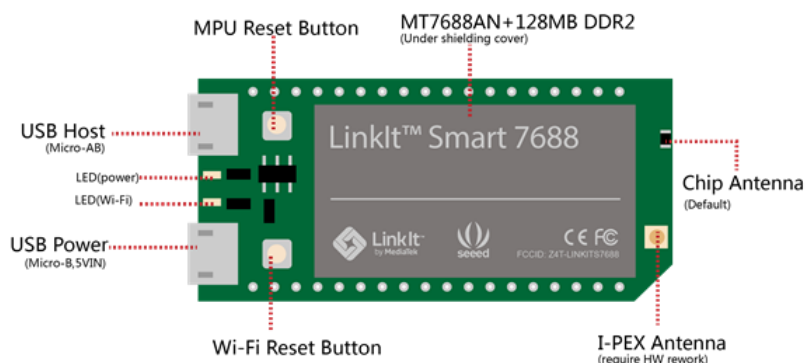


Інструкції до практичних робіт з модулем Linkit Smart 7688



Зміст

Встановлення необхідних програм	2
Робота 1. Перше підключення до модуля Linkit Smart 7688 через WiFi мережу	3
Робота 2. Написання перших програм на Python. Блумаємо світлодіодом	12
Робота 3. Керуємо яскравістю світлодіода за допомогою PWM	20
Робота 4. Підключаємо кнопку	23
Робота 5. Використовуємо переривання	26
Робота 6. Реалізуємо програмно рухомі вогники	29
Робота 7. Підключаємо електронний компас з інтерфейсом I2C	34
Підключення до WiFi роутеру та мережі Internet	40
Переведення модуля в режим точки доступу	43
Встановлення нового імені (dns адреси) модуля	44
Бонусна робота. Створення веб серверу на Python	47
Визначення IP адреси модуля	51
Оновлення прошивки модуля	52
Відновлення стандартних налаштувань модуля	55
Індикація режимів роботи модуля	56
Опис ліній введення-виведення модуля	57
Посилання	59

Встановлення необхідних програм

1. Для доступу до модуля Linkit Smart 7688 через WiFi мережу з операційної системи Windows встановіть **Bonjour print service** (для Linux і Mac OS X цього не потрібно робити):

https://support.apple.com/kb/DL999?locale=en_US

2. Для підключення до модуля Linkit Smart 7688 з ОС Windows через зашифроване SSH з'єднання встановіть програму Putty (для Linux і Mac OS X цього не потрібно робити):

- для 32-х Windows:

<https://the.earth.li/~sgtatham/putty/latest/w32/putty-0.68-installer.msi>

- для 64-х Windows:

<https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.68-installer.msi>

3. Для завантаження файлів на модуль по протоколу SCP через захищене SSH з'єднання з ОС Windows встановіть WinSCP (для Linux і Mac OS X цього не потрібно робити):

- детальний опис програми: <https://winscp.net/eng/docs/lang:uk>
- сторінка завантаження: <https://winscp.net/eng/download.php>
- пряме посилання на файл для встановлення:

<https://winscp.net/download/WinSCP-5.9.4-Setup.exe>

Робота 1.

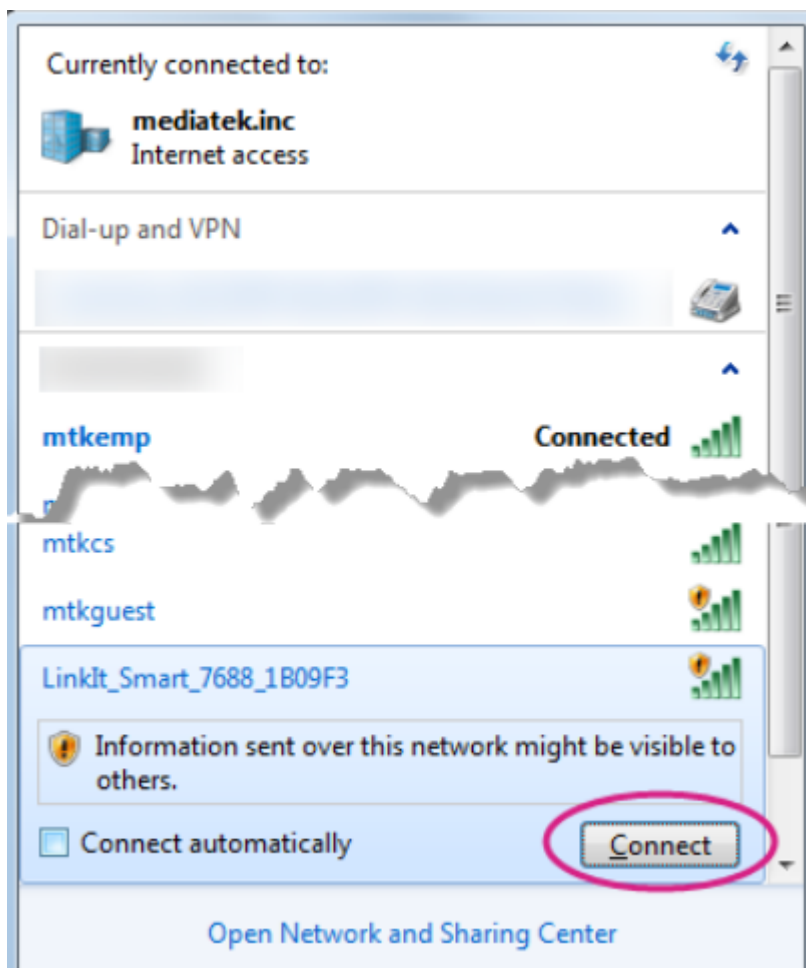
Перше підключення до модуля Linkit Smart 7688 через WiFi мережу

1. **Подайте напругу живлення** на модуль Linkit Smart 7688 підключивши його до USB порта комп'ютера. Вставте один кінець кабелю в USB порт комп'ютера, а інший кінець кабелю підключіть до micro USB порта модуля Linkit Smart 7688, біля якого написано PWR. В результаті повинен засвітитися зелений світлодіод, а червоний світлодіод спершу світлитиметься приблизно 30 секунд (протягом завантаження модуля), а потім почне блимати (після встановлення WiFi з'єднання).



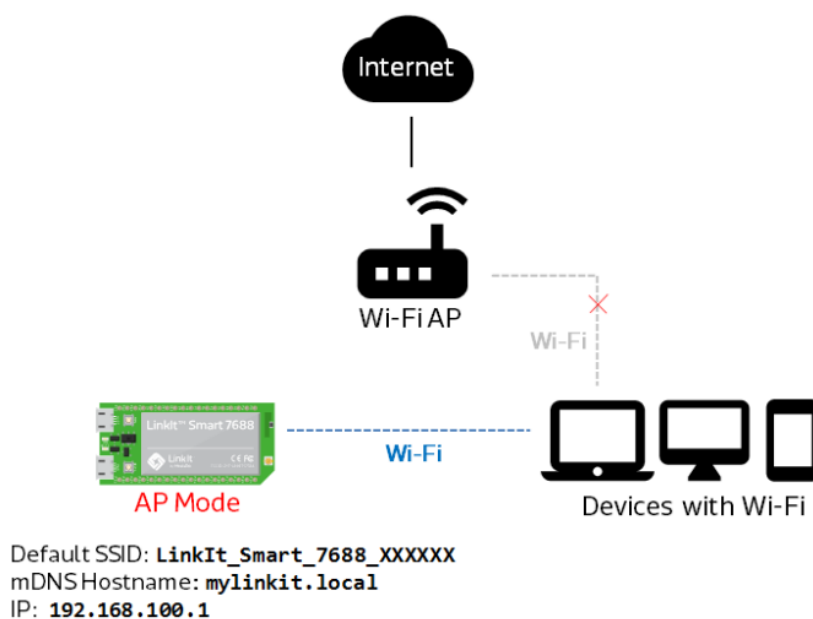
2. Підключіться до модуля Linkit Smart 7688 по WiFi

Після того, як модуль завантажився, він починає працювати, як WiFi точка доступу з іменем **LinkIt_Smart_7688_XXXXXX**. Підключіться до цієї точки доступу зі свого ноутбуку.



Після того, як ви підключитеся до модуля Linkit Smart 7688 в такому режимі, його червоний світлодіод почне блимати з частотою 3 рази на секунду.

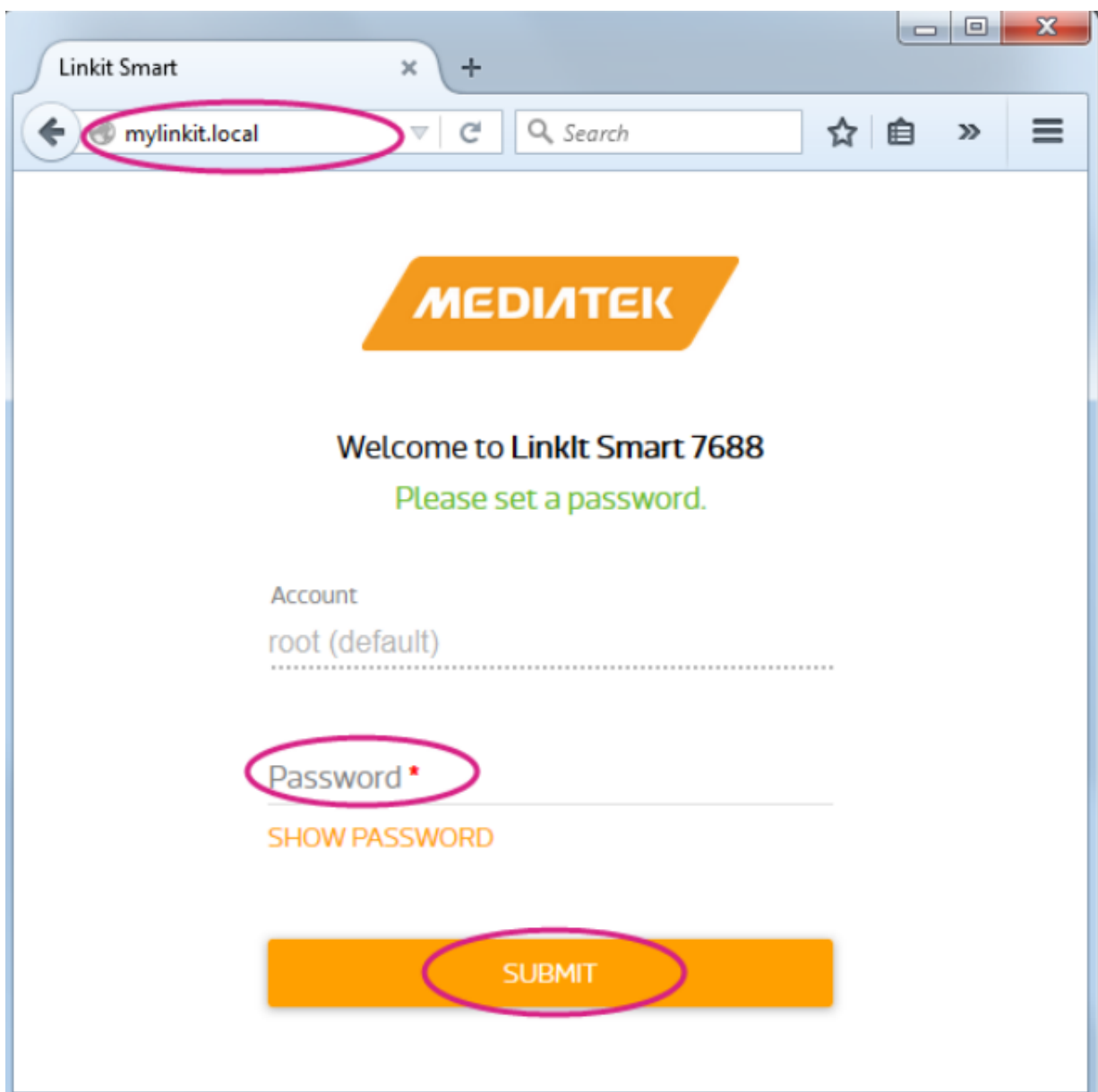
Зверніть увагу, що після того, як ви підключитеся до WiFi модуля Linkit Smart 7688 в такому режимі, доступ до Інтернету у вас може бути лише через провідне з'єднання.



Якщо ви мали доступ до Internet через WiFi канал, тепер цей зв'язок буде втрачено, оскільки ваша WiFi карта підключиться не до WiFi роутера, а до модуля Linkit Smart 7688. Цю проблему можна вирішити перевіривши модуль Linkit Smart 7688 в режим WiFi станції, що підключається до того ж WiFi роутера, що і ваш ноутбук. Як це зробити буде показано в одному з наступних розділів.

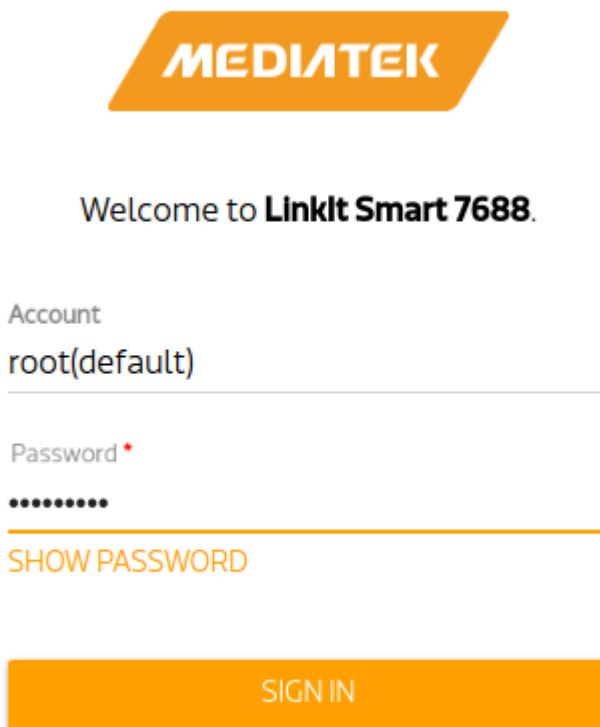
3. Зайдіть у веб інтерфейс керування модулем Linkit Smart 7688

Після підключення по WiFi до Linkit Smart 7688, відкрийте браузер і відкрийте у ньому веб інтерфейс керування модулем за адресою <http://mylinkit.local>



Якщо ви робите цей перший раз, вам необхідно ввести пароль, який ви будете надалі використовувати для підключення до модуля і натиснути **SUBMIT**.

Після цього з'явиться стандартне вікно контролю доступу до веб інтерфейсу керування модулем де вам запропонують ввести щойно заданий пароль.



MEDIATEK

Welcome to **Linkit Smart 7688**.

Account
root(default)

Password *
.....

SHOW PASSWORD

SIGN IN

Після введення паролю необхідно натиснути кнопку входу **SIGN IN**.

Якщо ви забули пароль, ви не зможете підключитися до модуля. Єдина можливість - це зкинути пароль разом з усіма налаштуваннями модуля. Як це зробити показано у одному з наступних розділів. Однак варто мати на увазі, що під час процедури зкидання паролю будуть видалені всі файли, створені вами, чи завантажені у модуль, а також всі встановлені програми. Тому краще добре запам'ятати пароль.

Після введення правильного паролю і натиснення на кнопку **SIGN IN** ви потрапляєте у веб інтерфейс спрощених налаштувань модуля. Тут ви можете переглянути ім'я та IP адресу модуля, задати новий пароль натиснувши кнопку **CONFIGURE**, оновити прошивку (як це зробити описано в одному з наступних розділів), також зробити зкидання всіх налаштувань прошивки до стандартних значень натиснувши кнопку **RESET**.

System information	Network
<h3>Platform information</h3> <p>Device name mylinkit</p> <hr/> <p>Current IP address 192.168.100.1</p> <hr/> <h3>Account information</h3> <p>Account root(default)</p> <hr/> <p>Password *</p> <hr/> <p>CONFIGURE</p>	
<h3>Software information</h3> <p>Boot loader version v0.8.2</p> <hr/> <p>Firmware version v0.9.4</p> <hr/> <p>UPGRADE FIRMWARE</p>	
<h3>Factory reset</h3> <p>Reset the device to its default settings. Important: This action will remove all your data and settings from the device.</p> <hr/> <p>RESET</p>	

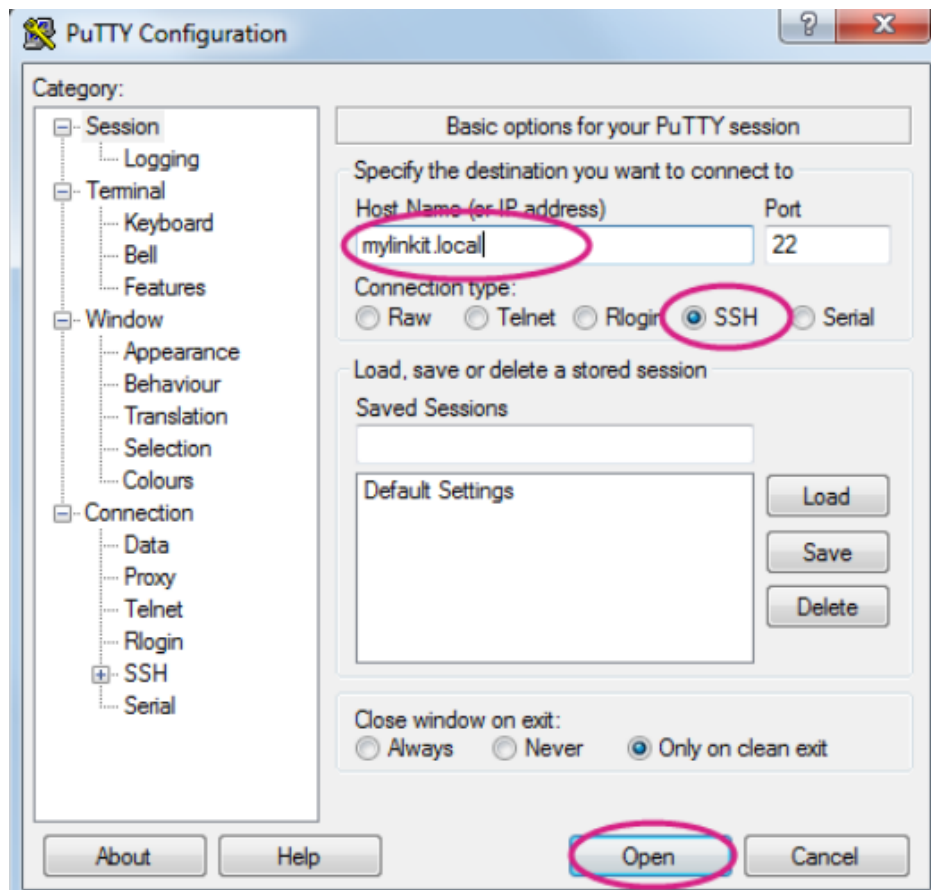
Для переходу в режим WiFi станції та підключення до WiFi роутеру необхідно натиснути кнопку **Network** у правому верхньому кутку. Детальніше про те, як підключити модуль Linkit Smart 7688 до WiFi роутеру написано в одному з наступних розділів.

Для переходу в режим розширених налаштувань необхідно натиснути посилання **OpenWrt** у верхньому правому куті. Однак поки не треба цього робити.

