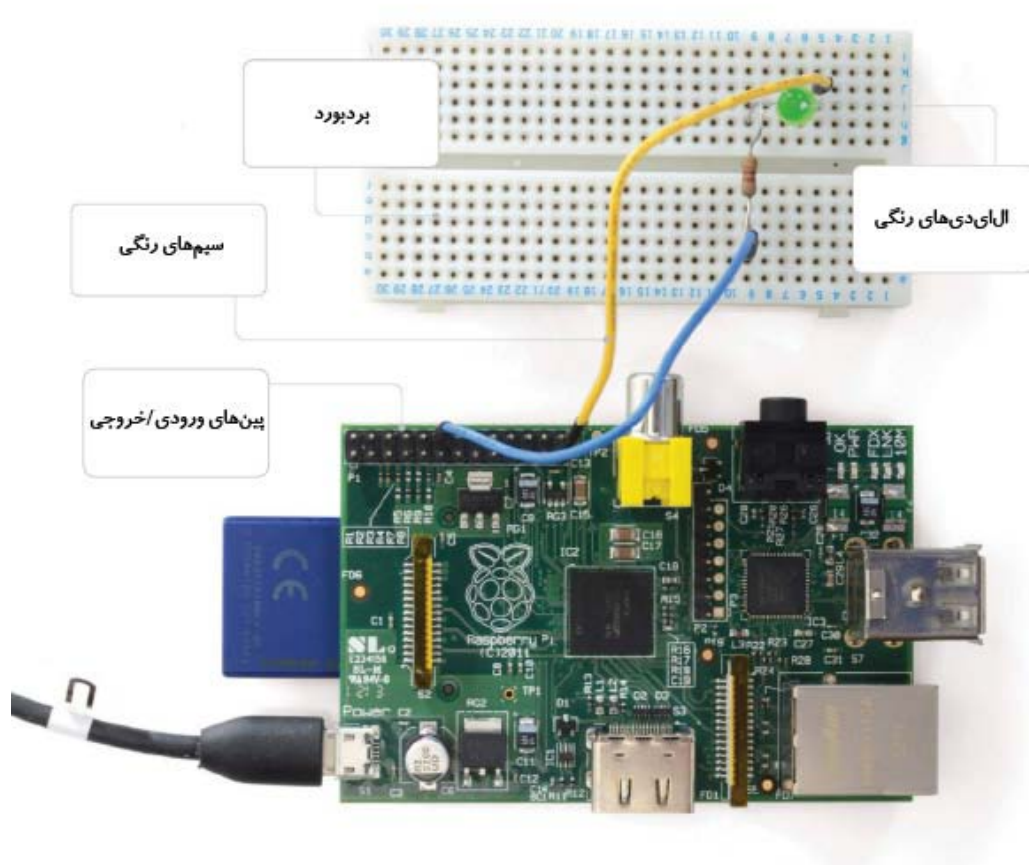
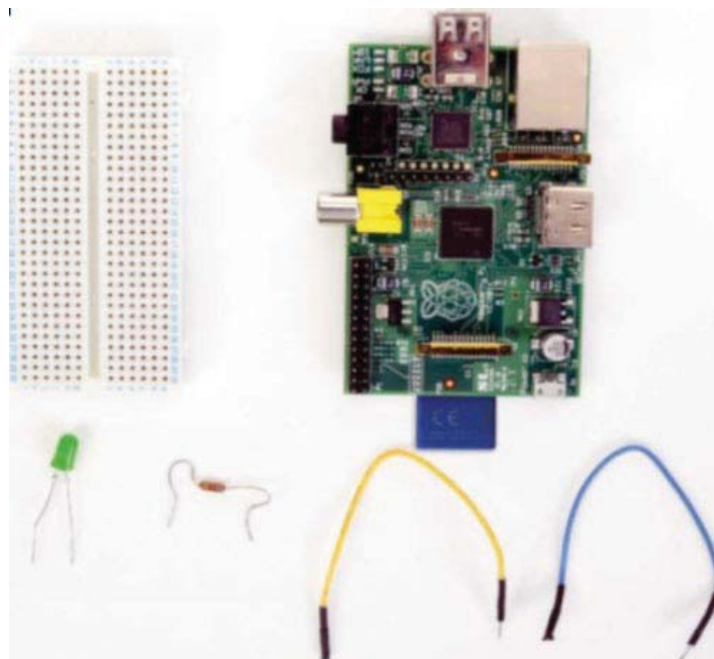


