



Quick Note QN28

**QN How to make Leds show ETH and Eroute
status on TransPort WR21**

18 November 2015

QN How to make Leds show ETH and Eroute status on TransPort WR21

Contents

1	Introduction	3
1.1	Outline	3
1.2	Corrections	3
1.3	Version	3
2	Writing The Python script	4
3	Running the Python script	6
3.1	Load the script on the TransPort via WEB GUI.....	6
3.2	Load the script on the TransPort via FTP	7
3.3	Run and auto-run the script.....	8
4	Testing LEDs behavior.....	10
4.1	ETH 0 UP & VPN UP.....	10
4.2	ETH 0 DOWN & VPN DOWN.....	11
4.3	ETH 0 UP & VPN DOWN.....	12
5	Python Script.....	13

1 INTRODUCTION

1.1 Outline

It can be useful for monitoring purposes, to use LEDs to check some status on the TransPort routers. This QN gives the instructions to write and use a simple Python script on TransPort WR21 in order to have LEDs reflecting the status of ETH interface and an IPsec VPN (Eroute 0).

1.2 Corrections

Requests for corrections or amendments to this application note are welcome and should be addressed to: tech.support@digicom.com

Requests for new application notes can be sent to the same address.

1.3 Version

Version Number	Status
0.1	Draft

2 WRITING THE PYTHON SCRIPT

The WR21 used in this QN has the following LEDs:



There is no LED indicating the ETH status and also no one for the Eroute0 (VPN tunnel). Assuming that we chose the “Service LED” as Eroute 0 status indicator and “WWAN” as ETH0 status indicator, the following script can be run in order to have those LEDs behaving as expected:

In order to obtain that behavior, the following python script must be run on the device:

```
#####  
#####  
from BaseWizard import write_cli  
import time  
from time import sleep  
  
# Commands to assign the LEDs you want to control to bank 3, the syntax is ledbnk <lednumber> <banknumber>  
# In our case we have set led 2 (WWAN) and 1 (SERVICE) to bank 3  
  
write_cli('ledbnk 2 3')  
write_cli('ledbnk 1 3')  
  
# With while TRUE and TRUE = 1 the script will run forever as soon as it is started  
  
TRUE = 1  
while TRUE :  
  
    #Check status of eth 1  
    result = write_cli('eth 1 status')  
    i = result.find('Connection Status: Connected',0)  
    if i > 0:  
        #LED command to turn ON the appropriate LED, syntax is ledn <lednumber> <ledbank> <ledstate>  
        #In that case we set led 2 (WWAN) to switch ON (statenumber=2)  
  
        write_cli('ledn 2 3 2')  
    else:  
        #LED command to turn OFF the appropriate LED, syntax is ledn <lednumber> <ledbank> <ledstate>  
        #In that case we set led 2 (WWAN) to switch OFF (statenumber=1)  
  
        write_cli('ledn 2 3 1')  
  
    #Check status of Eroute 0  
    result = write_cli('sastat 0 0')  
    i = result.find('SPI Eroute',0)  
    if i > 0:
```

QN How to make Leds show ETH and Eroute status on TransPort WR21

```
#LED command to turn ON the appropriate LED, syntax is ledn <lednumber> <ledbank> <ledstate>
#In that case we set led 1 (service) to switch ON (statenumber=2)
write_cli('ledn 1 3 2')
else:
#LED command to turn OFF the appropriate LED, syntax is ledn <lednumber> <ledbank> <ledstate>
#In that case we set led 1 (service) to switch OFF (statenumber=1)
write_cli('ledn 1 3 1')
```

```
time.sleep(0.25) #Run check every 250 ms
```

```
#=====
```

LEDs used and their behaviour can be modified, considering the following tables.

LED NAME / NUMBER:

LED	LED number
Service	1
WWAN	2
Signal1	3
Signal2	4
Signal3	5

LED state / LED state NUMBER:

LED state	LED state number
OFF	1
ON	2
RAPID FLASH	4
Cadence (long on, short off)	7
Cadence (long off, short on)	9
SLOW FLASH	10

3 RUNNING THE PYTHON SCRIPT

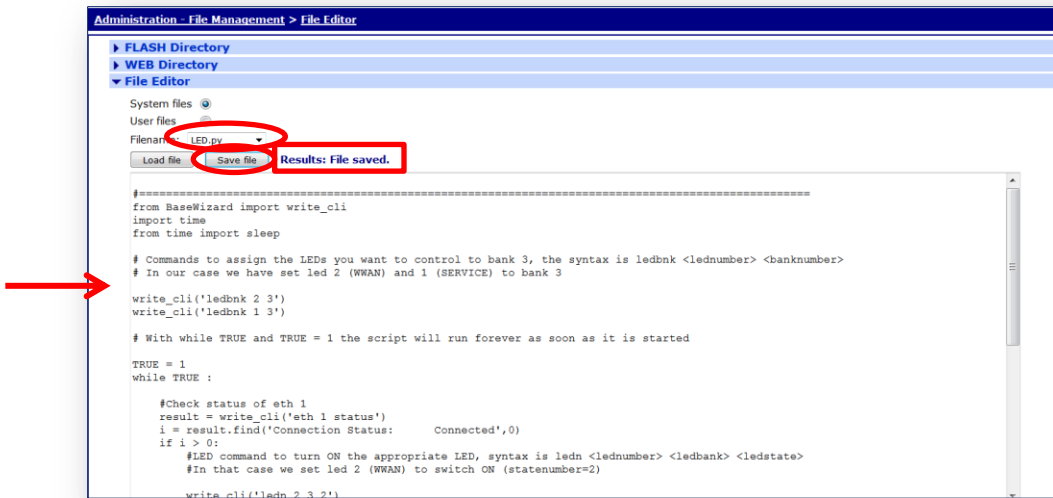
3.1 Load the script on the TransPort via WEB GUI