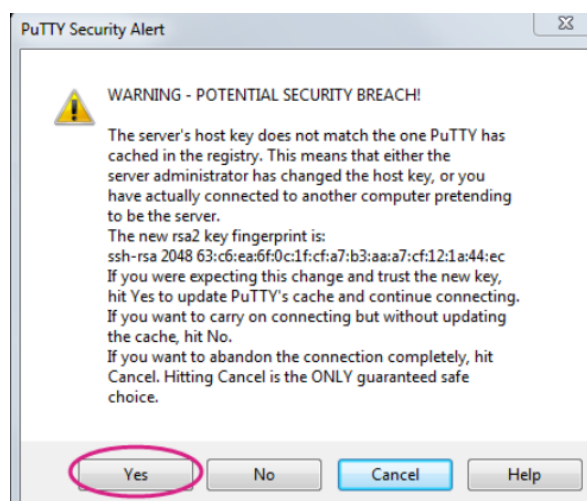
4. Підключення до терміналу модуля Linkit Smart 7688 через SSH

Інструкція для Windows

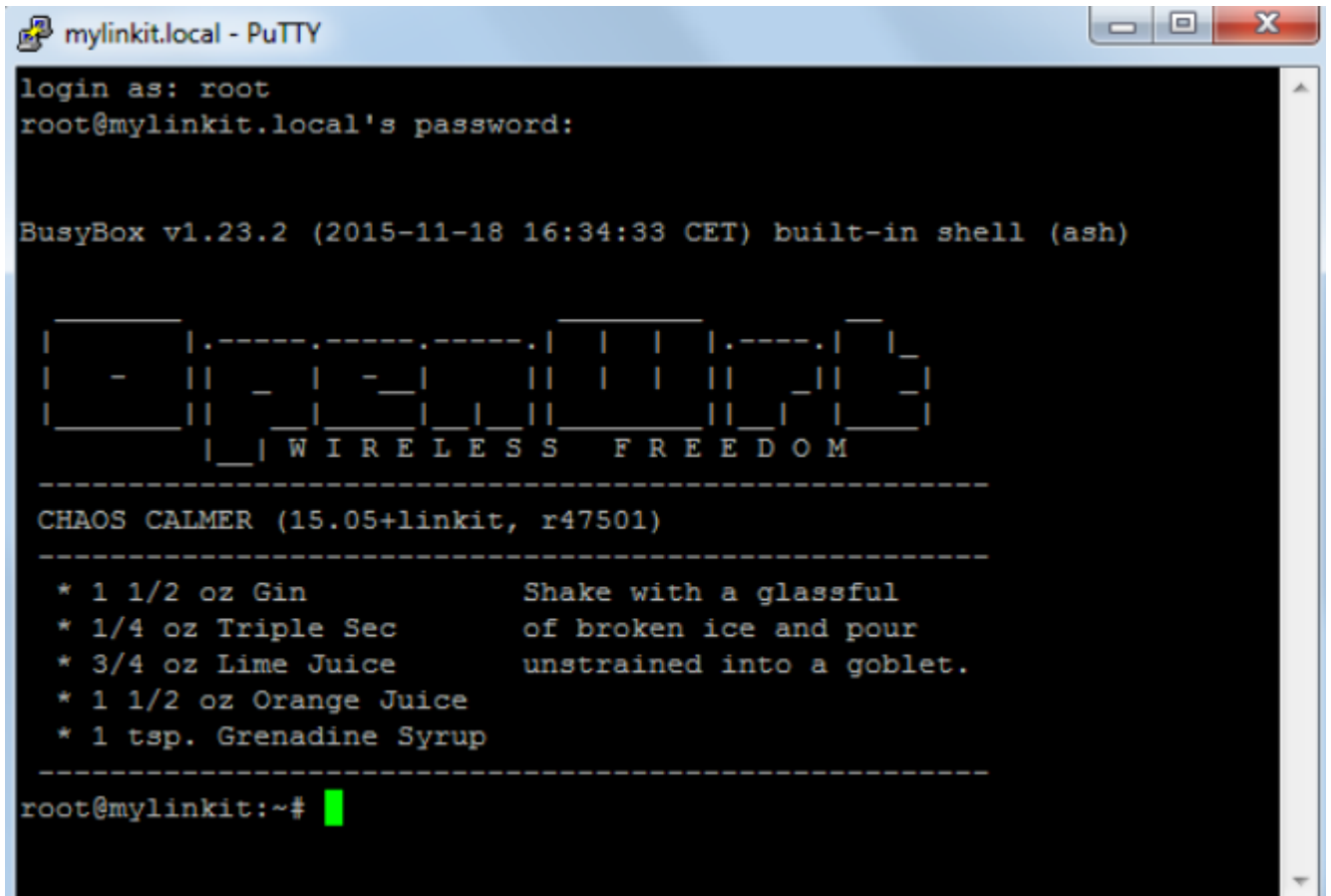
Запустіть програму PuTTY, яку ви встановили раніше. В полі Host Name напишіть адресу модуля mylinkit.local (Або IP адресу модуля. Як визначити IP адресу Linkit Smart 7688 розказано в одному з наступних розділів). Оберіть перемикач SSH та натисніть на кнопку **Open** для підключення до модуля.



При першій спробі підключення з'явиться інформаційне вікно з попередженням. Просто натисніть **YES**.



Відкриється вікно терміналу



```
mylinkit.local - PuTTY
login as: root
root@mylinkit.local's password:

BusyBox v1.23.2 (2015-11-18 16:34:33 CET) built-in shell (ash)

|_| W I R E L E S S   F R E E D O M
-----
CHAOS CALMER (15.05+linkit, r47501)
-----
* 1 1/2 oz Gin           Shake with a glassful
* 1/4 oz Triple Sec     of broken ice and pour
* 3/4 oz Lime Juice    unstrained into a goblet.
* 1 1/2 oz Orange Juice
* 1 tsp. Grenadine Syrup
-----
root@mylinkit:~# █
```

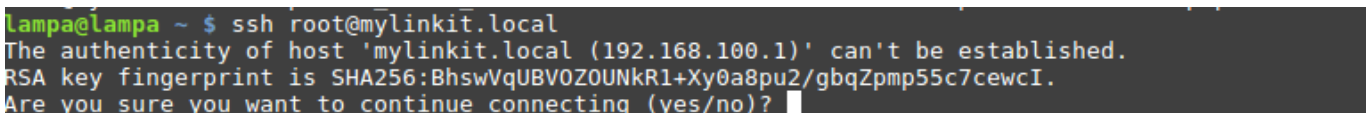
Введіть логін **root** і той пароль, що ви задали у веб інтерфейсі керування модуля.

Вітання! Ви отримали доступ до консолі (терміналу) Linux OpenWrt!

Інструкція для Linux і Mac

Запустіть з терміналу (консолі) команду **ssh root@mylinkit.local**

При першій спробі підключення з'явиться попередження. Просто напишіть **yes** та натисніть Enter.



```
lampa@lampa ~ $ ssh root@mylinkit.local
The authenticity of host 'mylinkit.local (192.168.100.1)' can't be established.
RSA key fingerprint is SHA256:BhswVqUBVOZOUNkR1+Xy0a8pu2/gbqZpmp55c7cewcI.
Are you sure you want to continue connecting (yes/no)? █
```

Вам запропонують ввести пароль **root@mylinkit.local's password: █**

Введіть пароль, який ви задали у веб інтерфейсі керування модуля та натисніть Enter.

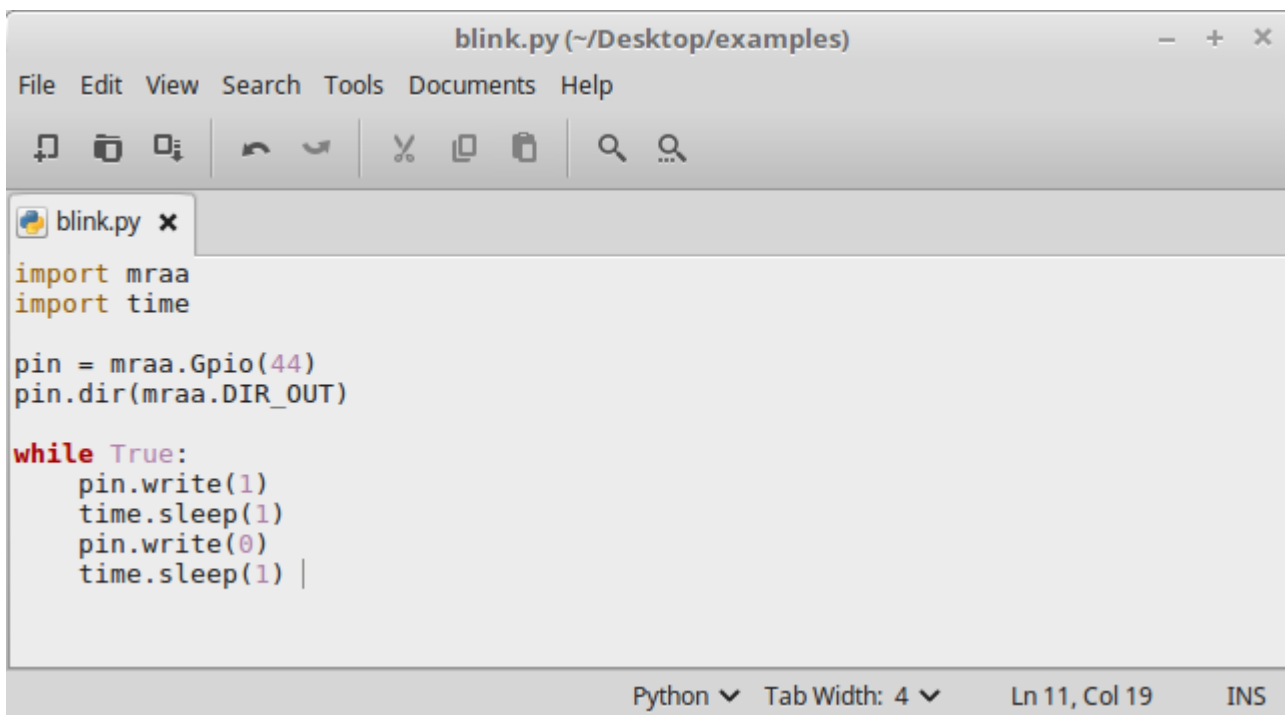
Робота 2.

Написання перших програм на Python. Блимаємо світлодіодом.

Створимо програму для миготіння червоним світлодіодом, що знаходиться на платі модуля Linkit Smart 7688.

В кінці попередньої роботи ви запустили програму, що миготіла червоним світлодіодом модуля Linkit Smart 7688. А зараз спробуйте написати таку програму самостійно.

На своєму комп'ютері створіть текстовий файл з іменем "blink.py". Додайте у файл наступний код на Python:



```
import mraa
import time

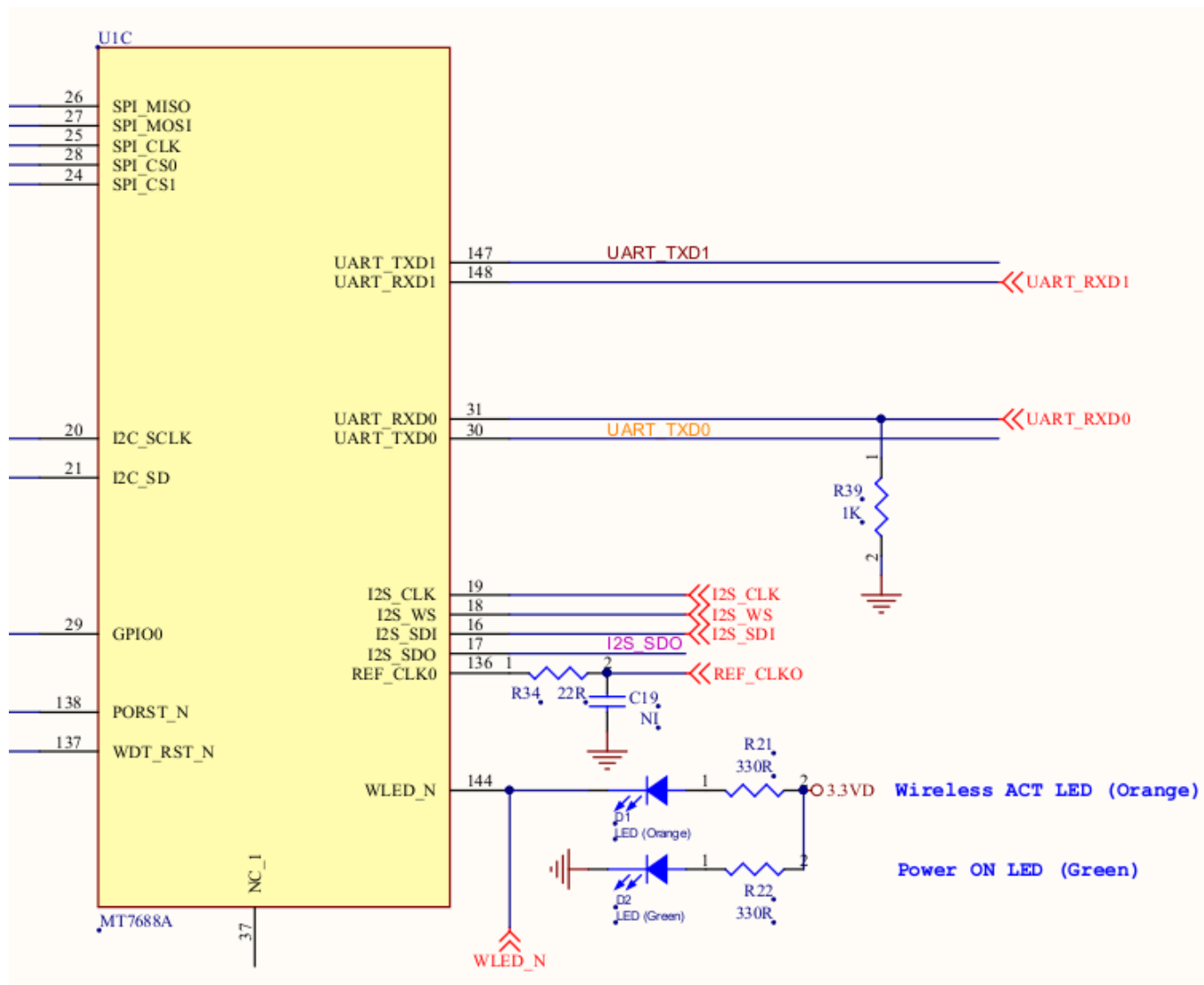
pin = mraa.Gpio(44)
pin.dir(mraa.DIR_OUT)

while True:
    pin.write(1)
    time.sleep(1)
    pin.write(0)
    time.sleep(1) |
```

За допомогою **import mraa** підключаємо бібліотеку mraa (портовану з Intel Edison), яка містить функції для керування залізом модуля з програм на Python. За допомогою функцій цієї бібліотеки будемо керувати напругою на виводі модуля.

За допомогою **import time** підключаємо бібліотеку time, що містить функцію затримки sleep(...), яка зупиняє виконання програми на заданий у секундах проміжок часу.

Створюємо змінну `pin = mraa.Gpio(44)`, яка буде описувати контакт модуля з номером 44. Саме до цього контакту підключено червоний світлодіод за наступною схемою:



WLED_N - це і є контакт №44. Як бачимо зі схеми, щоб засвітити червоний світлодіод необхідно подати на 44-й контакт напругу 0 Вольт. Якщо ж на 44-й контакт подати напругу 3.3 Вольт, світлодіод не світитиметься.

Виклик функції `pin.dir(mraa.DIR_OUT)` налаштовує пін №44 працювати на вихід (тобто формувати певну напругу -- 0 Вольт, або 3.3 Вольт).

Виклик функції `pin.write(1)` приводить до появи на виводі №44 напруги 3.3 Вольт і світлодіод перестає світитися.

Виклик функції `pin.write(0)` приводить до появи на виводі №44 напруги 0 Вольт і світлодіод починає світитися.

Виклик функції **time.sleep(1)** зупиняє виконання програми на 1 секунду (створює затримку).

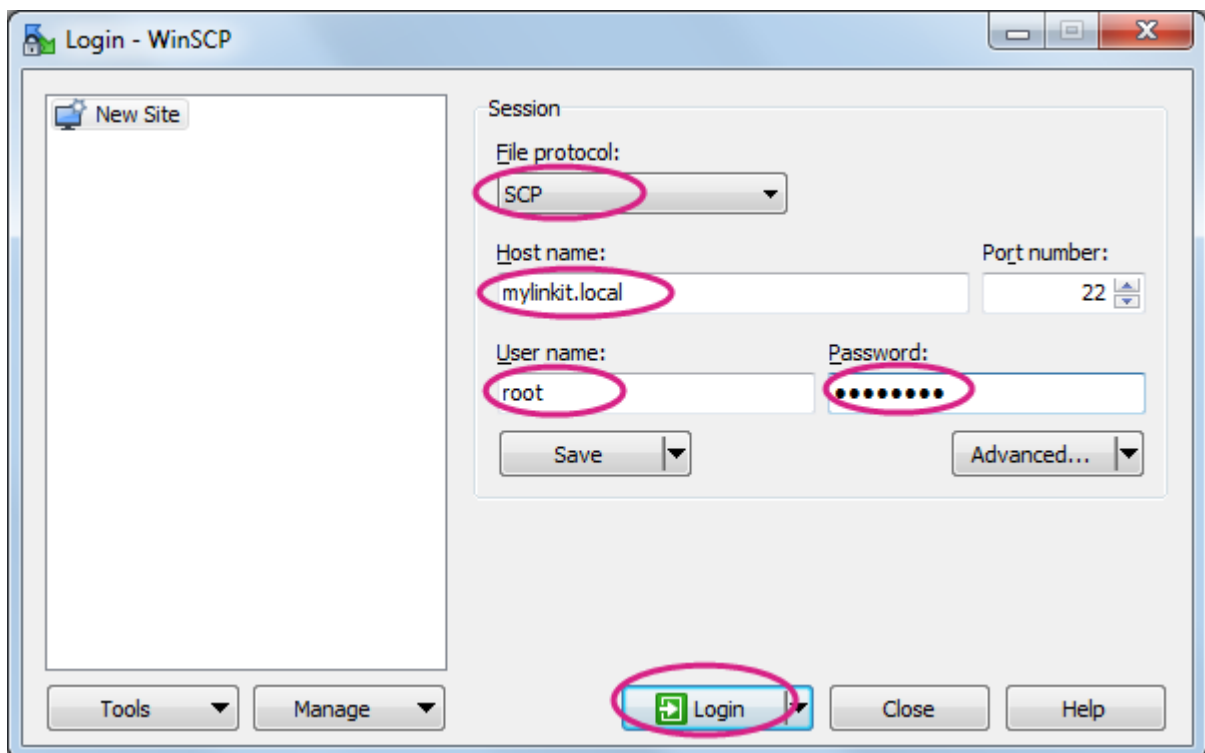
while True: створює нескінченний цикл. Зверніть увагу, що відступи у вигляді табуляції дозволяють визначити яка частина програми працює всередині циклу.

Отже ви створили файл з першою програмою на Python. Тепер необхідно завантажити цей файл в модуль Linkit Smart 7688, в папку /root . Зробіть це, використовуючи наведені нижче інструкції.

Завантаження файлів у модуль Linkit Smart 7688 через зашифроване з'єднання

Інструкція для Windows

Запустіть програму WinSCP, яку ви раніше встановили.



У полі File protocol оберіть **SCP**

У полі Host name вкажіть адресу модуля Linkit Smart 7688: mylinkit.local

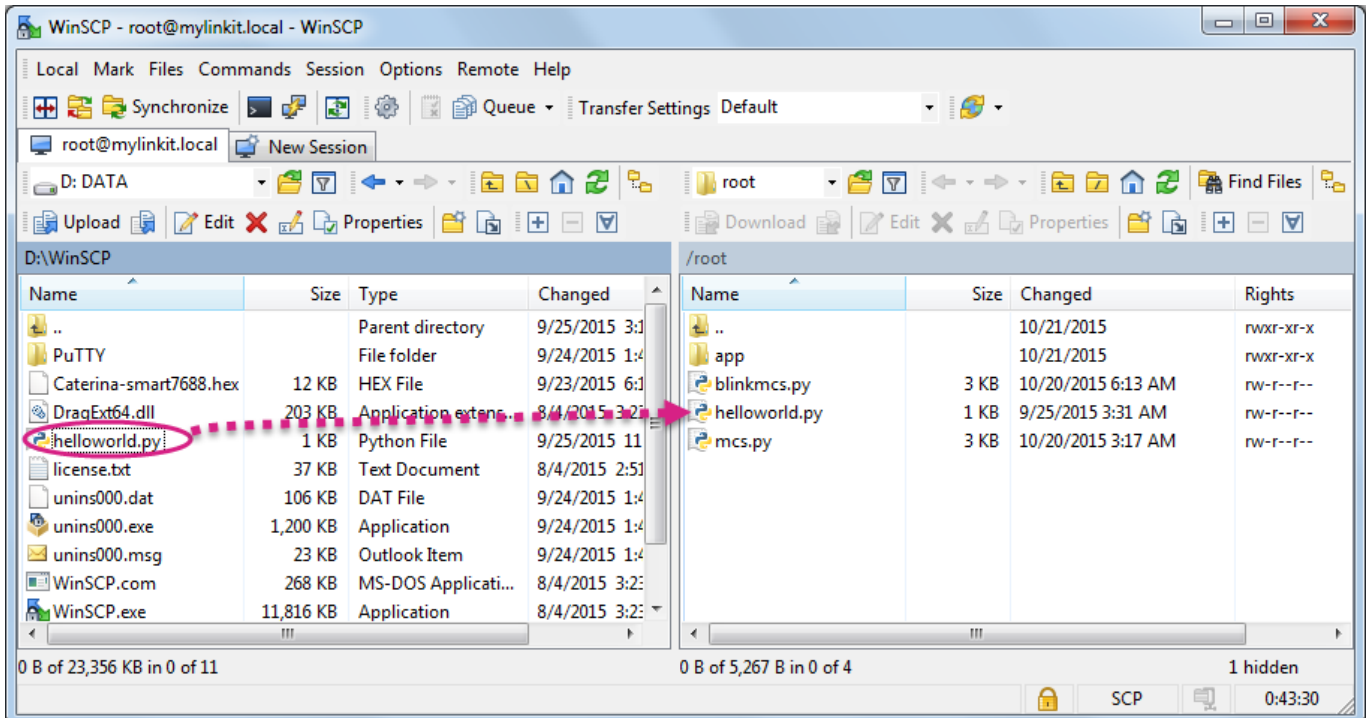
Номер порта: **22**

В полі User name вкажіть **root**

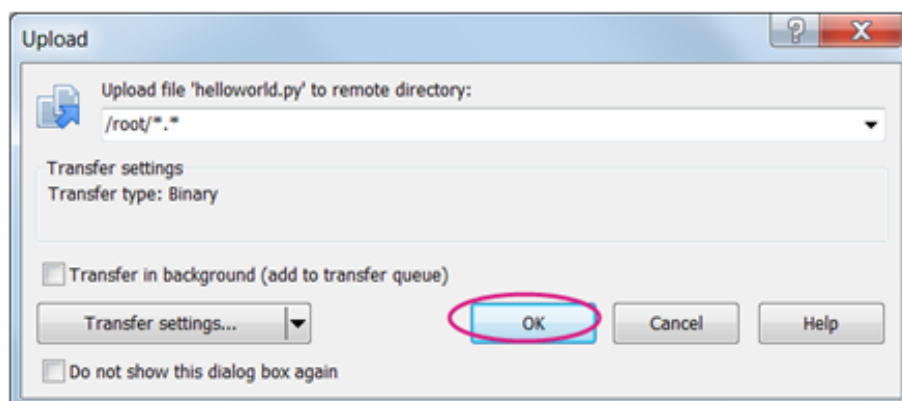
В полі Password введіть пароль, який ви раніше задали у веб інтерфейсі керування модулем

Натисніть **Login**. При появі попередження натисніть **OK**.