ساخت led چشمک زن با رزبری پای (۱۹)

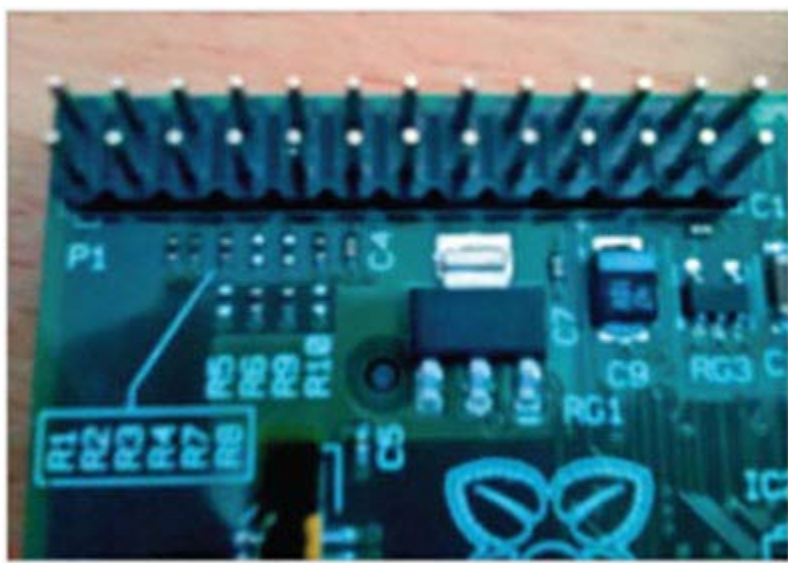
یکی از قابلیت‌های Pi کاربرد آن در پروژه‌های الکترونیکی است. شاید قبلاً از Arduino به‌عنوان یک میکروکنترلر استفاده کرده باشید Pi. به دلیل پردازنده و حافظه توسعه‌یافته قابلیت‌های بیش‌تری نسبت به Arduino فراهم ساخته است.



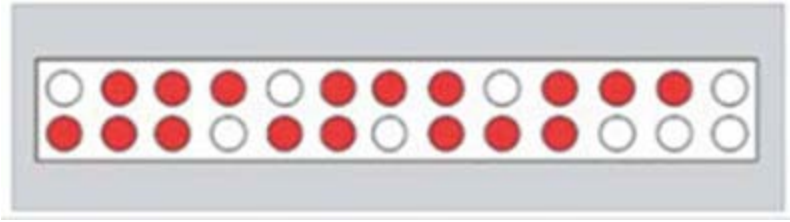
Raspberry Pi به‌علت داشتن یک (PWM سرنام (Pulse Width Modulation پهن خروجی و تعدادی پهن‌های ورودی/خروجی همه‌منظوره، می‌تواند در کنترل سخت‌افزارهای الکترونیکی مانند زنگ اخبار، چراغ‌ها و کلیدها مورد استفاده قرار گیرد. با استفاده از مراحل که در ادامه شرح داده می‌شود، قادر خواهید بود ال‌ای‌دی‌ها را با استفاده از Pi کنترل کنید. قبل از این‌که کدنویسی را شروع کنید، به ساختار و نحوه قرار گرفتن پین‌ها روی Pi نگاهی بیاندازید. بسته به این‌که چه زمانی Pi را خریداری کرده‌اید، ممکن است از نسخه ۱ یا ۲ استفاده کنید. نحوه قرار گرفتن پهن‌های ورودی/خروجی همه‌منظوره در این دو نسخه کمی متفاوت است. در این‌جا با نسخه ۱ کار خواهیم کرد.



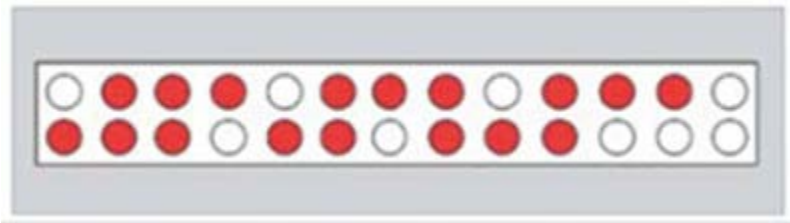
اگر به گوشه سمت چپ بالای تابلو نگاه کنید، یک برجسب سفید رنگ به نام 'P1' مشاهده می‌کنید. پین ۱ در این قسمت قرار دارد و بالای آن نیز پین ۲ است. این الگو ادامه دارد تا به پین ۲۶ برسید.