Відкриється вікно обміну файлами.



У лівій панелі оберіть файл на своєму комп'ютері, який ви хочете завантажити у модуль Linkit Smart 7688 та перетягніть цей файл у праву панель. З'явиться вікно **Upload** в якому необхідно буде підтвердити переміщення файлу натиснувши **OK**.



Як бачите, файл **blink.py** присутній.

Для запуску програми **blink.py** запустіть з консолі команду **python blink.py**

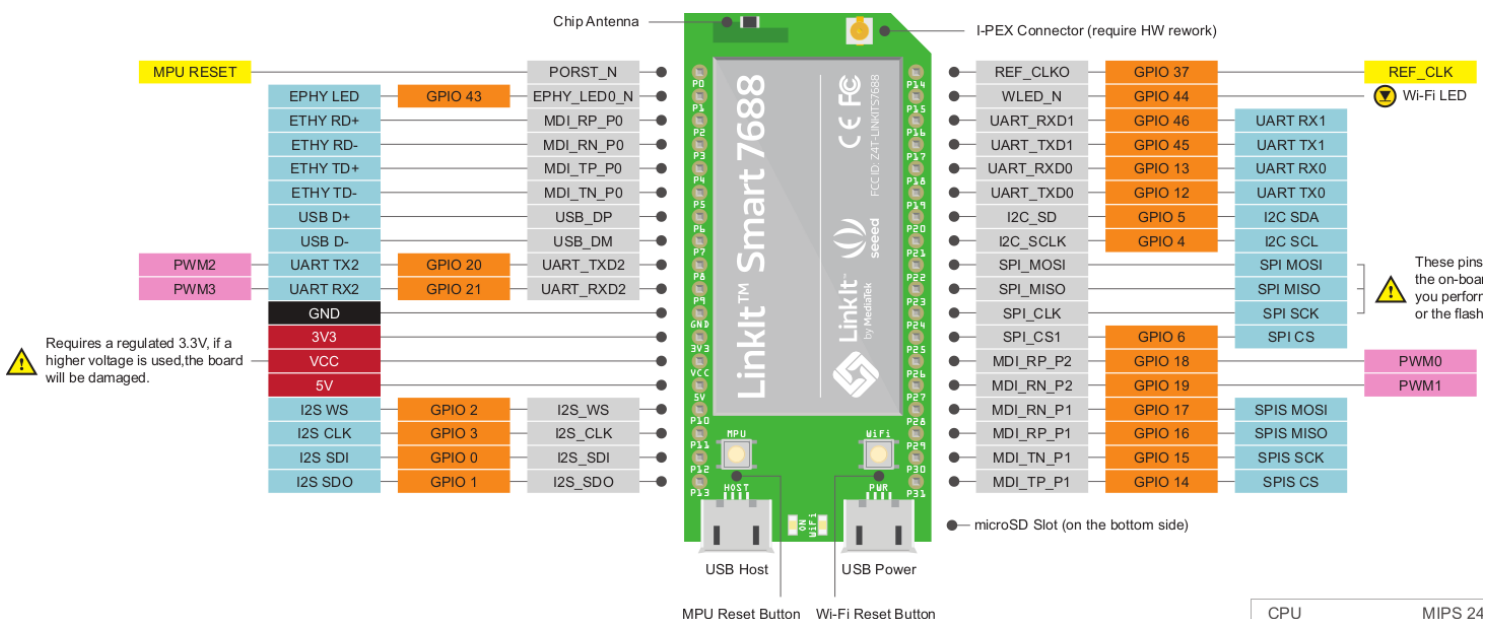
```
-----
|_ | W I R E L E S S   F R E E D O M
-----
CHAOS CALMER (15.05.1, r49203)
-----
* 1 1/2 oz Gin           Shake with a glassful
* 1/4 oz Triple Sec     of broken ice and pour
* 3/4 oz Lime Juice     unstrained into a goblet.
* 1 1/2 oz Orange Juice
* 1 tsp. Grenadine Syrup
-----
root@mylinkit:~# ls
blink.py
root@mylinkit:~# python blink.py
```

Переконайтеся, що після запуску програми червоний світлодіод почав блимати, змінюючи свій стан (світиться / не світиться) через кожну секунду.

Для зупинки програми натисніть комбінацію клавіш **Ctrl + C**

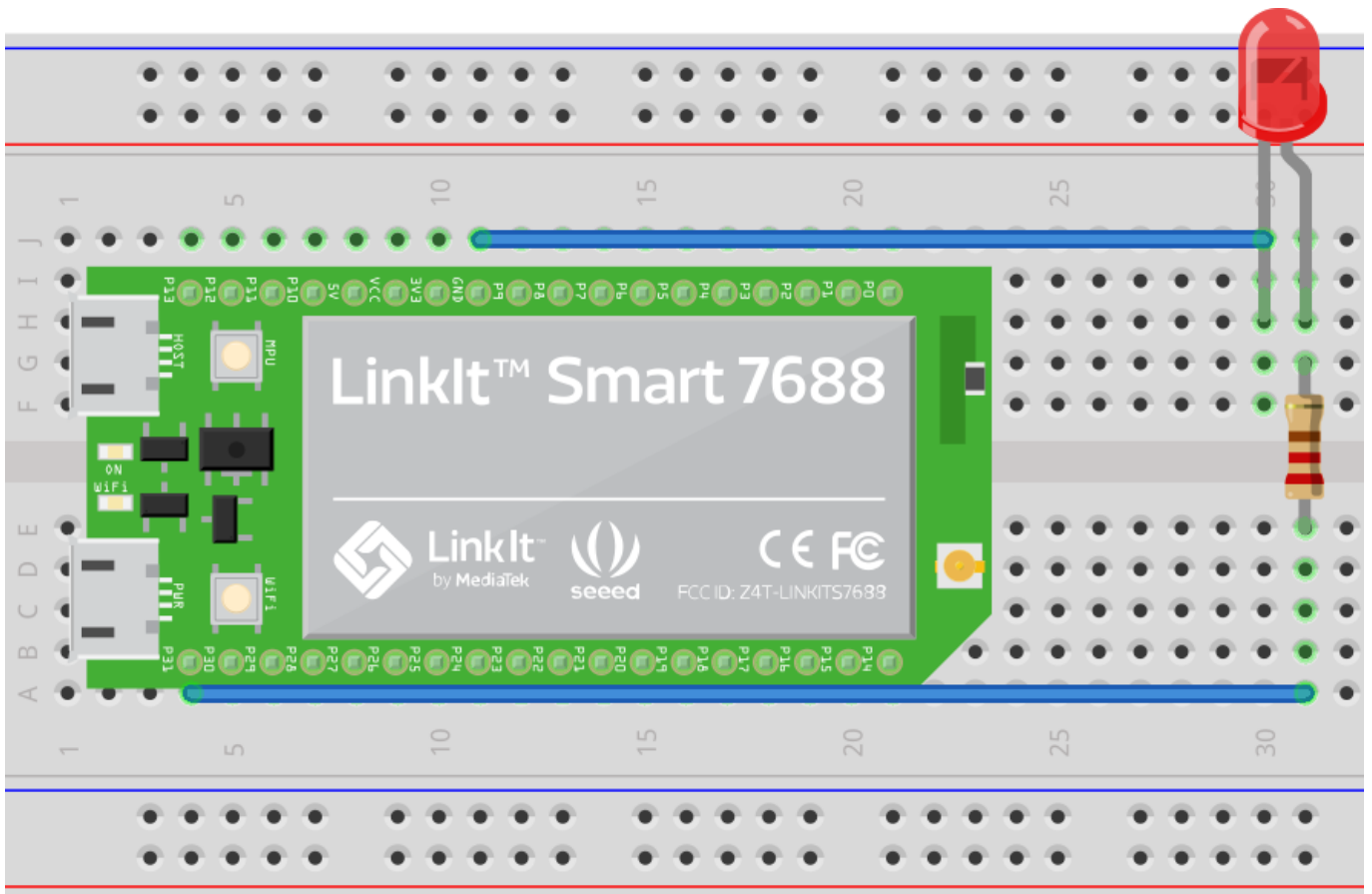
Підключимо свій власний світлодіод до модуля **Linkit Smart 7688**.

Тепер підключіть червоний світлодіод до контакта №14. Цей контакт знаходиться біля USB роз'єму живлення і підписаний P31. Зверніть увагу, що порядкові номери контактів, написані на модулі, не відповідають номерам, які необхідно використовувати в програмах. Де знаходиться необхідний вам контакт ви можете визначити з рисунку:

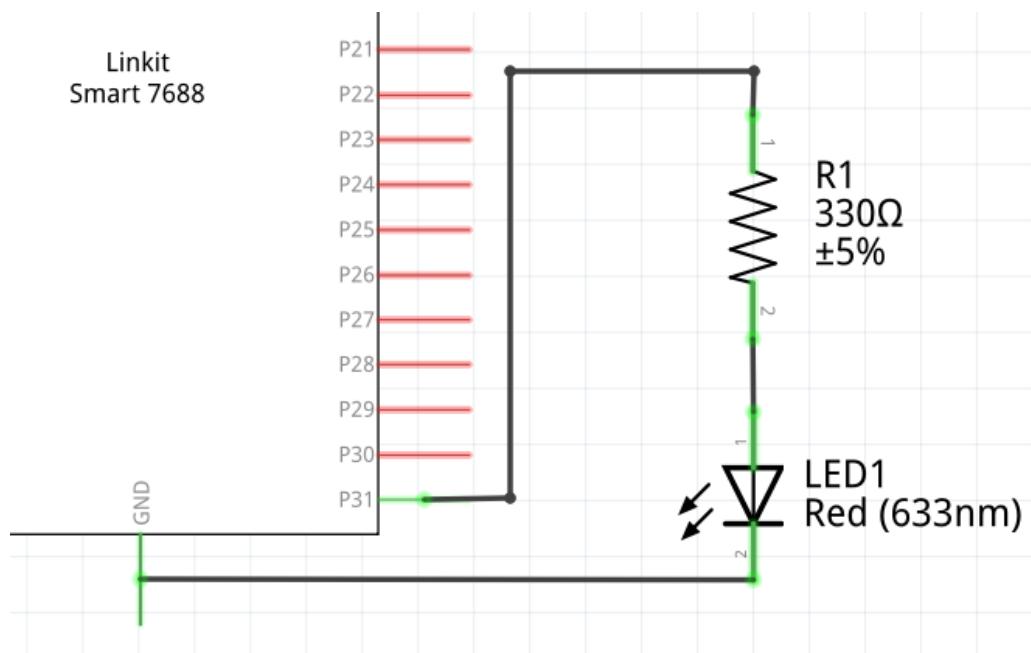


Цей рисунок у збільшеному вигляді присутній в кінці даного документа.

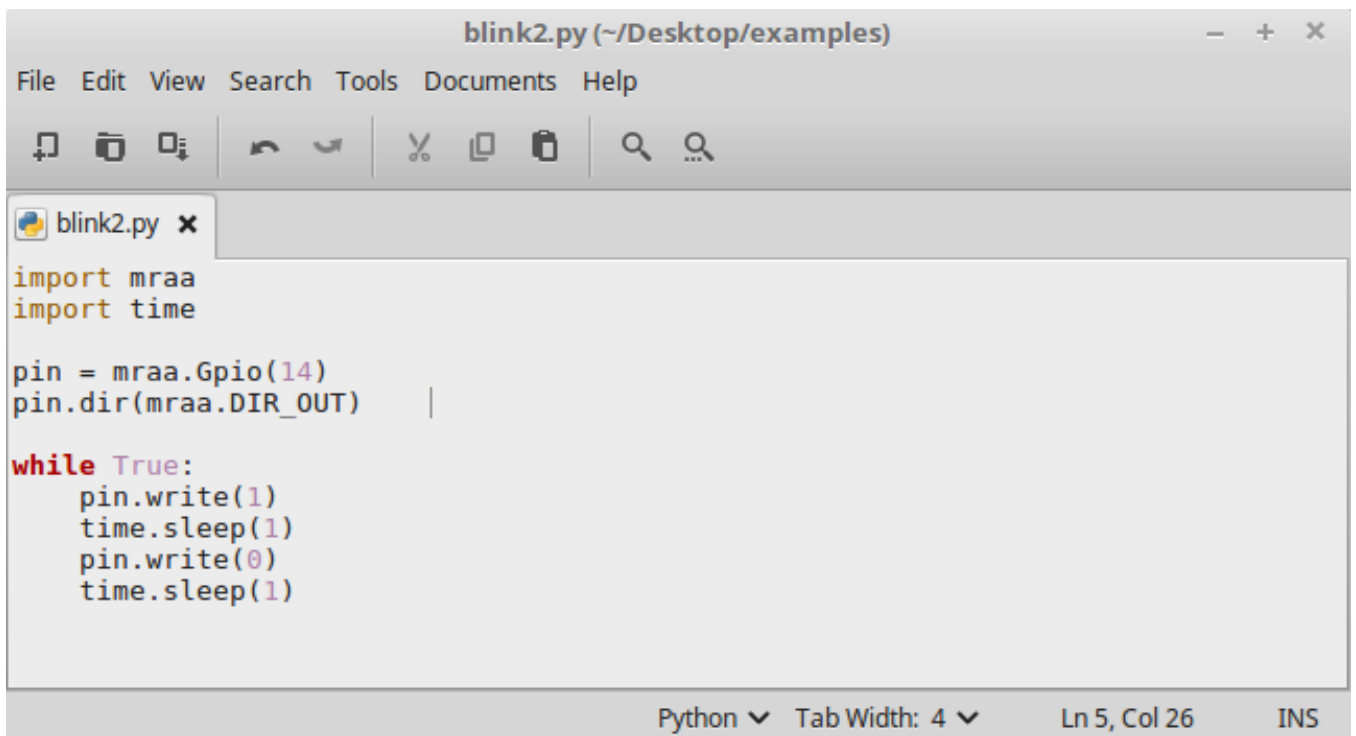
Схему підключення на макетній платі наведено на рисунку нижче. Опір резистора необхідно обрати порядку 330 Ом. До резистора необхідно підключити довший контакт світлодіода (анод), а до виводу модуля з надписом GND підключіть короткий контакт світлодіода (катод).



Звичайна схема підключення світлодіода виглядатиме так:



На своєму комп'ютері створіть текстовий файл з іменем "blink2.py". Додайте у файл наступний код на Python:



```
blink2.py (~/Desktop/examples)
File Edit View Search Tools Documents Help
import mraa
import time

pin = mraa.Gpio(14)
pin.dir(mraa.DIR_OUT)

while True:
    pin.write(1)
    time.sleep(1)
    pin.write(0)
    time.sleep(1)
```

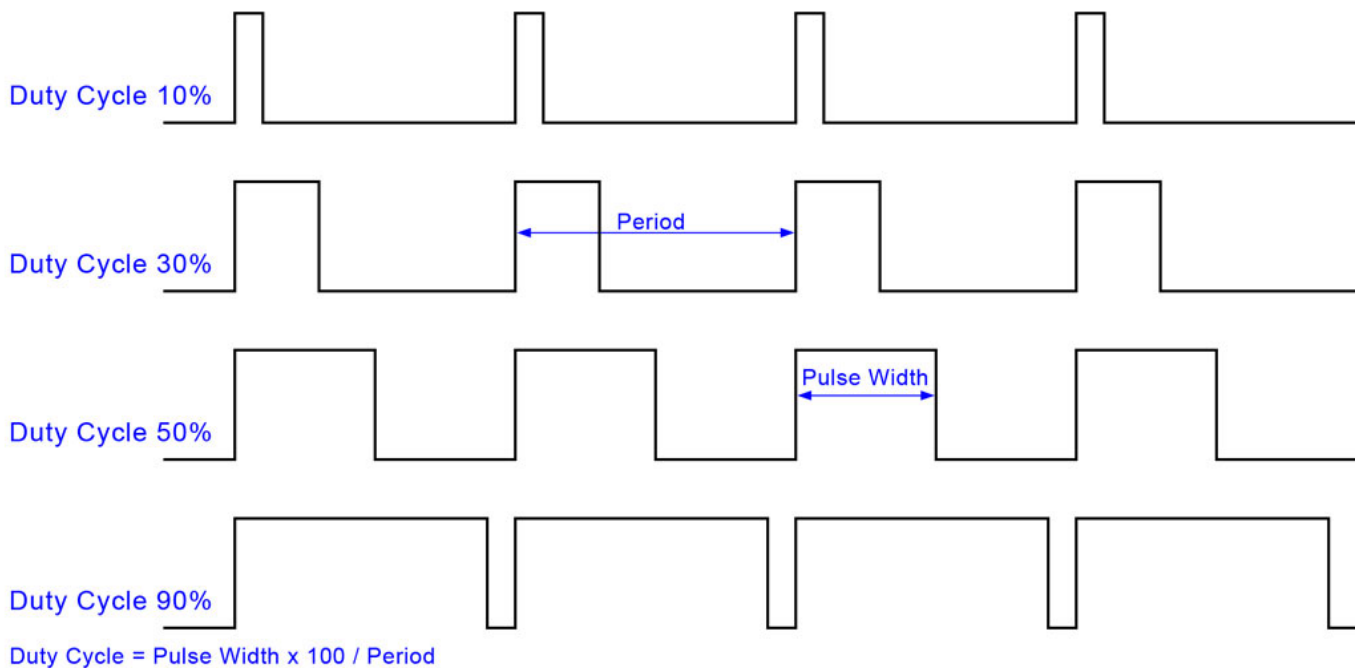
Зверніть увагу, що дана програма відрізняється від попередньої. По-перше, тут ми працюємо з виводом модуля №14, до якого підключений світлодіод. По-друге, тепер світлодіод засвічується після появи на виводі №14 напруги 3.3В (**pin.write(1)**) і перестає світитися після появи на виводі №14 нуля вольт (**pin.write(0)**).

Завантажте дану програму у модуль, запустіть і перевірте, що світлодіод дійсно миготить. Спробуйте змінювати затримку функції **time.sleep(1)** і задати затримку на 1 секунду, а 0.3 секунди.