پین ۱ ولتاژ ۳,۳ ولت دارد و پین اصلی برای تأمین برق ال‌ای‌دی است. پین‌های ۲ و ۴ ولتاژ ۵ ولت و پین‌های ۶، ۹، ۱۴، ۲۰ و ۲۵ پین‌های اتصال به زمین دارند.



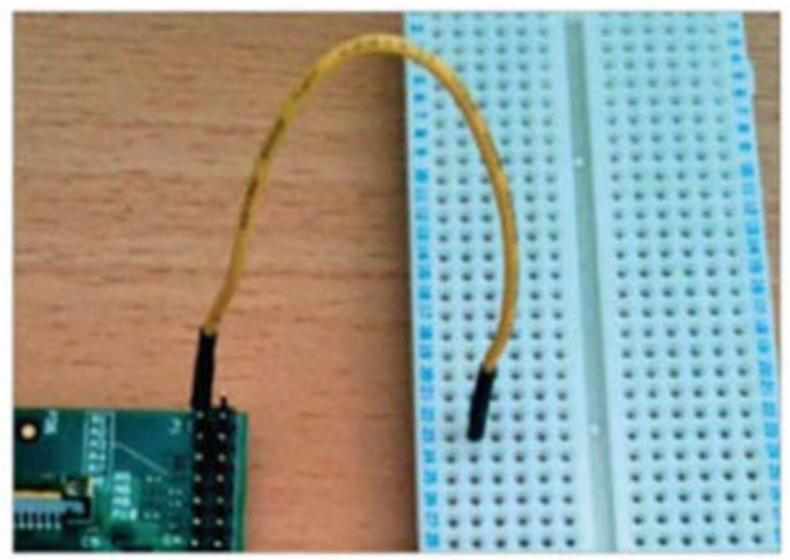
پین ۱ ولتاژ ۳,۳ ولت دارد و پین اصلی برای تأمین برق ال‌ای‌دی است. پین‌های ۲ و ۴ ولتاژ ۵ ولت و پین‌های ۶، ۹، ۱۴، ۲۰ و ۲۵ پین‌های اتصال به زمین دارند.



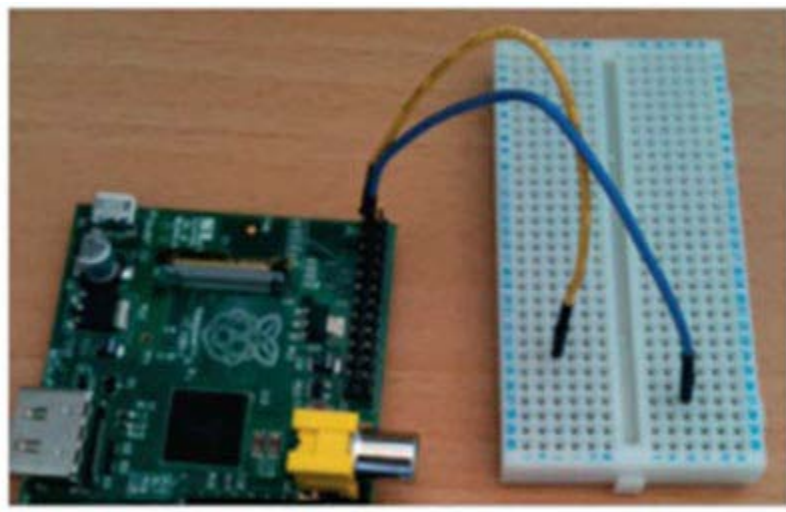
حال بردبرد، دو سیم، یک مقاومت ۲۷۰ اهم و یک ال‌ای‌دی بردارید. اگر دقت کنید، یکی از پایه‌های ال‌ای‌دی اندکی خمیدگی دارد. این پایه را در نظر داشته باشید؛ زیرا در مراحل بعدی مهم است. مطمئن شوید که Pi به برق وصل نباشد.



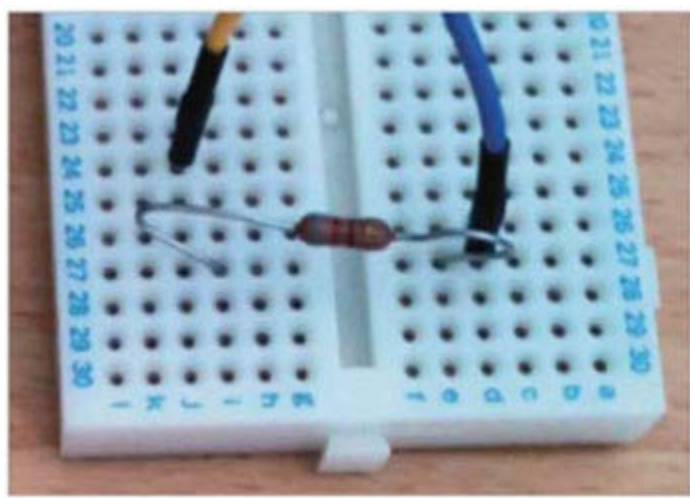
یک پایه سیم را به پین شماره ۱ روی Pi و پایه دیگر را به بردبرد وصل کنید. مهم نیست سیم را به کدام قسمت از بردبرد وصل کنید، فقط دقت کنید جایی که وصل کردید در اطراف آن شکاف‌های کافی برای ال‌ای‌دی‌ها وجود داشته باشد.



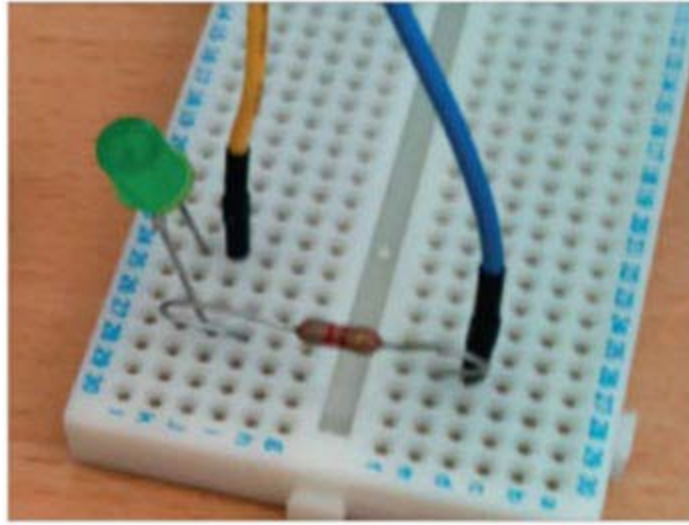
سر ماده سیم را به پین ۶ و سر دیگر را به بردبرد وصل کنید. دقت کنید که باید فضای کافی برای مقاومت خود در نظر بگیرید. نکته مهمی که باید به آن توجه کنید این است که تحت هیچ شرایطی مقدار مقاومتی که انتخاب می‌کنید، کمتر از مقدار مقاومت ال‌ای‌دی نباشد.



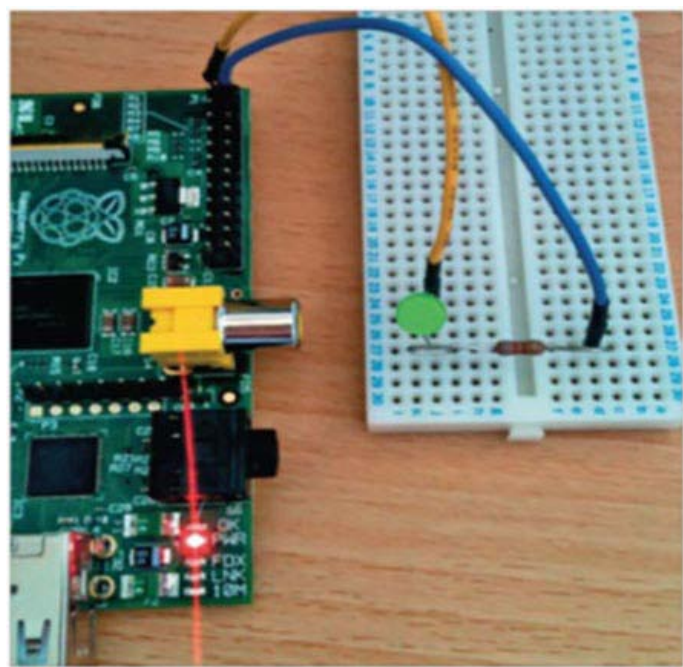
حال یک سر مقاومت را نزدیک سیم متصل شده به زمین و سر دیگر را به یک شکاف زیر سیم ۳,۳ ولت وصل کنید.