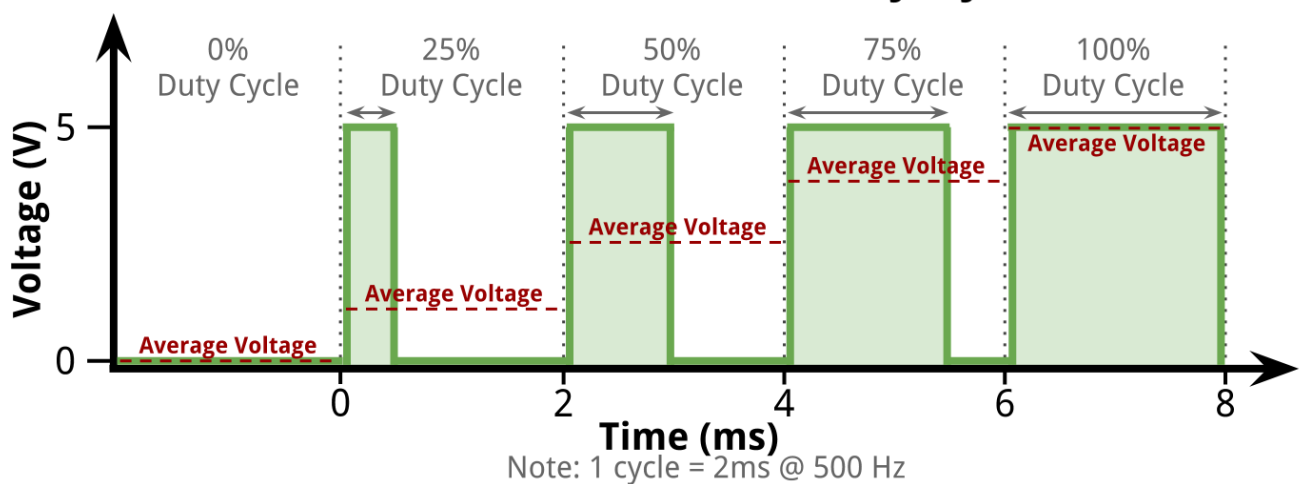
Робота 3.

Керуємо яскравістю світлодіода за допомогою широтно-імпульсної модуляції (PWM)

Широтно-імпульсна модуляція (PWM) дозволяє змінювати середню напругу на виході мікросхеми за рахунок зміни тривалості імпульсу при незмінному періоді імпульсів.

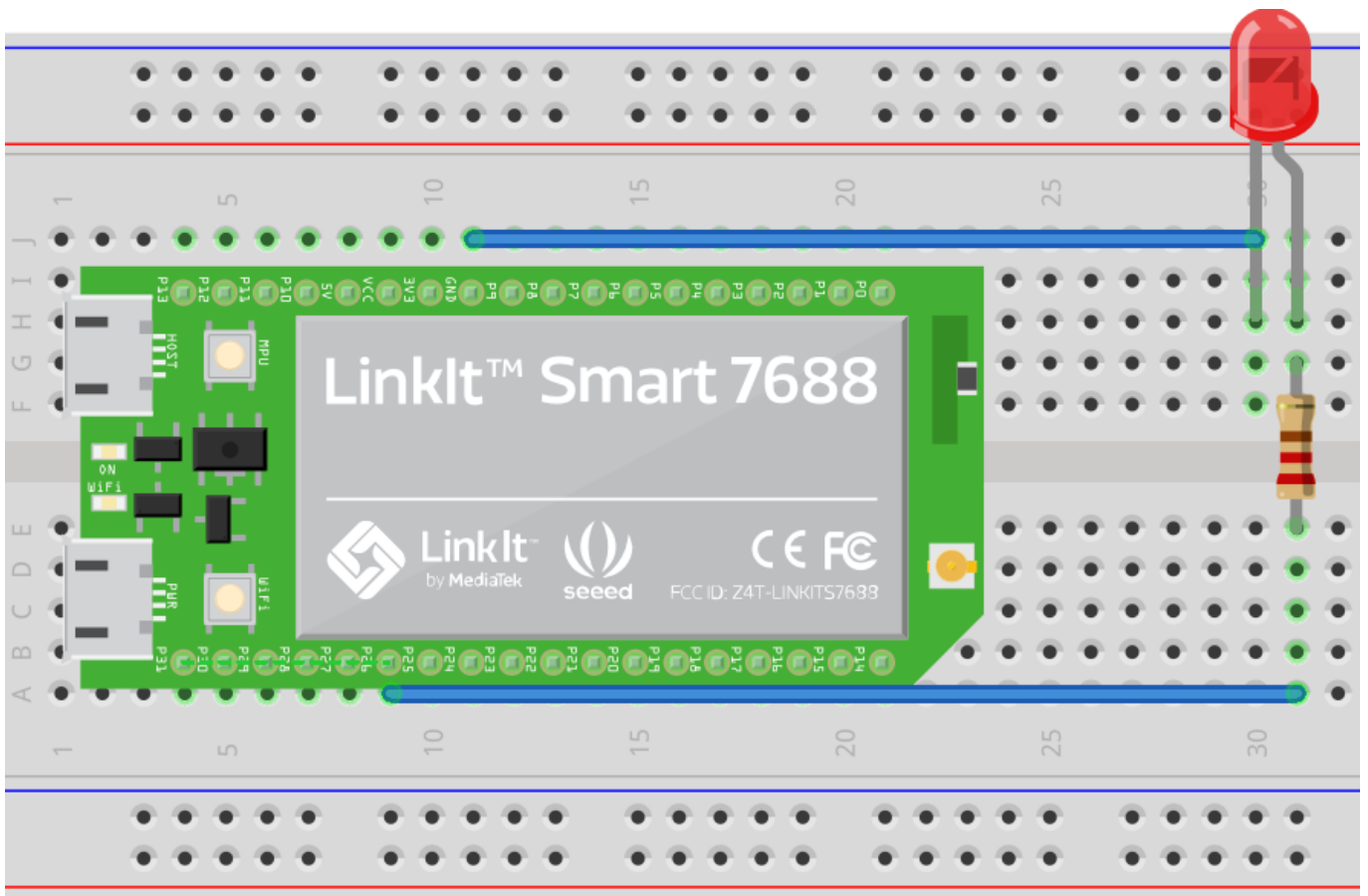


Pulse Width Modulation Duty Cycles



За допомогою PWM ми будемо змінювати середню напругу прикладену до світлодіоду, що приведе до зміни яскравості світіння.

Складіть схему зі світлодіодом підключеним до PWM виходу:



На своєму комп'ютері створіть текстовий файл з іменем "pwm.py". Додайте у файл наступний код на Python:

```
pwm.py (~/Desktop/examples)
File Edit View Search Tools Documents Help
import mraa
import time

pin = mraa.Pwm(18)
pin.period_ms(2)
pin.enable(True)
pin.write(0.25)

while True:
    time.sleep(1)
```

Python Tab Width: 4 Ln 7, Col 21 INS

За допомогою **pin = mraa.Pwm(18)** створюємо PWM на виводі 18.

За допомогою **pin.period_ms(2)** задаємо частоту PWM сигналу рівну 500 Гц (період 2мс).

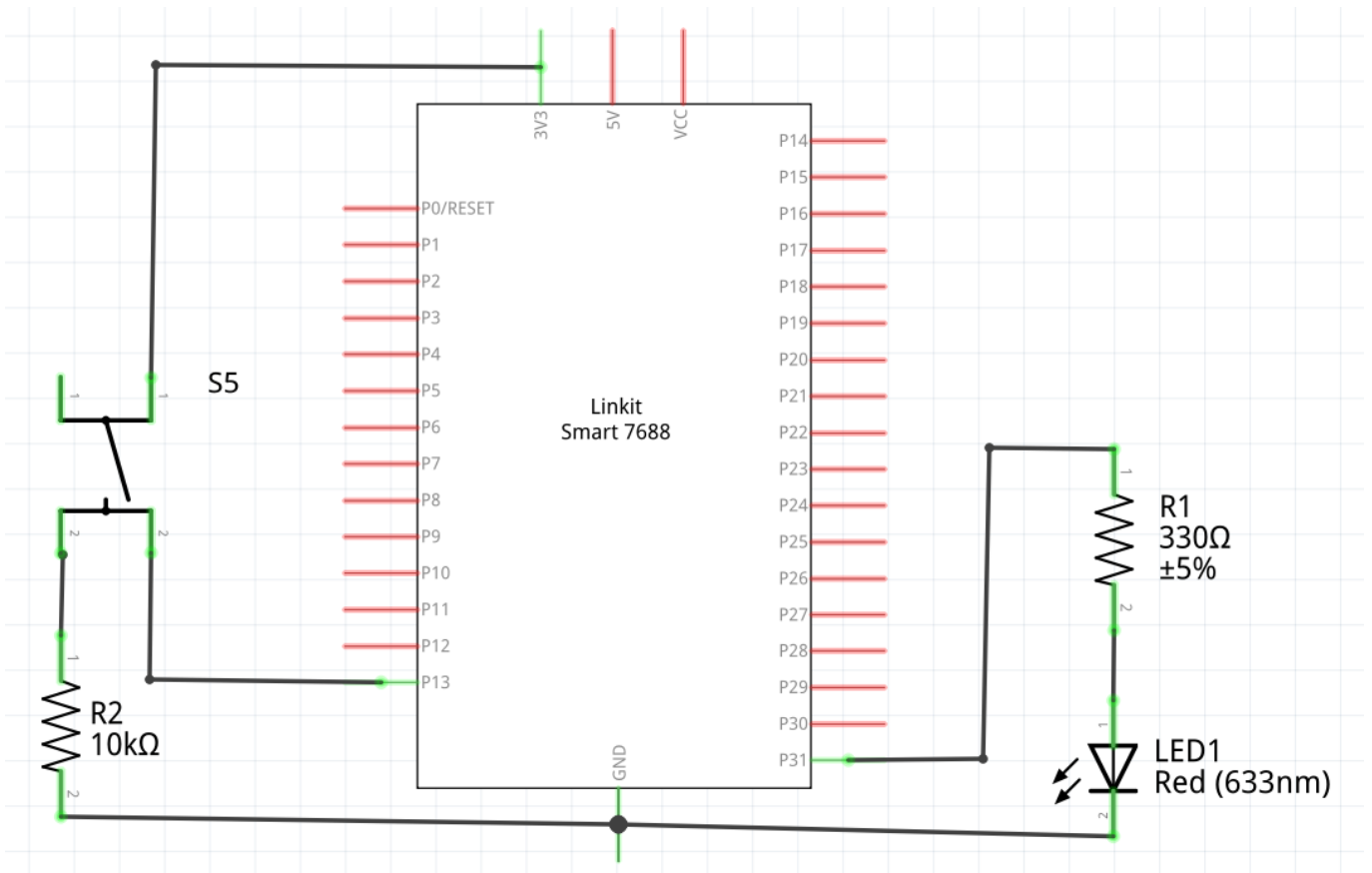
За допомогою **pin.enable(True)** активуємо PWM вихід.

За допомогою **pin.write(0.25)** задаємо коефіцієнт заповнення (duty cycle).

Завантажте програму **pwm.py** у модуль і запустіть її. Спробуйте змінювати коефіцієнт заповнення, роблячи його рівним 0.1, 0.3, 0.5, 0.75. Переконайтеся, що це приводить до зміни яскравості світіння світлодіоду.

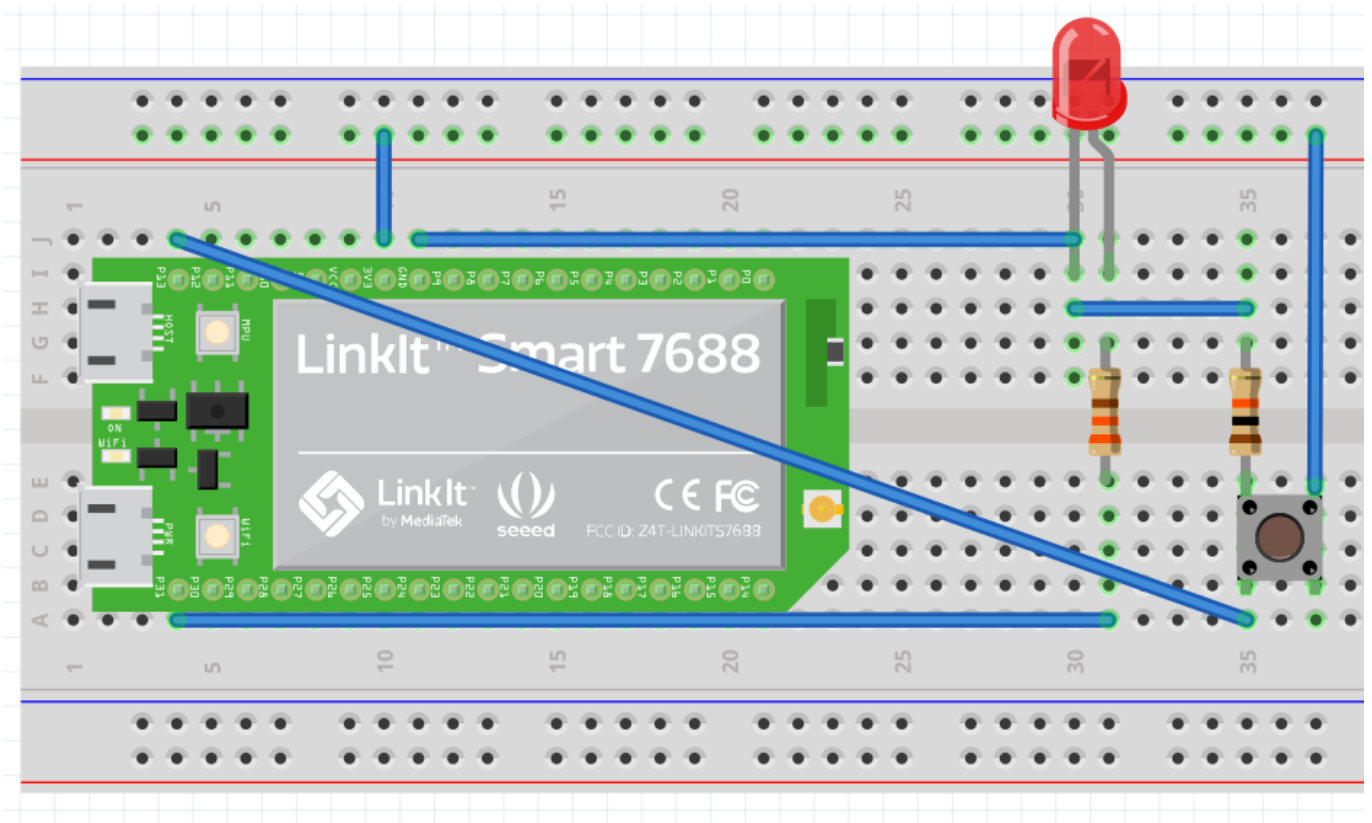
Робота 4. Підключаємо кнопку

Зараз ми підключимо кнопку до модуля Linkit Smart 7688. Схема підключення наведена нижче. Коли ми підключаємо до модуля живлення за допомогою USB кабеля, на виводі 3V3 з'являється напруга 3.3 Вольта. Коли перемикач розімкнутий (кнопка не натиснута), контакт №1 (підписаний на модулі, як P13) буде з'єднаний з нулем вольт (GND) через резистор. Якщо ми натиснемо кнопку, перемикач замкнеться, через резистор почне протікати струм і до резистору буде прикладена напруга 3.3 Вольта, яка також буде присутня на контакті №1 (підписаний на модулі, як P13). Опір резистора підключеного до кнопки обираємо 10 КОм.



Зверніть увагу, що до модуля Linkit Smart 7688 також підключено світлодіод. Далі ми напишемо програму на Python, яка буде засвічувати світлодіод, коли кнопка натиснута. Якщо кнопка не натиснута, світлодіод не буде світитися.

Складіть наведену вище схему на макетній платі:



На своєму комп'ютері створіть текстовий файл з іменем "button.py". Додайте у файл наступний код на Python:

```
button.py (~/Desktop/examples)
File Edit View Search Tools Documents Help
button.py x blink2.py x
import mraa
import time

button = mraa.Gpio(1)
button.dir(mraa.DIR_IN)

led = mraa.Gpio(14)
led.dir(mraa.DIR_OUT)

button_state = 0

while True:
    button_state = button.read()
    print "GPIO1 state:", button_state
    led.write(button_state)
    time.sleep(0.1)
```

Частина програми, що керує світлодіодом вам уже знайома з попередніх прикладів.

За допомогою

```
button = mraa.Gpio(1)  
button.dir(mraa.DIR_IN)
```

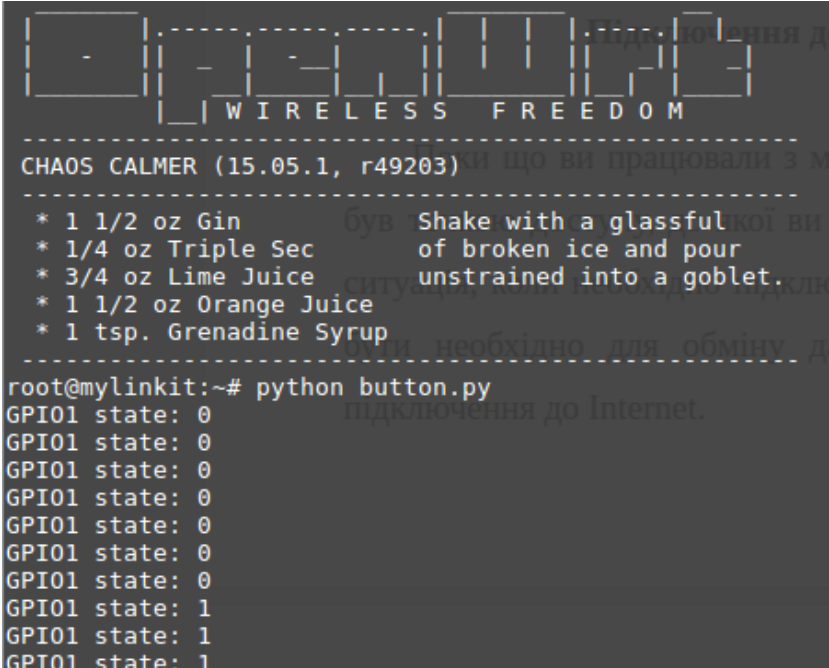
ми створюємо змінну **button**, що буде описувати контакт модуля з номером 1 (підписаний на модулі, як P13) та вказуємо, що ми будемо зчитувати стан цього контакту. Якщо до контакту прикладена напруга 3.3 Вольта, при спробі зчитування отримаємо число 1. Якщо до контакту прикладена напруга 0 Вольт, при спробі зчитування отримаємо число 0.

Далі ми створюємо змінну **button_state** і записуємо в неї число 0:

```
button_state = 0
```

Після цього в нескінченному циклі зчитуємо значення кнопки за допомогою функції **read()** і записуємо одержане значення у змінну **button_state**. Далі виводимо зчитане значення кнопки на консоль (термінал) модуля за допомогою функції **print**. Після цього ми передаємо стан змінної **button_state** на той вихід модуля до якого підключений світлодіод (за допомогою **led.write(button_state)**). Якщо **button_state = 1**, на виході до якого підключено світлодіод встановлюється 3.3 Вольта і світлодіод починає світитися. Якщо **button_state = 0**, на виході до якого підключено світлодіод встановлюється 0 Вольт і світлодіод перестає світитися.

Завантажте програму **button.py** у модуль і запустіть її. Переконайтеся, що після натиснення кнопки світлодіод починає світитися, а після відпускання кнопки перестає світитися. Зчитаний стан кнопки виводиться на консоль модуля.



```
-----  
CHAOS CALMER (15.05.1, r49203)  
-----  
* 1 1/2 oz Gin Shake with a glassful  
* 1/4 oz Triple Sec of broken ice and pour  
* 3/4 oz Lime Juice unstrained into a goblet.  
* 1 1/2 oz Orange Juice  
* 1 tsp. Grenadine Syrup  
-----  
root@mylinkit:~# python button.py  
GPIO1 state: 0  
GPIO1 state: 0  
GPIO1 state: 0  
GPIO1 state: 0  
GPIO1 state: 0  
GPIO1 state: 0  
GPIO1 state: 0  
GPIO1 state: 1  
GPIO1 state: 1  
GPIO1 state: 1
```

Робота 5. Використовуємо переривання.

У програми з попереднього прикладу є недолік. Необхідно досить часто зчитувати значення кнопки, щоб не пропустити момент, коли її натискають, або відпускають. Якщо програма замість зчитування значення кнопки буде робити щось інше (наприклад, миготіти світлодіодом, чи відображати код веб сторінки) і в цей час хтось натисне кнопку, а потім швидко відпустить, програма цього не помітить, оскільки не зчитувала значення кнопки.

Для вирішення такої проблеми використовують принцип переривань. Спершу пишуть основну програму, що виконує певну функцію. Потім створюють допоміжну програму, яка називається “підпрограма переривання”. Визначають умову після виконання якої запуститься підпрограма переривання. Ідея полягає у тому, що більшість часу працює основна програма. Якщо виконується визначена умова, основна програма зупиняється (переривається) і запускається підпрограма переривання. Після закінчення підпрограми переривання, основна програма продовжує свою роботу з того місця, де її перервали. Якщо підпрограми переривань будуть короткими і їх виконання займатиме мало часу, вони майже не впливатимуть на роботу основної програми.

Розглянемо принцип роботи переривань на прикладі. будемо в основній програмі миготіти світлодіодом, а на натиснення кнопки реагувати в перериванні. Схему залишимо із попереднього завдання. Однак змінимо програму на Python.

На своєму комп’ютері створіть текстовий файл з іменем “interrupt.py”. Додайте у файл наступний код на Python:

```
interrupt.py (~/Desktop/examples)
File Edit View Search Tools Documents Help
interrupt.py x
import mraa
import time

def callback(userdata):
    print button.read()

global button
button = mraa.Gpio(1)
button.dir(mraa.DIR_IN)
button.isr(mraa.EDGE_BOTH, callback, None)

led = mraa.Gpio(14)
led.dir(mraa.DIR_OUT)

while True:
    led.write(1)
    time.sleep(0.5)
    led.write(0)
    time.sleep(0.5)

Python Tab Width: 4 Ln 1, Col 1 INS
```

Код для миготіння світлодіодом вам уже знайомий із попередніх прикладів.

За допомогою

```
def callback(userdata):  
    print button.read()
```

описуємо підпрограму перивання, яка буде викликатися після виконання певної умови. Ім'я підпрограми переривання -- **callback**. Однак ви можете задати інше ім'я. Єдине, що робить підпрограма переривання в даному прикладі -- це за допомогою функції **print** виводить на консоль модуля Linkit Smart 7688 значення на вході, до якого підключена кнопка, зчитане за допомогою **button.read()**. Зверніть увагу, що змінна **button**, що описує контакт модуля до якого підключена кнопка, створюється в основній програмі, а зчитується в підпрограмі переривання. Для того, щоб змінна **button**, створена в основній програмі, була доступна з підпрограми переривання, необхідно зробити цю змінну глобальною написавши в основній програмі **global button** перед записом у змінну **button**.

За допомогою

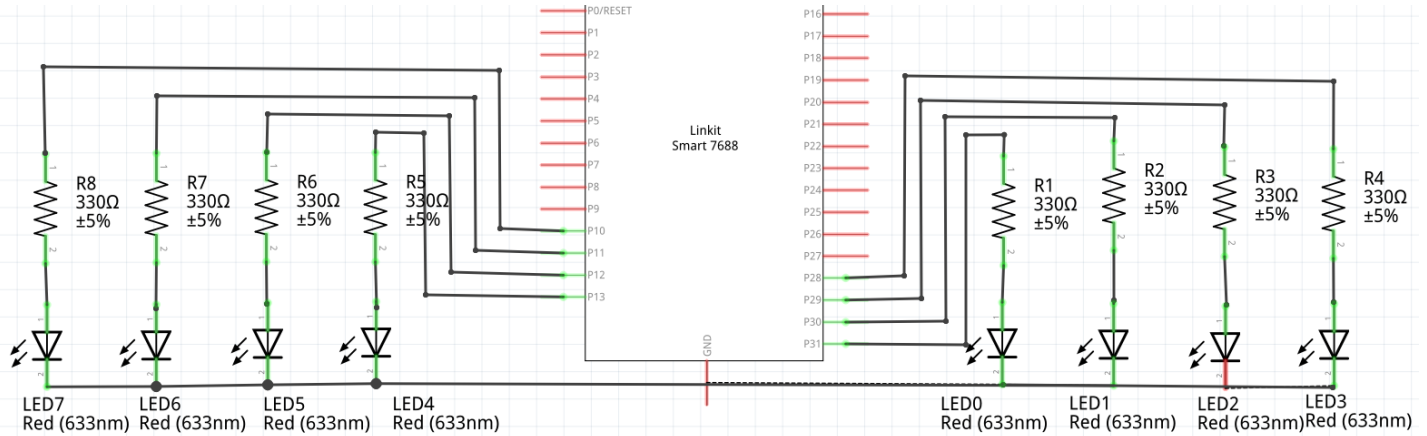
button.isr(mraa.EDGE_BOTH, callback, None)

ми визначаємо, що після зміни напруги на вході, до якого підключена кнопка **button**, необхідно зробити переривання і викликати програму з іменем **callback**. **EDGE_BOTH** означає, що переривання необхідно виконувати як при зміні напруги на вході з 0 до 3.3 Вольт, так і при зміні напруги на вході з 3.3 до 0 Вольт. Третій аргумент завжди повинен бути **None**.

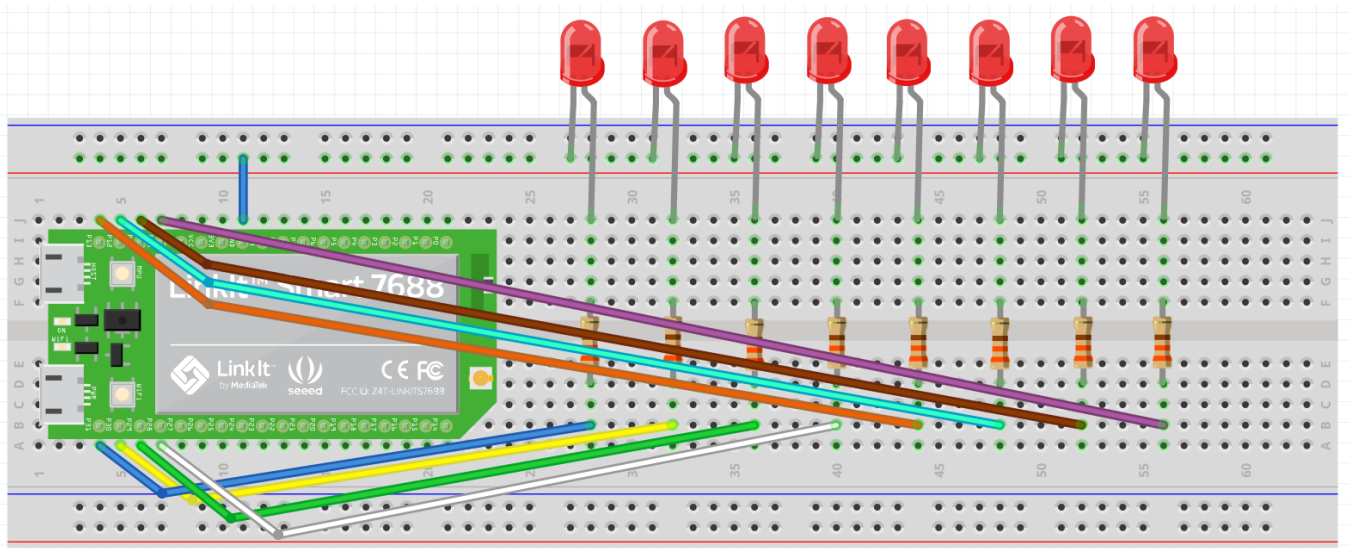
Завантажте програму **interrupt.py** у модуль і запустіть її. Переконайтеся, що світлодіод миготить і при цьому одразу після натиснення на кнопку на консоль виводиться число 1, а одразу після відпускання кнопки на консоль виводиться число 0. Як бачите, підпрограма переривання викликається лише після зміни напруги на вході, до якого підключена кнопка (при натисканні, або відпусканні кнопки).

Робота 6. Реалізуємо програмно рухомі вогники

Підключіть до модуля Linkit Smart 7688 вісім світлодіодів за наступною схемою:



З'єднання на макетній платі:



Опір резисторів повинен лежати в межах кількох сотень Ом (наприклад, 330Ом).

На своєму комп'ютері створіть текстовий файл з іменем "running_leds.py".
Додайте у файл наступний код на Python:

```
running_leds.py x
import mraa
import time

def drive_leds(leds, state):
    for i in range(len(leds)):
        leds[i].write(state >> i & 0x01)

led0 = mraa.Gpio(14)
led1 = mraa.Gpio(15)
led2 = mraa.Gpio(16)
led3 = mraa.Gpio(17)
led4 = mraa.Gpio(1)
led5 = mraa.Gpio(0)
led6 = mraa.Gpio(3)
led7 = mraa.Gpio(2)

led0.dir(mraa.DIR_OUT)
led1.dir(mraa.DIR_OUT)
led2.dir(mraa.DIR_OUT)
led3.dir(mraa.DIR_OUT)
led4.dir(mraa.DIR_OUT)
led5.dir(mraa.DIR_OUT)
led6.dir(mraa.DIR_OUT)
led7.dir(mraa.DIR_OUT)

leds = [led0, led1, led2, led3, led4, led5, led6, led7]

state = 1

while True:
    drive_leds(leds, state)
    state = state << 1 | ~(state >> 7) & 0x01
    time.sleep(0.5)
```

Як бачите, ми створюємо 8 змінних (**led0-led7**), кожна з яких описує контакт модуля до якого підключений світлодіод. Далі налаштовуємо ці вісім контактів працювати на вихід (щоб можна було програмно встановлювати на кожному виході напругу 0 Вольт, або 3.3 Вольт). Після цього створюємо масив **leds**, елементами якого є змінні **led0-led7**:

```
leds = [led0, led1, led2, led3, led4, led5, led6, led7]
```

Далі створюємо змінну **state**, кожен біт якої визначатиме світлитися відповідний світлодіод, чи ні. Біт змінної **state** з номером 0 відповідає світлодіоду з номером 0, біт змінної **state** з номером 7 відповідає світлодіоду з номером 7 і т.д. Якщо відповідний біт приймає значення 1, світлодіод повинен світитися. Якщо відповідний біт приймає значення 0, світлодіод не повинен світитися. Під час створення змінної **state** присвоюємо їй значення 1. Це означає, що в такому

випадку повинен світитися лише світлодіод з номером 0, бо лише біт з номером 0 приймає значення 1.

Функція **drive_leds** засвічує світлодіоди масиву **leds** у відповідності до значень бітів змінної **state**:

```
def drive_leds(leds, state):  
    for i in range(len(leds)):  
        leds[i].write(state >> i & 0x01)
```

Як бачите, функція приймає два аргументи - масив **leds** і змінну **state**. Всередині функції, в циклі **for**, проходимося по кожному біту змінної **state** (з номерами від 0 до 7) і в залежності від значення біту або засвічуємо світлодіод з відповідним номером, або не засвічуємо. Змінна **i** містить номер ітерації циклу. Функція **len** - це стандартна функція Python, що повертає розмір (кількість елементів) масиву, який передається в функцію **len**. Функція **range(arg)** повертає масив чисел у кількості **arg** зі значеннями від **0** до **arg-1**. Наприклад:

```
>>> range(10)  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Отже цикл **for i in range(arg)** виконає операції тіла циклу **arg** разів, при цьому на кожній ітерації циклу змінна **i** прийматиме нове значення з діапазону від **0** до **arg-1**. Зміна значень змінної **i** буде відбуватися в порядку: **0, 1, 2, 3, ..., arg-2, arg-1**. Для доступу до елемента масиву **leds** з номером **i** використаємо **leds[i]**.

Конструкція **state >> i & 0x01** дозволяє виділити значення **i**-го біту змінної **state**. Зсуваємо змінну **state** вправо на **i** бітів (**i**-й біт стає бітом з номером 0, тобто наймолодшим), а потім робимо побітовий **AND** результату з числом 1, тобто обнуляємо всі біти окрім наймолодшого. Якщо **i**-тий біт змінної **state** приймав значення 1, в результаті даної операції отримаємо число 1. Якщо **i**-тий біт змінної **state** приймав значення 0, в результаті даної операції отримаємо число 0.

У нескінченному циклі основної програми робимо наступне:

while True:

drive_leds(leds, state)

state = state << 1 | (~(state >> 7) & 0x01)

time.sleep(0.5)

На кожній ітерації циклу виводимо на світлодіоди значення бітів змінної **state** за допомогою **drive_leds(leds, state)**.

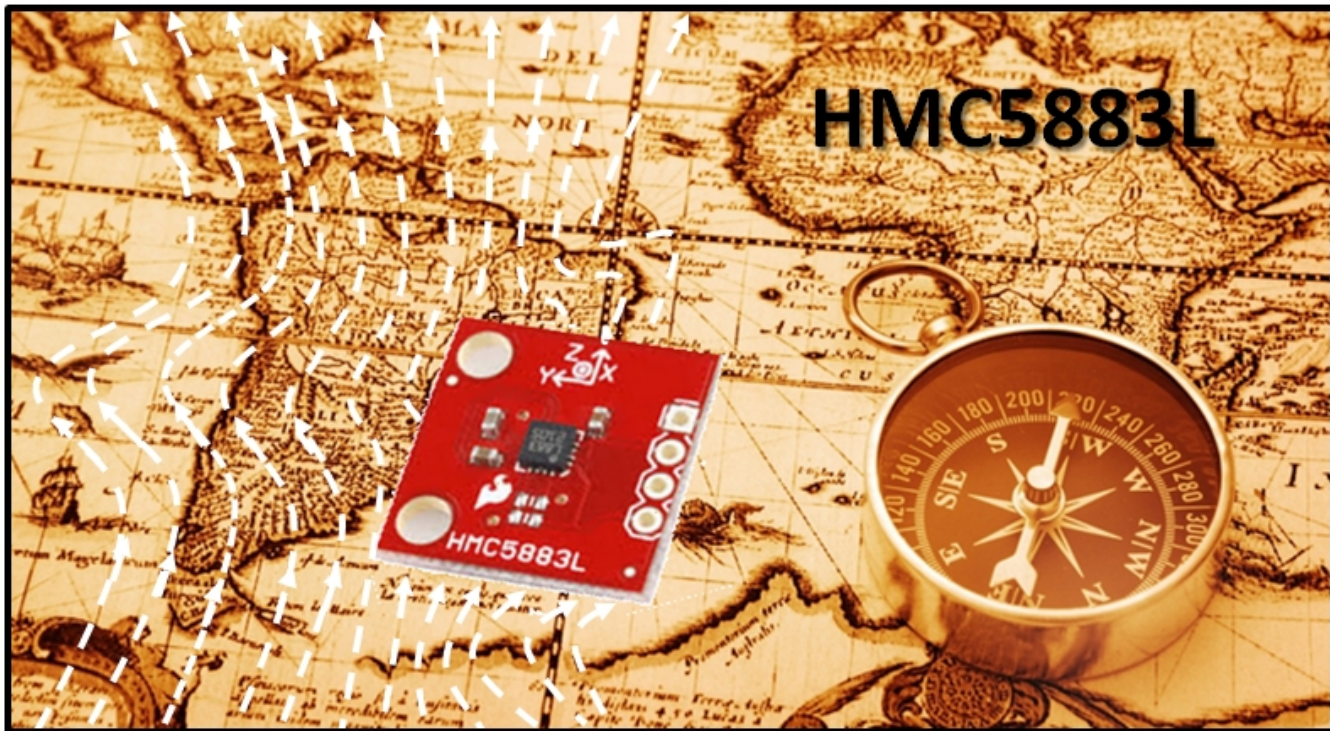
Після цього за допомогою **state << 1 | (~(state >> 7) & 0x01)** зсуваємо вміст змінної **state** на один розряд вліво і при цьому переписуємо інвертоване значення 7-го біту старого значення **state** в нульовий біт результату виразу. Отриманий результат записуємо в змінну **state**. Спробуйте уявити, як це відбувається.

Далі робимо затримку на 0.5 секунд за допомогою **time.sleep(0.5)**. Після завершення затримки переходимо до наступної ітерації нескінченного циклу і описані дії повторюються.

Завантажте програму **running_leds.py** у модуль і запустіть її. Переконайтеся, що світлодіоди блимають у відповідності, до алгоритму роботи програми.

Робота 7. Підключаємо електронний компас з інтерфейсом I2C

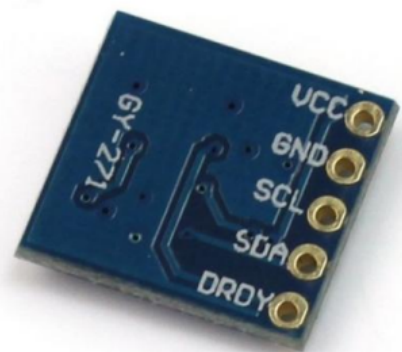
В даній роботі ми підключимо до модуля Linkit Smart 7688 електронний компас (магнітометр) HMC5883L інтерфейсом I2C. Будемо зчитувати значення з компасу і виводити його в термінал модуля.



Електронний компас (магнітометр) HMC5883L може виглядати по різному:



Gnd
Vcc
SDA
SCL



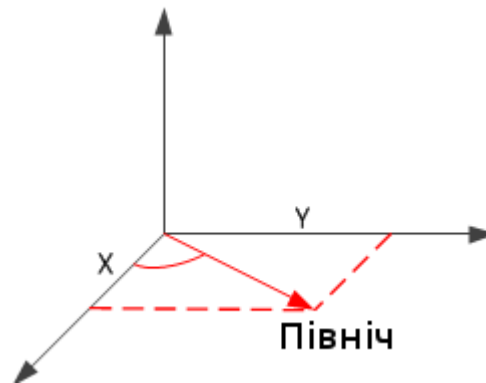
Однак у всіх варіантів модуля є спільні контакти: VCC, GND, SDA, SCL.

До контакту підключають “+” джерела напруги 3 ... 5 Вольт.

До контакту GND підключають “-” цього ж джерела напруги.

Контакти SDA та SCL використовують для передачі даних по інтерфейсу I2C.

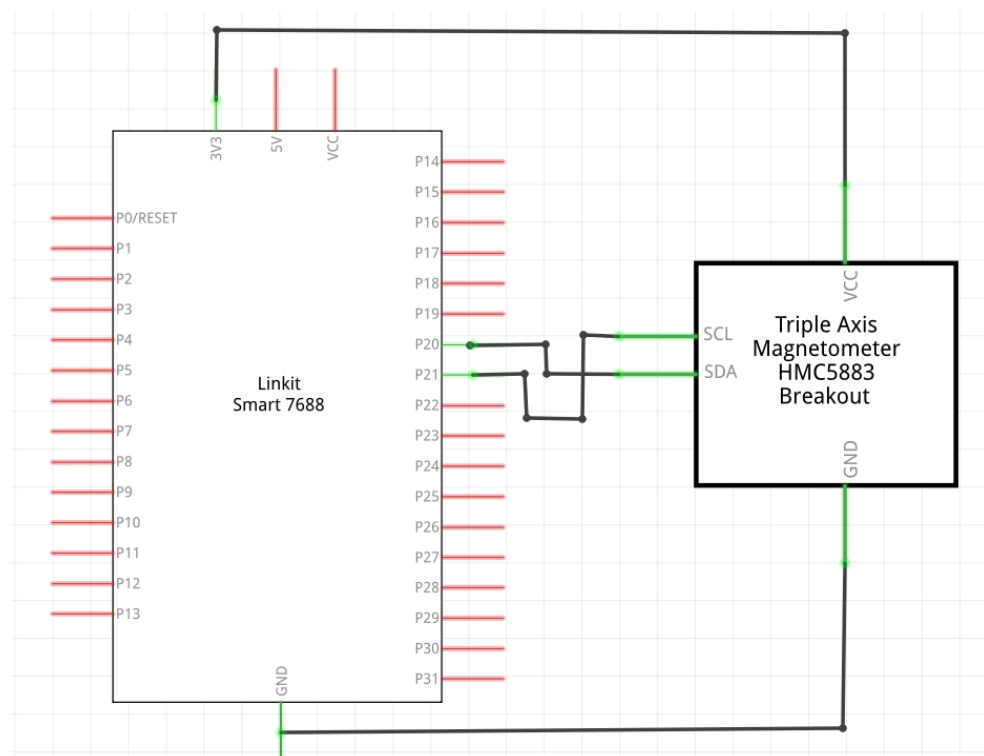
Електронний компас (магнітометр) HMC5883L дозволяє вимірювати напруженість (силу) магнітного поля Землі по осях X, Y, Z. З цих даних можна розрахувати кут між напрямком на північ і напрямком на який вказує вісь x компасу, як $\arctg(H_y/H_x)$, де H_y та H_x це напруженості магнітного поля Землі по осях Y та X компаса HMC5883L:



Для модуля Linkit Smart 7688 існує бібліотка UPM (портована з Intel Edison), що дозволяє дуже просто взаємодіяти з безліччю датчиків з програм на C, Python, NodeJS. Приклади роботи з різними датчиками з використанням UPM та програм на Python: <https://github.com/intel-iot-devkit/upm/tree/master/examples/python>

Датчик HMC5883L теж підтримується бібліотекою UPM.

Спершу підключіть HMC5883L до модуля Linkit Smart 7688:



Контакт GND датчика HMC5883L необхідно підключити до контакту GND модуля Linkit Smart 7688.

Контакт VCC датчика HMC5883L необхідно підключити до контакту 3.3V модуля Linkit Smart 7688.

Контакт SCL датчика HMC5883L необхідно підключити до контакту модуля Linkit Smart 7688 біля якого написано P21 (це вихід SCL модуля Linkit Smart 7688).

Контакт SDA датчика HMC5883L необхідно підключити до контакту модуля Linkit Smart 7688 біля якого написано P20 (це контакт SDA модуля Linkit Smart 7688).

На своєму комп'ютері створіть текстовий файл з іменем "i2c_compass.py".
Додайте у файл наступний код на Python:

```
i2c_compass.py x
from time import sleep
import pyupm_hmc5883l as hmc5883l

hmc = hmc5883l.Hmc5883l(0)
hmc.set_declination(0.3)

while True:
    hmc.update() # Update the data
    pos = hmc.coordinates() # Read raw coordinates
    hdg = hmc.heading() # Read heading
    dir = hmc.direction() # Read direction

    # Print values
    print("Coord: %5d %5d %5d" % (pos[0], pos[1], pos[2]))
    print("Heading: %5.2f" % (hdg))
    print("Direction: %3.2f\n" % (dir))

    # Sleep for 1 s
    sleep(1)
```

Програма доволі проста. За допомогою `hmc = hmc5883l.Hmc5883l(0)` створюємо змінну `hmc`, що описуватиме електронний компас. Магнітне поле в різних куточках Землі дещо викривлюється і для корекції отриманих значень необхідно виконати `hmc.set_declination(0.3)` Детальніше про визначення цього параметру можна почитати тут:

<http://www.meccanismocomplesso.org/en/arduino-magnetic-magnetic-magnetometer-hmc5883l>

На кожній ітерації нескінченного циклу зчитуємо покази датчика HMC5883L з використанням `hmc.update()`. За допомогою `pos = hmc.coordinates()` зчитуємо напруженості магнітного поля Землі по X, Y, Z в масив `pos`. За допомогою `dg = hmc.heading()` зчитуємо кут між напрямком на північ і напрямком на який вказує вісь x компасу. Далі виводимо одержані значення на консоль модуля Linkit Smart 7688, робимо затримку на 1 секунду і переходимо до наступної ітерації нескінченного циклу.

Завантажте програму `i2c_compass.py` у модуль і запустіть її. Переконайтеся, що при повороті макетної плати змінюється кут між напрямком на північ і напрямком на який вказує вісь x компасу (Heading).