ال‌ای‌دی را بردارید. پایه خم شده را کنار سیم ۳/۳ ولت و پایه دیگر را کنار مقاومت (روبروی سیم متصل به زمین) وصل کنید.



USB را به برق، لپتاپ یا کامپیوتر وصل کنید تا روشن شود. به محض وصل شدن به برق، ال ای دی باید روشن شود.



این ال ای دی را از طریق کدنویسی کنترل کنید. پیش از آغاز، به روز بودن همه برنامه‌ها را با استفاده از دستور `sudo apt-get dist-upgrade` چک کنید. برای کنترل ال ای دی باید به ریشه دسترسی داشته باشید، پس در پایانه دستور `su` را وارد کنید. اگر نمی‌خواهید از این دستور استفاده کنید، باید به‌خاطر داشته باشید که قبل از نوشتن هر دستوری کلمه `'sudo'` را اضافه کنید. یک کتابخانه بسیار مفید به‌نام `GPIO پایتون` وجود دارد که کار با پین‌های `GPIO` را بسیار راحت می‌کند. با استفاده از دستور زیر، این کتابخانه را دریافت کنید و فایل‌های آن را از حالت فشرده خارج سازید.

```
wget https://pypi.python.org/packages  
source/R/RPi.GPIO/RPi.GPIO-0.5.2a.tar.gz
```

```
tar xzf Rpi.GPIO-0.5.2a.tar.gz
cd Rpi.GPIO-0.5.2a
```

با استفاده از دستور زیر کتابخانه را دانلود کنید.

```
sudo apt-get install python-dev
sudo python setup.py install
```

ابتدا یک قطعه کد پایتون بنویسید و کتابخانه را در آن وارد کنید و درون یک بلاک try-except قرار دهید تا مطمئن شوید همه چیز به درستی انجام می‌شود. برای اطمینان از درست بودن کد، دستور sudo python gpio.py را وارد کنید.

```
cd /
cd Desktop
sudo nano gpio.py
try:
    import RPi.GPIO as GPIO
except RuntimeError:
    print("Error importing GPIO lib")
```

gpio.py نام کدی است که نوشتید. دوباره کد را در Nano باز و حالت ورودی/خروجی به تابلو را تنظیم کنید.

```
sudo nano gpio.py
GPIO.setmode(GPIO.BOARD)
```

برای این که یک پین به درستی کار خود را انجام دهد، باید مشخص شود که از نوع ورودی است یا خروجی. این کار در کتابخانه GPIO از طریق فراخوانی تابع GPIO.setup انجام می‌شود. شماره پین را به این تابع ارسال و یکی از دو حالت GPIO.OUT یا GPIO.IN را برای آن تعریف کنید. به دلیل این که قرار است در این پروژه از ال‌ای‌دی استفاده کنید، پین از نوع خروجی است.

```
GPIO.setup(12, GPIO.OUT)
```

قدم بعدی این است که مشخص کنید پین یک خروجی تولید کند و سپس راهی برای خارج شدن از برنامه پیدا کند. در اینجا باز هم از کلاس GPIO استفاده و تابع PWM را فراخوانی کنید. این تابع دو پارامتر ورودی دارد. یکی شماره پین و دیگری مقدار فرکانس به هرتز که در اینجا ۰,۵ است.

```

p = GPIO.PWM(12, 0.5)
p.start(1)
input('Press return to stop:')
p.stop()
GPIO.cleanup()

```

برای اینکه به ال‌ای‌دی یک تایمر اضافه کنید تا به‌تدریج محو شود، باید کتابخانه time را به برنامه اضافه و پین ۱۲ را تنظیم کنید که فرکانس ۵۰ هرتز دارد.

```

import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.OUT)
p = GPIO.PWM(12, 50) # channel=12
frequency=50Hz
p.start(0)

```

حال با استفاده از یک بلاک try-except، میزان برق ال‌ای‌دی را اندازه بگیرید و اگر به یک سطح مشخص رسید، فرایند معکوس شود. برای این کار ابتدا برنامه را ذخیره کنید و ببندید و مجدد آن را باز کنید و کد زیر را بنویسید.

```

while 1:
    for dc in range(0, 101, 5):
        p.ChangeDutyCycle(dc)
        time.sleep(0.1)
    for dc in range(100, -1, -5):
        p.ChangeDutyCycle(dc)
        time.sleep(0.1)
except KeyboardInterrupt:
    pass
p.stop()
GPIO.cleanup()

```

منبع: [مجله شبکه مترجم: سارا بصیری](#)