```
root@mylinkit:~# python i2c_compass.py
Coord: 359 -98 -396
Heading: 1.92
Direction: 0.03

Coord: 359 -96 -398
Heading: 2.22
Direction: 0.04

Coord: 357 -97 -399
Heading: 1.99
Direction: 0.03
```

Для людей з досвідом у програмуванні й електроніці є завдання підвищеної складності. З датчиком HMC5883L можна працювати не лише за допомогою бібліотеки UPM, а напряду через I2C (бібліотека MRAA дозволяє працювати напряду з I2C). Ви можете зчитувати і записувати дані в регістри датчику, підключеного до модуля за допомогою інтерфейсу I2C.

Програмний код для зчитування показів датчику HMC5883L напряду через інтерфейс I2C з програми на Python наведений нижче:

```
i2c_compass2.py x
import mraa
from math import atan2
from time import sleep

def toSigned(val):
    if (val >= 32768):
        return val - 65535 + 1
    else:
        return val

i2c = mraa.I2c(0)

i2c.address(0x1e)

i2c.writeByte(0x02)
i2c.writeByte(0x00)

while True:

    i2c.writeByte(0x03)
    x = i2c.read(2)
    xval = toSigned(x[0] << 8 | x[1]);

    i2c.writeByte(0x05)
    z = i2c.read(2)
    zval = toSigned(z[0] << 8 | z[1])

    i2c.writeByte(0x07);
    y = i2c.read(2);
    yval = toSigned(y[0] << 8 | y[1])

    direction = atan2(yval, xval) + 0.31;

    heading = direction * 180/3.14;

    if (heading < 0):
        heading += 360.0;

    print "x=%d" % ( xval)
    print "y=%d" % ( yval )
    print "z=%d" % ( zval)
    print "direction = %d" % (direction)
    print "heading = %d" % (heading)

    sleep(1)
```

Спершу треба задати адресу датчику на шині I2C (це константа 0x1E) за допомогою **i2c.address(0x1e)**. Щоб записати дані в певний регістр модуля, необхідно спершу записати у модуль номер регістру i, одразу після цього, значення регістру. Наприклад, для того, щоб перевести датчик HMC5883L в режим постійного зчитування необхідно записати 0 в регістр з адресою 0x02. Для цього необхідно виконати:

i2c.writeByte(0x02)

i2c.writeByte(0x00)

Щоб зчитати дані з регістру модуля, необхідно спершу записати у модуль номер регістру і, одразу після цього, зчитати значення регістру. Оскільки дані про напруженість магнітного поля займають 16 біт, необхідно зчитувати 2 байти підряд. Дані про напруженість магнітного поля Землі по осі X містяться в регістрі модуля з адресою 0x03. Дані про напруженість магнітного поля Землі по осі Z містяться в регістрі модуля з адресою 0x05. Дані про напруженість магнітного поля Землі по осі Y містяться в регістрі модуля з адресою 0x07.

Більш детально регістри датчику HMC5883L описані в документації:

https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf

Щоб зчитати 2 байти, що характеризують напруженість магнітного поля Землі по осі X необхідно виконати:

```
i2c.writeByte(0x03)
```

```
x = i2c.read(2)
```

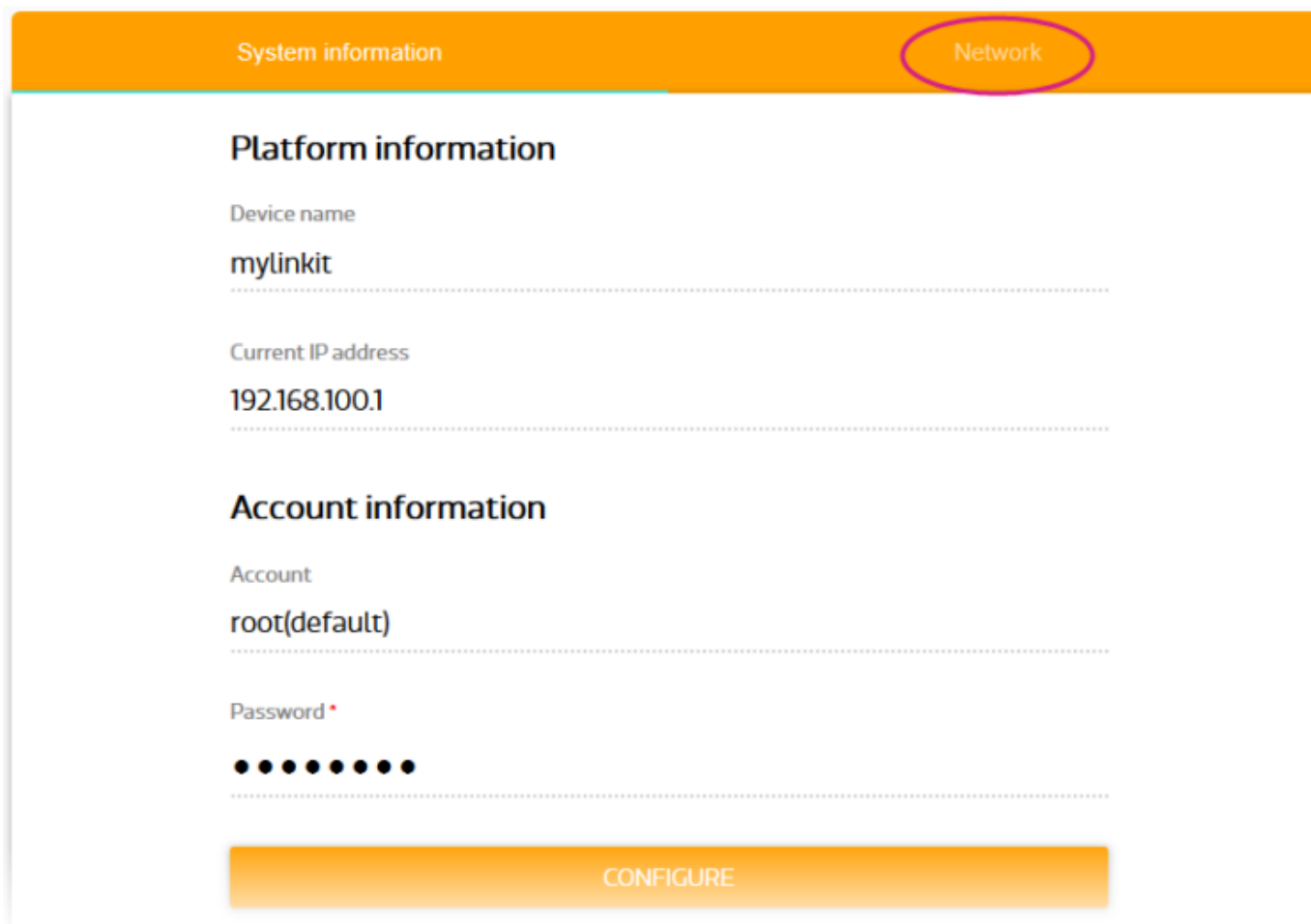
Завантажте програму **i2c_compass2.py** у модуль і запустіть її. Переконайтеся, що вона працює так само, як і попередня програма **i2c_compass.py**.

Підключення до WiFi роутера та мережі Internet

Поки що ви працювали з модулем в режимі точки доступу. Linkit Smart 7688 був точкою доступу, до якої ви підключалися зі свого комп'ютера. Часто виникає ситуація, коли необхідно підключити модуль до існуючого WiFi роутера. Це може бути необхідно для обміну даними по WiFi між кількома модулями та для підключення до Internet.

Для переведення модуля Linkit Smart 7688 з режиму точки доступу в режим WiFi станції, щоб він міг підключитися до WiFi роутера, необхідно виконати наступні дії:

1. Виконайте WiFi підключення до модуля зі свого комп'ютеру, доки модуль ще працює в режимі точки доступу.
2. Зайдіть у веб інтерфейс керування модулем (адреса у браузері: <http://mylinkit.local>) та натисніть кнопку **Network** у правому верхньому куті.



3. Оберіть **Station Mode** та **WiFi** мережу до якої модуль Linkit Smart 7688 буде підключатися після завантаження (наприклад, це може бути WiFi мережа Belka). Якщо мережа захищена паролем, встановіть пароль до WiFi мережі (якщо мережа відкрита, пароль встановлювати непотрібно). Натисніть на кнопку **CONFIGURE & RESTART**. Це приведе до збереження налаштувань і перезавантаження модуля.

The screenshot shows the 'Network' tab of the configuration interface. The 'Station mode' radio button is selected. A detected Wi-Fi network named 'mtkemp (83%)' is listed. A 'REFRESH' button is positioned below the network name. A 'Password' field is visible with a 'SHOW PASSWORD' link. At the bottom, the 'CONFIGURE & RESTART' button is highlighted with a red circle.

Після перезавантаження модуль уже не буде працювати, як точка доступу. Натомість Linkit Smart 7688 буде намагатися підключитися до WiFi мережі, яку ви щойно обрали.

Якщо до мережі немає доступу, або ви вказали невірний пароль, модуль не зможе підключитися до заданої WiFi мережі. В такому разі для підключення до модуля по WiFi необхідно перевести його назад в режим точки доступу. Як це зробити показано в одному з наступних розділів.

Якщо Linkit Smart 7688 успішно підключився до заданої WiFi мережі, на своєму комп'ютері ви теж можете підключитися до цієї ж WiFi мережі і для доступу до модуля використовувати уже знайому адресу mylinkit.local

Переведення модуля в режим точки доступу

Припустимо, ви налаштували модуль Linkit Smart 7688 на підключення до певної WiFi мережі, яка зараз недоступна. Щоб підключитися до веб інтерфейсу керування модулем, або до терміналу модуля через SSH вам необхідно перевести модуль в режим точки доступу і підключитися до цієї точки доступу.

Для переведення модуля Linkit Smart 7688 в режим точки доступу вам необхідно після завантаження модуля натиснути кнопку біля якої написано WiFi і утримувати її більше 5 секунд, але менше 20-ти секунд (якщо утримувати цю кнопку 20 і більше секунд відбудеться зкидання налаштувань модуля до стандартних і видалення створених і завантажених вами файлів).

Відео інструкція по переведенню модуля в режим точки доступу:

<https://www.youtube.com/watch?v=GZ48q2xxflw>

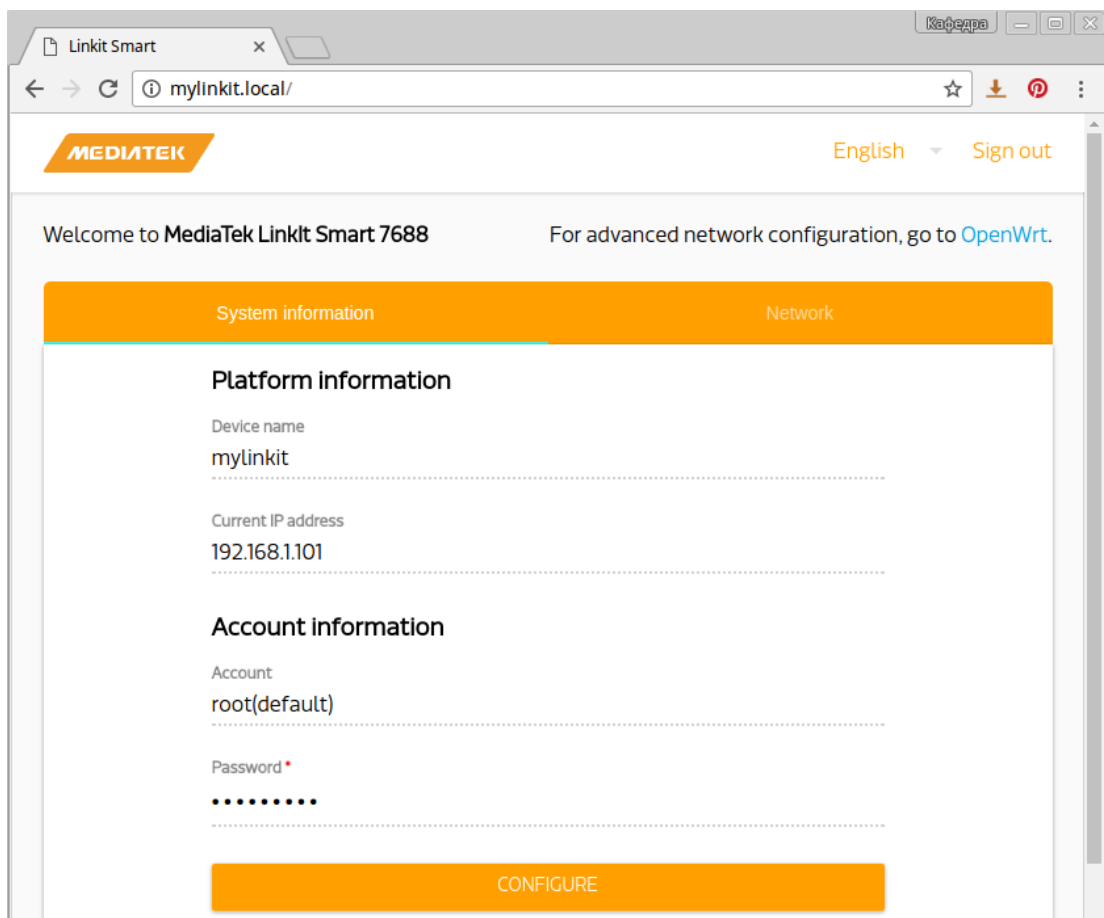
Встановлення нового імені (dns адреси) модуля

Коли модуль Linkit Smart 7688 працював в режимі точки доступу для підключення до нього ми використовували адресу <http://mylinkit.local>

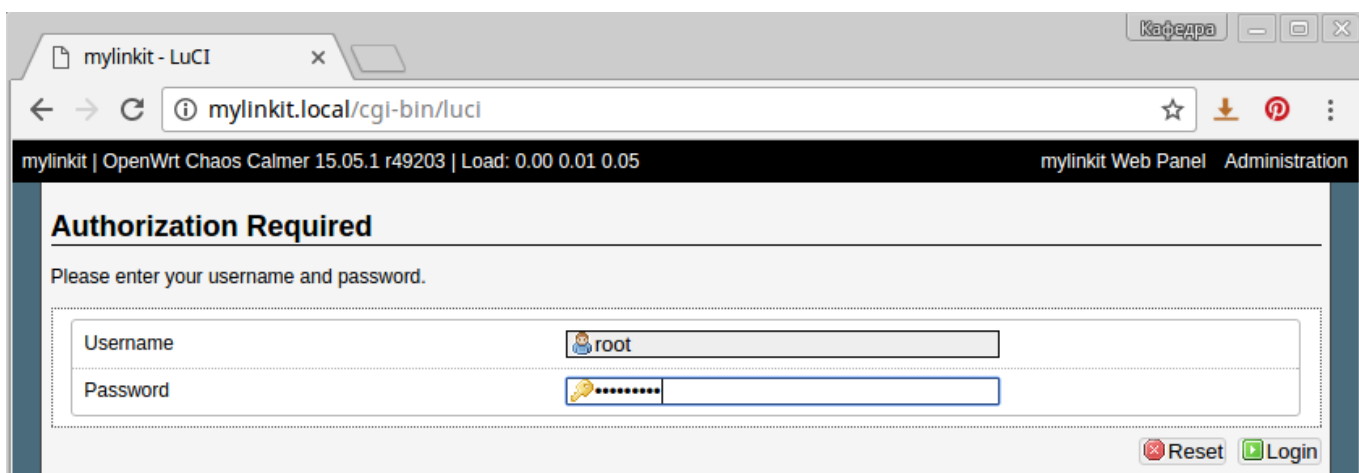
Якщо в одній мережі будуть присутні кілька модулів Linkit Smart 7688 підключених до одного WiFi роутера, кожному модулю необхідно буде встановити унікальне ім'я.

Для того, щоб задати унікальне ім'я для Linkit Smart 7688 виконайте наступні кроки:

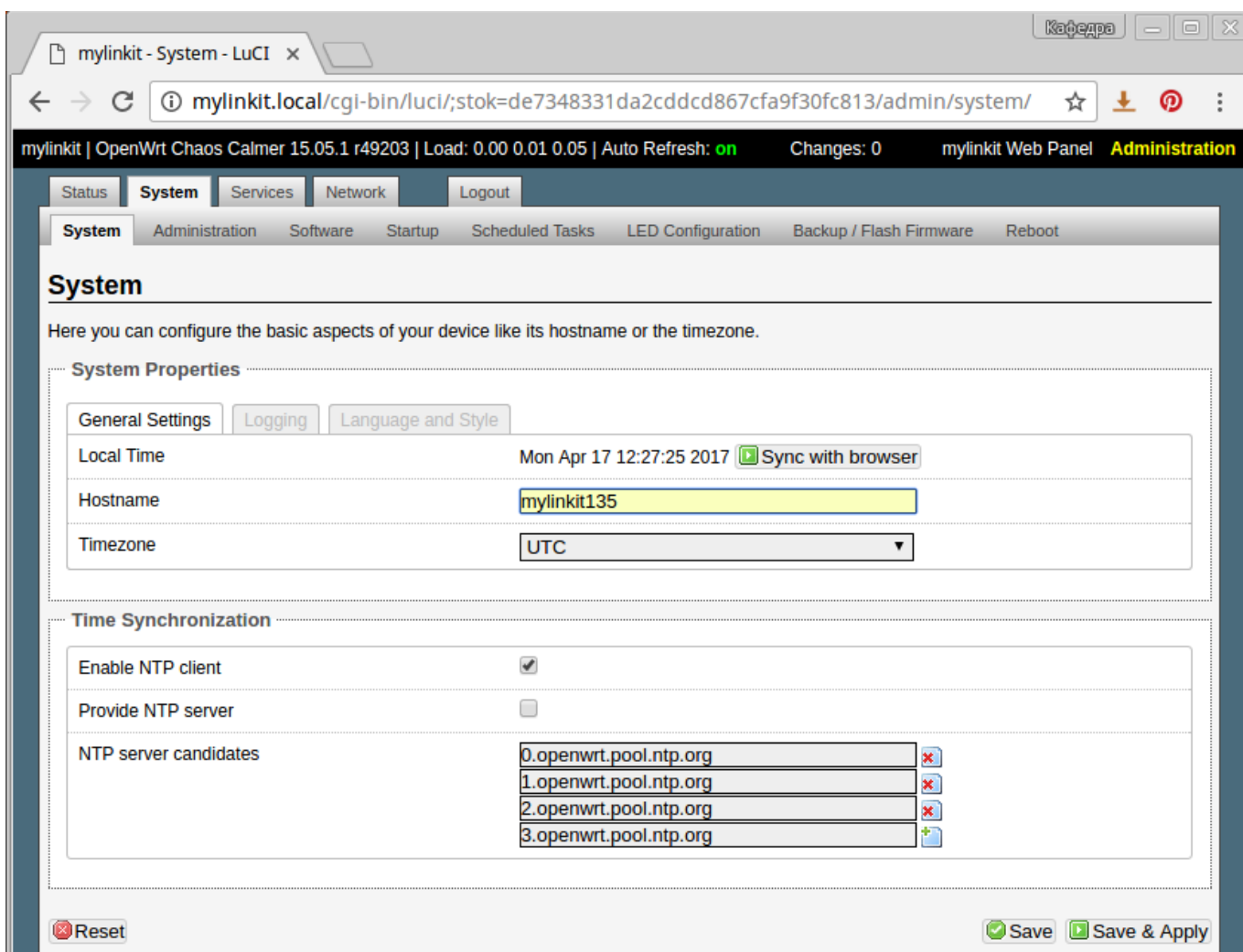
1. Поки модуль ще знаходиться в режимі точки доступу підключіться до цієї точки доступу по WiFi і зайдіть у веб інтерфейс керування за адресою <http://mylinkit.local>
2. Для переходу до адміністративної панелі OpenWrt натисніть на посилання OpenWrt у верхньому правому куті.



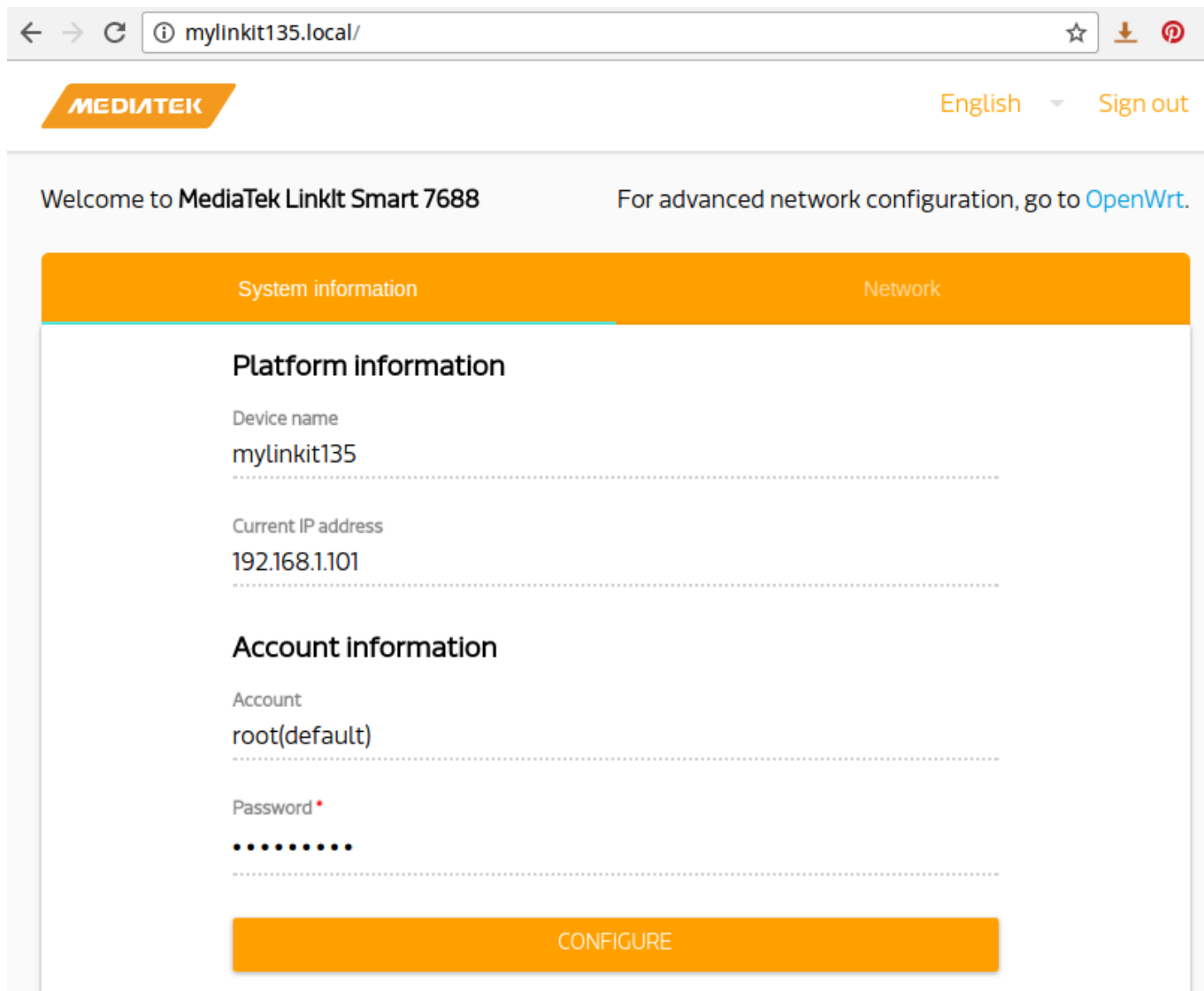
3. Введіть пароль, який ви раніше використовували для доступу до веб інтерфейсу керування та натисніть кнопку **Login**



4. Натисніть на вкладку **System**. В полі **Hostname** введіть нове ім'я модуля, наприклад mylinkit135 (старі ім'я було mylinkit). Натисніть кнопку **Save&Apply** для збереження налаштувань.



5. Перезавантажте модуль (з веб інтерфейсу керування, або відключивши і знову підключивши USB кабель). Після перезавантаження нова адреса модуля буде <http://mylinkit135.local>



The screenshot shows a web browser window with the address bar containing `mylinkit135.local/`. The page header features the MediaTek logo on the left, and language selection (English) and a sign-out link on the right. The main content area is titled "Welcome to MediaTek LinkIt Smart 7688" and includes a link to "OpenWrt" for advanced network configuration. Below this is a navigation bar with two tabs: "System information" (active) and "Network". The "System information" tab displays the following details:

- Platform information**
 - Device name: mylinkit135
 - Current IP address: 192.168.1.101
- Account information**
 - Account: root(default)
 - Password: [masked]

A large orange button labeled "CONFIGURE" is positioned at the bottom of the system information section.

Бонусна робота. Створення веб серверу на Python

В цьому прикладі ми створимо веб сервер на Python, будемо відображати кілька веб сторінок (спершу простішу, а потім більш складну), а також виводити на веб сторінку результати вимірювання електронного компаса.

Спершу вам необхідно налаштувати модуль Linkit Smart 7688 на підключення до WiFi роутеру з якого є доступ в мережу Інтернет. Як це зробити розказано в одному з попередніх розділів, що називається “Підключення до WiFi роутеру та мережі Internet”.

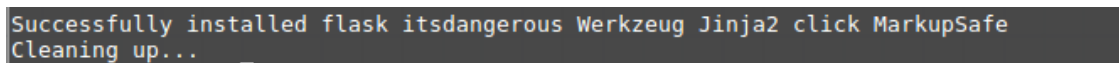
Після підключення до мережі Інтернет для встановлення веб серверу Flask (написаного на Python) виконайте з консолі модуля Linkit Smart 768 команду

pip install flask



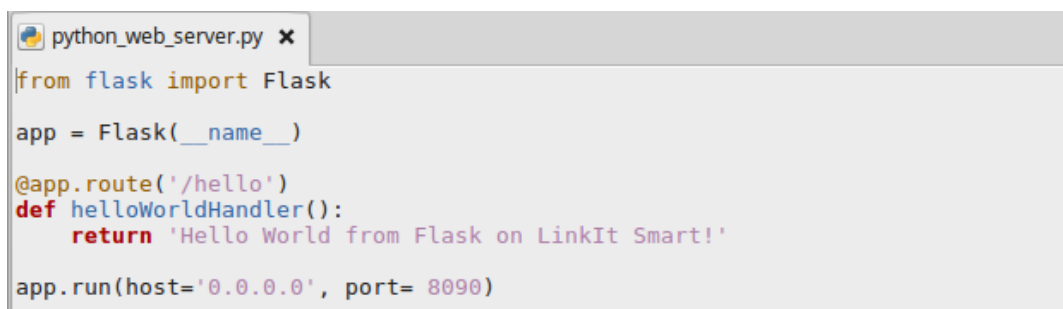
```
WIRELESS FREEDOM
-----
CHAOS CALMER (15.05.1, r49203)
-----
* 1 1/2 oz Gin           Shake with a glassful
* 1/4 oz Triple Sec     of broken ice and pour
* 3/4 oz Lime Juice     unstrained into a goblet.
* 1 1/2 oz Orange Juice
* 1 tsp. Grenadine Syrup
-----
root@mylinkit:~# pip install flask
Downloading/unpacking flask
  Downloading Flask-0.12.1-py2.py3-none-any.whl (82kB): 82kB downloaded
Downloading/unpacking itsdangerous>=0.21 (from flask)
  Downloading itsdangerous-0.24.tar.gz (46kB): 46kB downloaded
Running setup.py (path:/tmp/pip_build_root/itsdangerous/setup.py) egg_info for package itsdangerous
```

Встановлення Flask потребує кілька хвилин. Після завершення встановлення на консоль модуля буде виведена наступна інформація:



```
Successfully installed flask itsdangerous Werkzeug Jinja2 click MarkupSafe
Cleaning up...
```

На своєму комп'ютері створіть текстовий файл з іменем “python_web_server.py”. Додайте у файл наступний код на Python:



```
python_web_server.py x
from flask import Flask

app = Flask(__name__)

@app.route('/hello')
def helloWorldHandler():
    return 'Hello World from Flask on LinkIt Smart!'

app.run(host='0.0.0.0', port= 8090)
```

Програмний код дуже простий. Спершу ми підключаємо модуль веб серверу Flask за допомогою **from flask import Flask**. Далі створюємо веб сервер за допомогою **app = Flask(__name__)**. Тепер змінна **app** описує створений веб сервер. Зверніть увагу, що ім'я веб серверу задаємо за допомогою вбудованої константи **__name__**.

За допомогою

```
@app.route('/hello')
```

```
def helloWorldHandler():
```

```
    return 'Hello World from Flask on LinkIt Smart!'
```

створюємо функцію, що викликається при спробах доступу до веб сторінки з іменем **“/hello”**. Працює це наступним чином. Коли ви пишете адресу веб сторінки у браузері, браузер підключається до веб серверу і просить надати html код цієї веб сторінки (для її відображення у вікні браузеру). В такому випадку веб сервер Flask викликає функцію, яку ми задали для веб сторінки з адресою **“/hello”** і ця функція повинна повернути html код веб сторінки (у вигляді строки). Цей html код і передається браузеру. В нашому випадку повертається строка **Hello World from Flask on LinkIt Smart!**, яка і буде відображатися у вікні браузера.

Врешті решт запускаємо створений веб сервер:

```
app.run(host='0.0.0.0', port= 8090)
```

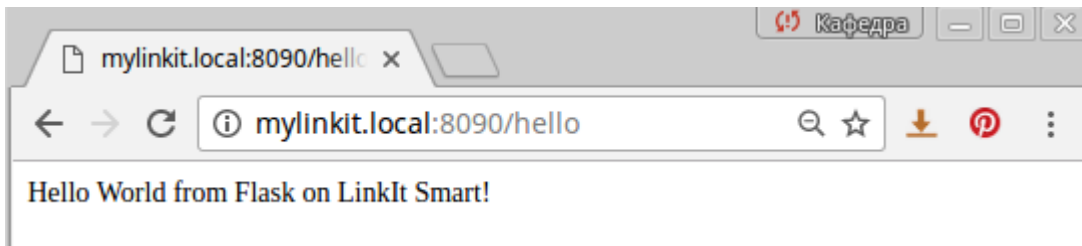
Для підключення до веб серверу будемо використовувати порт **8090**, оскільки стандартний порт **80** вже зайнятий веб сервером сторінки налаштувань модуля.

Завантажте програму **python_web_server.py** у модуль і запустіть її. Відкрийте браузер у себе на комп'ютері, напишіть у полі адреси <http://mylinkit.local:8090/hello> і натисніть **Enter**.

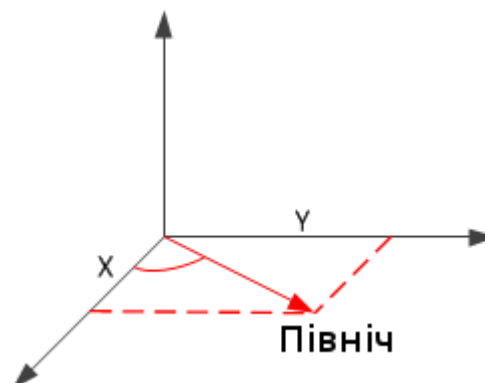
На консоль модуля одразу ж виведеться інформація про те, що браузер з певної IP адреси (в даному прикладі це адреса 192.168.1.100) зчитує веб сторінку веб сервера з адресою **“/hello”**

```
root@mylinkit:~# python python_web_server.py
* Running on http://0.0.0.0:8090/ (Press CTRL+C to quit)
192.168.1.100 - - [23/Apr/2017 12:14:21] "GET /hello HTTP/1.1" 200 -
192.168.1.100 - - [23/Apr/2017 12:14:21] "GET /favicon.ico HTTP/1.1" 404 -
```

А у вікні браузера відобразиться строка **Hello World from Flask on LinkIt Smart!**, яку ми повернули з функції **helloWorldHandler()**



Тепер ускладнимо задачу і створимо більш серйозну веб сторінку, що буде відображати результати зчитані з електронного компасу hmc5883l, а саме -- виводити кут між напрямком на північ і напрямком на який вказує вісь x компасу.



На своєму комп'ютері створіть текстовий файл з іменем "python_web_server2.py". Додайте у файл наступний код на Python:

```
python_web_server2.py x
from flask import Flask
import pyupm_hmc5883l as hmc5883l

hmc = hmc5883l.Hmc5883l(0)
hmc.set_declination(0.3)

app = Flask(__name__)

@app.route('/hello')
def helloWorldHandler():
    hmc.update()
    heading = int(hmc.heading())

    page = "<html> "
    page = page + "<head> "
    page = page + "<meta http-equiv='refresh' content='1' /> "
    page = page + "<title>Linkit 7688 Demo</title> "
    page = page + "<style> "
    page = page + "body { background-color: #cccccc; font-family: Arial, Helvetica, Sans-Serif; Color: #000088; } "
    page = page + "</style>"
    page = page + "</head>"
    page = page + "<body>"
    page = page + "<h1>Hello from Linkit 7688!</h1>"
    page = page + "<p>Compass heading: " + str(heading) + " deg </p>"
    page = page + "</body>"
    page = page + "</html>"

    return page

app.run(host='0.0.0.0', port= 8090)
```

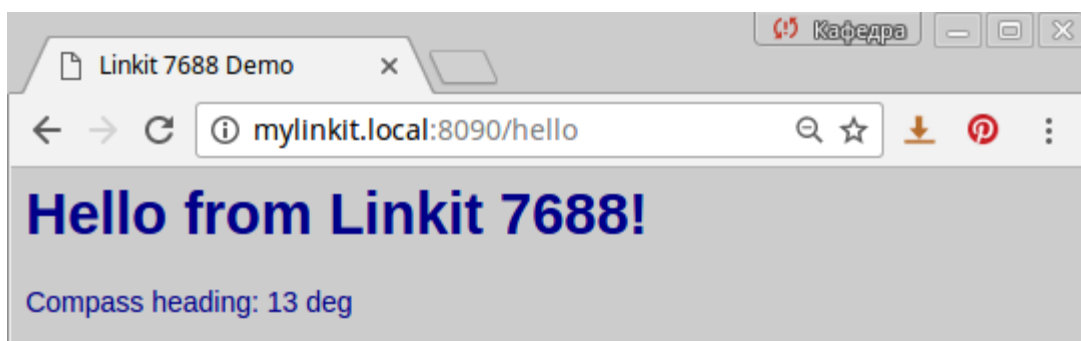

Підключіть до модуля Linkit Smart 7688 електронний компас hmc5883l. Як це зробити показано в **роботі №7** даного документу. Програмний код для роботи з електронним компасом за допомогою бібліотеки UPM описано в **роботі №7** даного документу.

Щойно створена програма працює досить просто. Коли браузер запитує від веб серверу веб сторінку з адресою **"/hello"**, викликається функція **helloWorldHandler**, що повертає веб серверу html код веб сторінки у вигляді строки. В даному прикладі ми вже використовуємо повноцінний html для опису веб сторінки. Для автоматичного перезавантаження сторінки кожну секунду використовуємо `<meta http-equiv='refresh' content='1'/>`. Кожного разу перед тим, як повернути код веб сторінки, ми вимірюємо за допомогою електронного компасу кут між напрямком на північ і напрямком на який вказує вісь x компасу і вставляємо цей кут в HTML код веб сторінки:

```
page = page + "<p>Compass heading: " + str(heading) + " deg </p>".
```

Таким чином ви можете виводити на веб сторінку інформацію одержану з будь яких датчиків.

Завантажте програму **python_web_server2.py** у модуль і запустіть її. Відкрийте браузер у себе на комп'ютері, напишіть у полі адреси <http://mylinkit.local:8090/hello> і натисніть **Enter**. Переконайтеся, що кут між напрямком на північ і напрямком на який вказує вісь x компасу виводиться на веб сторінку і веб сторінка оновлюється кожну секунду.



Визначення IP адреси модуля

IP адресу модуля Linkit Smart 7688 можна визначити двома шляхами:

1. Переглянути у веб інтерфейсі керування:

Platform information

Device name

mylinkit135

Current IP address

192.168.1.101

2. Підключитися до терміналу модуля через ssh і написати в консолі **ifconfig**
IP адреса модуля знаходиться справа біля напису arcli0. Для даного прикладу IP адреса буде 192.168.1.101

```
root@mylinkit135:~# ifconfig
arcli0  Link encap:Ethernet  HWaddr 9E:65:F9:0E:76:88
        inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::9c65:f9ff:fe0e:7688/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

br-lan  Link encap:Ethernet  HWaddr 9C:65:F9:1E:68:DD
        inet addr:192.168.100.1  Bcast:192.168.100.255  Mask:255.255.255.0
        inet6 addr: fd28:7ece:7856::1/60  Scope:Global
        inet6 addr: fe80::9e65:f9ff:fe1e:68dd/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:135 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 B)  TX bytes:20461 (19.9 KiB)

eth0    Link encap:Ethernet  HWaddr 9C:65:F9:1E:68:DD
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:135 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:23630 (23.0 KiB)
        Interrupt:5
```

Оновлення прошивки модуля

Всередині модуля Linkit Smart 7688 виконується операційна система Linux OpenWrt. Періодично з'являються нові версії цієї операційної системи з новими можливостями програм.

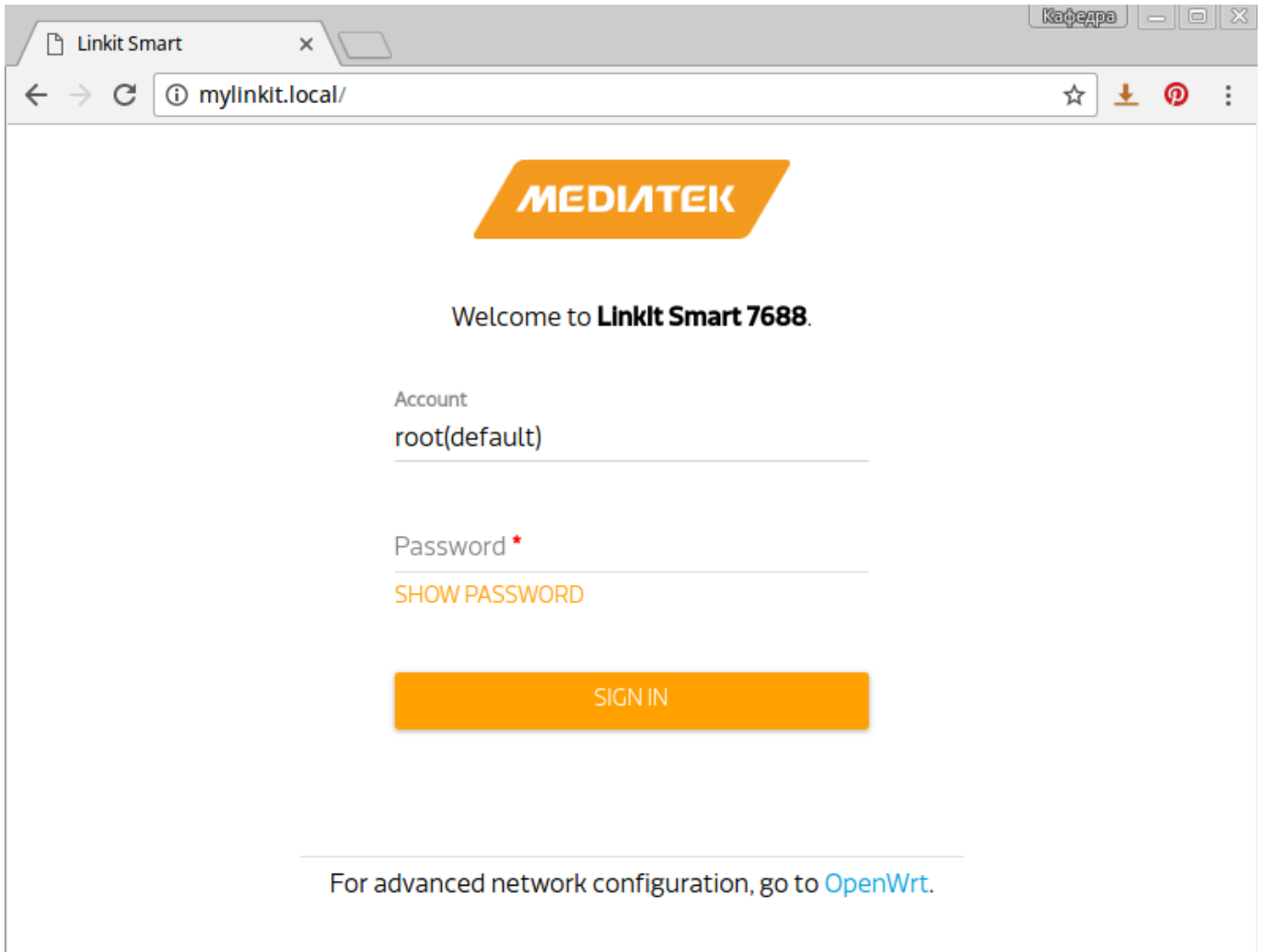
Для оновлення версії Linux OpenWrt на модулі Linkit Smart 7688 виконайте наступні кроки:

1. Завантажте найновішу версію відповідної прошивки з сайту виробника модуля (файл **lks7688.img**):

<https://docs.labs.mediatek.com/resource/linkit-smart-7688/en/downloads>

Станом на 17 квітня 2017 року найновішою є версія 0.94

2. Підключіть модуль до USB порту комп'ютера
3. Якщо модуль працює в режимі точки доступу, підключіться до цієї точки доступу (LinkIt_Smart_7688_1E7688). Якщо модуль працює в режимі WiFi станції, значить ви вже налаштували його для підключення до певної WiFi мережі.
4. Підключіться до веб сторінки керування модулем за допомогою браузера (адреса: mylinkit.local)



5. Натисніть на кнопку UPGRADE FIRMWARE

Software information

Boot loader version

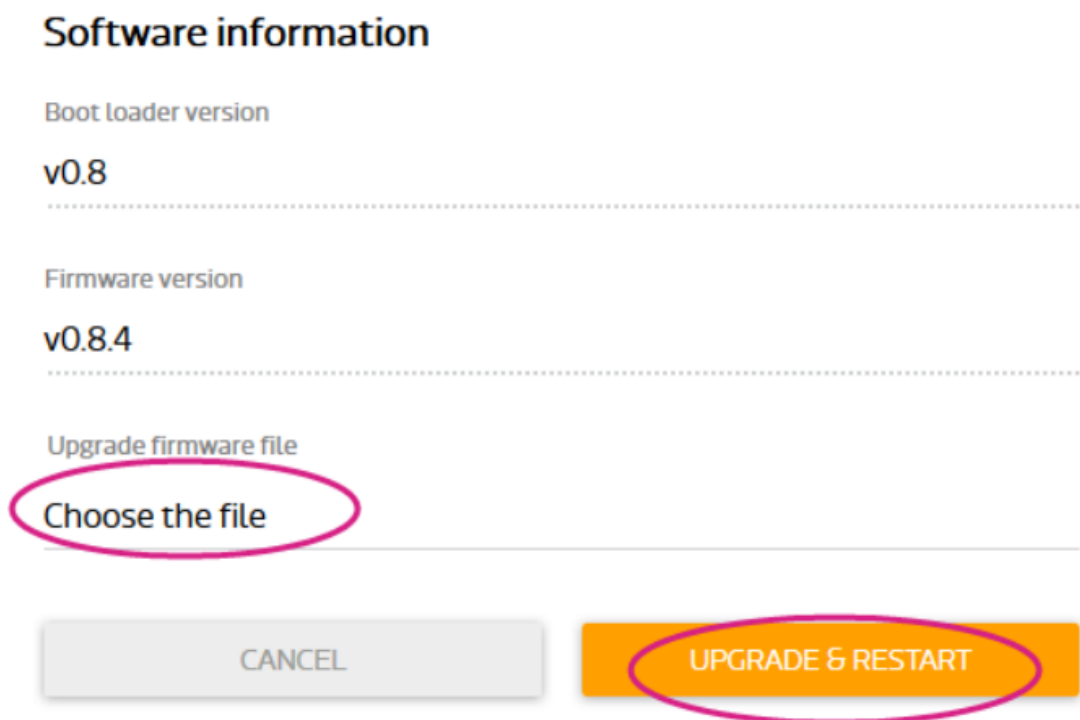
v0.8

Firmware version

v0.8.4

UPGRADE FIRMWARE

6. За допомогою кнопки **Choose the file** оберіть файл **lks7688.img** та натисніть на кнопку **UPGRADE & RESTART**. Нова прошивка почне завантажуватись в модуль. Ні в якому разі не відключайте живлення (USB кабель) від модуля до завершення процедури прошивки! Перепрошивка триватиме близько 3-х хвилин. До завершення процедури прошивки червоний світлодіод швидко блиматиме. Після перепрошивки нова версія Linux OpenWrt почне завантажуватись. Протягом завантаження (30 секунд) червоний світлодіод буде світитися, а після завантаження OpenWrt червоний світлодіод погасне.



7. Після завершення прошивки ви можете знову підключитися до веб інтерфейсу керування модулем і пересвідчитись, що номер версії прошивки змінився на новий.

Відео інструкція до процесу оновлення прошивки модуля:

<https://www.youtube.com/watch?v=GtmbgN0VbyM>

Відновлення стандартних налаштувань модуля

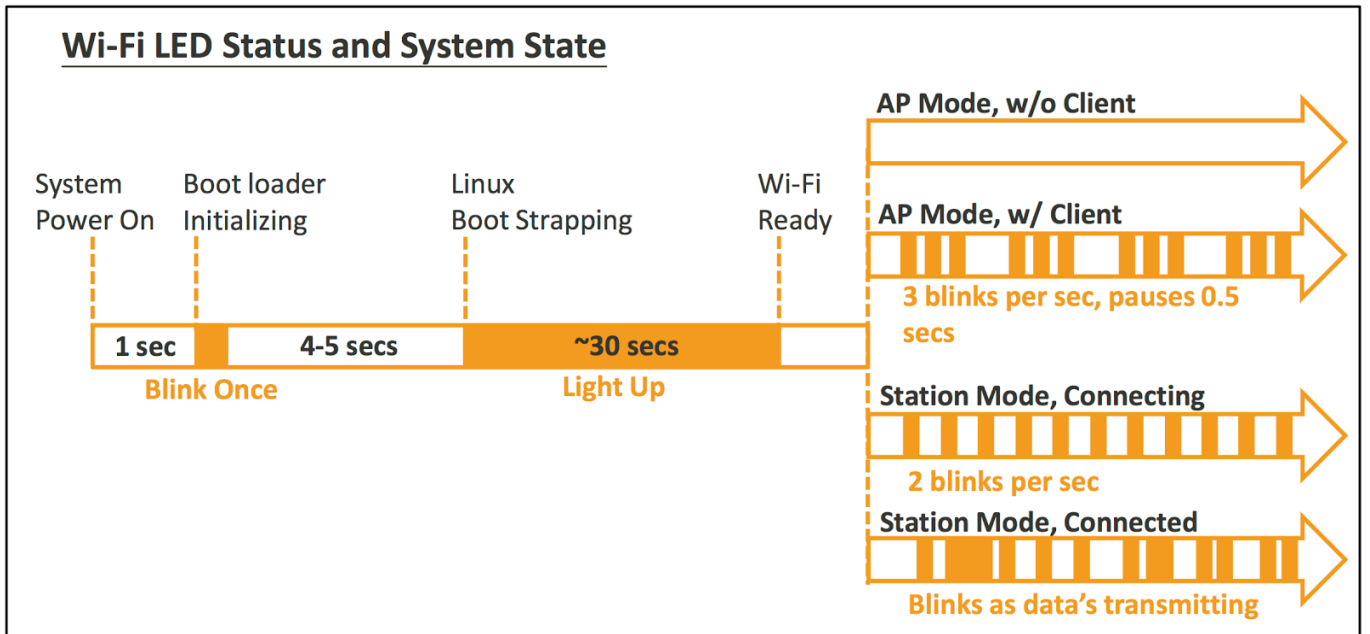
Іноді можлива ситуація, що ви забули пароль, який встановили раніше, або налаштували модуль для підключення до певної WiFi мережі, яка зараз недоступна. В такому випадку може допомогти відновлення налаштувань модуля до стандартних. Важливо розуміти, що після такого процесу всі налаштування встановлені вами раніше і всі завантажені/створені файли (програми та скрипти) будуть автоматично видалені.

Для відновлення налаштувань модуля необхідно після його завантаження натиснути кнопку біля якої написано **WiFi** і утримувати її натисненою не менше 20 секунд, після чого кнопку необхідно відпустити. Після цього почнеться процес відновлення налаштувань протягом якого червоний світлодіод швидко блиматиме кілька секунд. Після завершення відновлення модуль перезавантажиться. Протягом перезавантаження (30 секунд) червоний світлодіод буде світитися, а після завантаження OpenWrt червоний світлодіод погасне. Тепер параметри прошивки стандартні, модуль працює в режимі точки доступу, ви можете підключитися до нього і задати необхідні налаштування (новий пароль і т.д.).

Відео інструкція процесу зкидання:

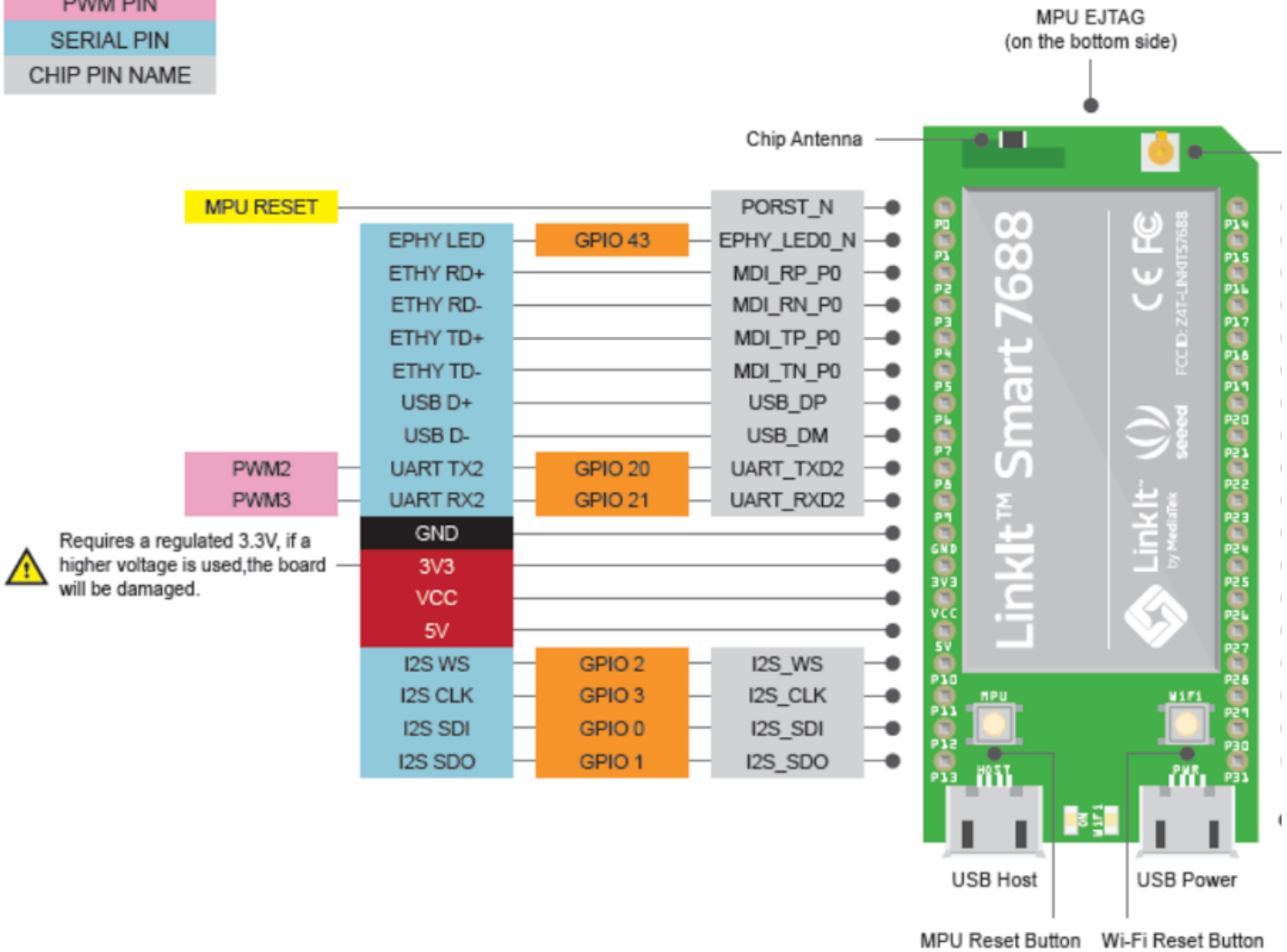
<https://www.youtube.com/watch?v=Y0r5c0mwm5Y>

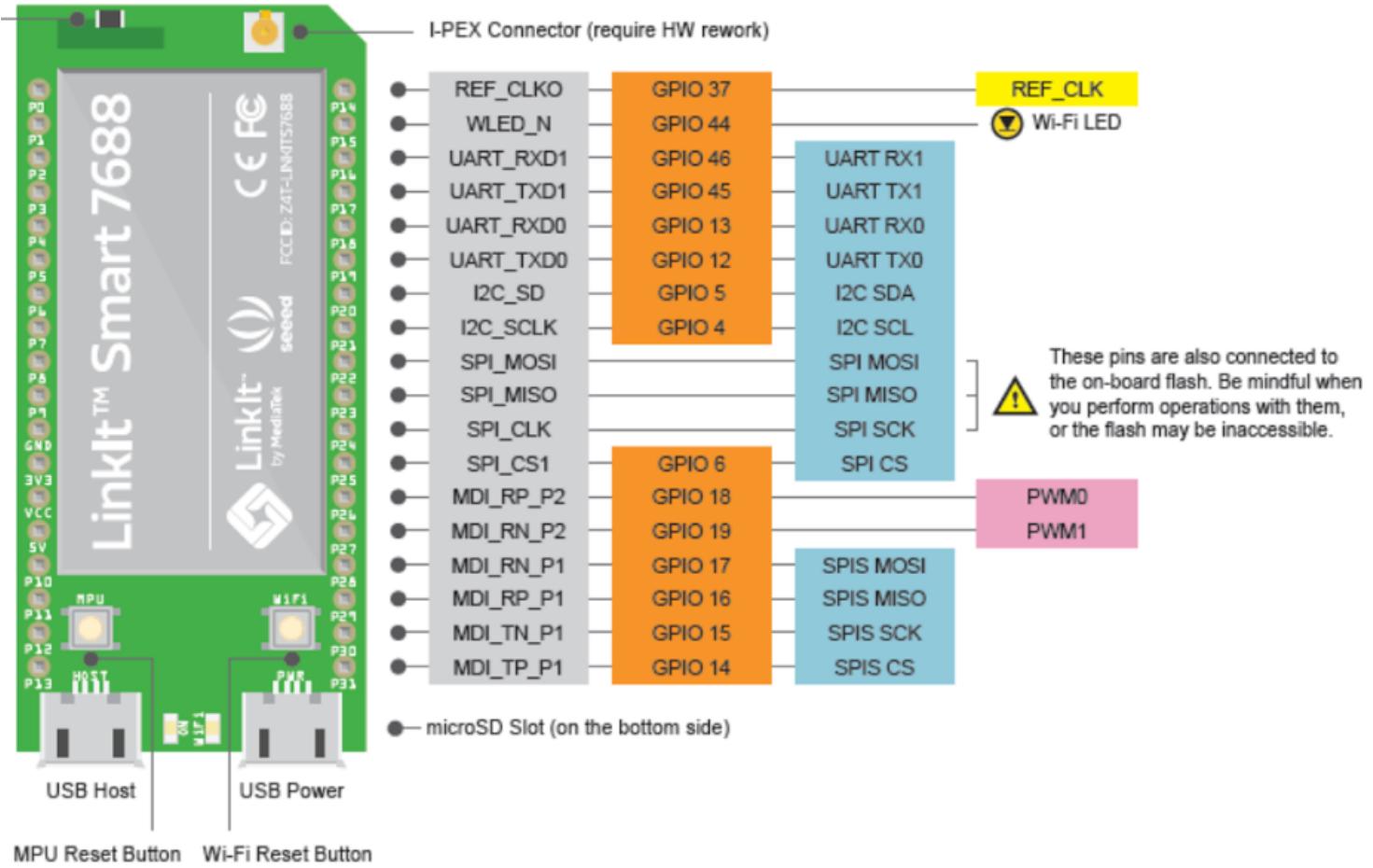
Індикація режимів роботи модуля за допомогою червоного світлодіода



Опис ліній введення-виведення модуля

GND
POWER
CONTROL
DIGITAL PIN
PWM PIN
SERIAL PIN
CHIP PIN NAME





Посилання

1. Опис і початок роботи (рос.):

<https://www.kv.by/post/1048802-sobstvennoe-ustroystvo-dlya-inter-neta-veshchey-s-mediatek-linkit-smart-7688-duo>

2. Початок роботи (англ.):

<https://goo.gl/yQDVVH>

3. Приклад програмування на Python:

<https://docs.labs.mediatek.com/resource/linkit-smart-7688/en/tutorials/peripheral-support-on-linkit-smart-7688-development-board/using-mraa-in-python>

4. Приклад роботи з датчиками з використанням бібліотеки UPM:

<https://docs.labs.mediatek.com/resource/linkit-smart-7688/en/tutorials/peripheral-support-on-linkit-smart-7688-development-board/using-upm-in-python>

5. Перелік датчиків, що підтримуються бібліотекою Python UPM:

<https://iotdk.intel.com/docs/master/upm/python/modules.html>

6. Приклад програмування на NodeJS:

<https://docs.labs.mediatek.com/resource/linkit-smart-7688/en/tutorials/peripheral-support-on-linkit-smart-7688-development-board/using-mraa-in-node-js>

7. Інструкції до програмування на C:

<https://docs.labs.mediatek.com/resource/linkit-smart-7688/en/tutorials/c-c++-programming>

8. Developers guide:

<https://labs.mediatek.com/en/download/ih80Qtjo>

9. Запуск веб серверу на Python:

<https://techtutorialsx.wordpress.com/2016/12/27/linkit-smart-duo-running-a-flask-server/>

10. Приклади роботи з хмарою на NodeJS і Python:

<https://mcs.mediatek.com/7688/>

https://mcs.mediatek.com/resources/latest/tutorial/7688_led_tutorial

11. Зкидання налаштувань модуля до стандартних (включаючи зкидання паролю):

<https://www.youtube.com/watch?v=Y0r5c0mwm5Y>

12. Оновлення прошивки модуля:

<https://www.youtube.com/watch?v=GtmbgN0VbyM>

13. Підключення до терміналу OpenWrt за допомогою usb-uart перетворювача:

<https://docs.labs.mediatek.com/resource/linkit-smart-7688/en/tutorials/firmware-and-bootloader/bootloader-and-kernel-console>

14. Переведення модуля в режим точки доступу:

<https://www.youtube.com/watch?v=GZ48q2xxflw>

15. Лінії введення-виведення модуля

http://www.cnx-software.com/wp-content/uploads/2015/12/Link_Smart_7688_Pinout.png

g