In the WEB GUI, browse to Administration-File management > File editor, click in the Filename field (where there is "--edit--"):



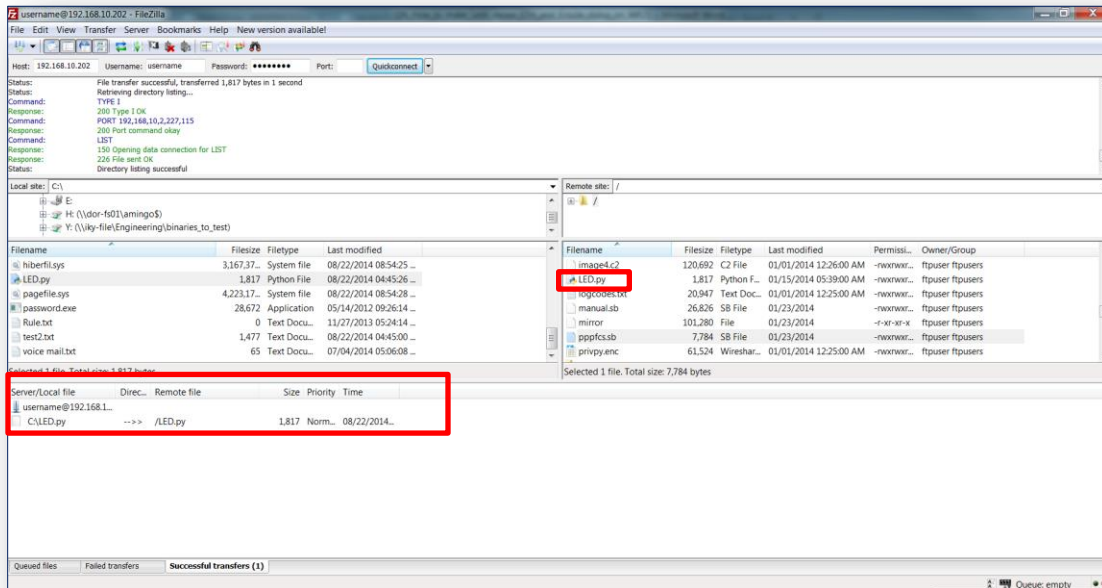
Type a name for the script (for LED.py). Then, in the blank window below, paste the script modified as per the desired scenario. Then click on Save File and then you should see:

QN How to make Leds show ETH and Eroute status on TransPort WR21



3.2 Load the script on the TransPort via FTP

Write the script into a file (a Python script is a text file containing python commands and may be created using a normal plain text editor) and give a name to it (as LED.py) and then transfer it to the TransPort via FTP, for example using an FTP client as Filezilla:



QN How to make Leds show ETH and Eroute status on TransPort WR21

3.3 Run and auto-run the script

In order to have the script running , go in the Administration – Execute command section of the WEB GUI (or in the command line) and type:

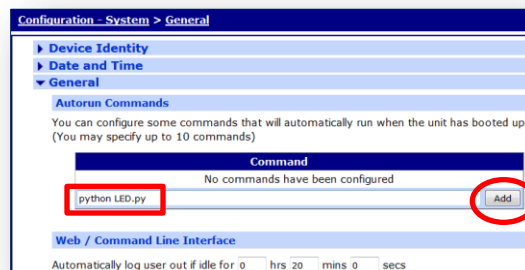
python LED.py



It is also possible to make the script running also if the device reboots, going in the WEB GUI to Configuration - system > General, then in the autorun command section type the command

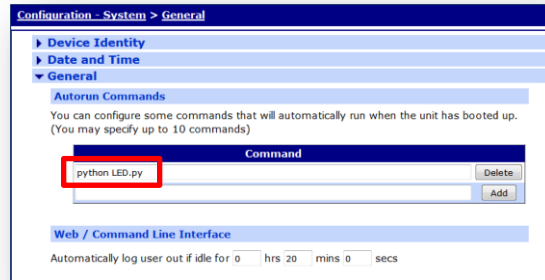
python LED.py

and click Add:



You should then see that the command is added:

QN How to make Leds show ETH and Eroute status on TransPort WR21

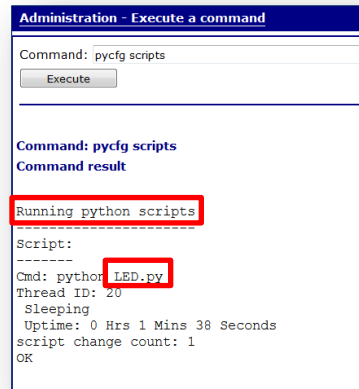


and click apply.

It is always possible to check which python script are running by issuing the following command:

pycfg scripts

In this case, the output of the command is the following:



4 TESTING LEDS BEHAVIOR

Once the script is running, some tests can be performed to check if it is working as expected. For this QN we assume that the TransPort is configured to have an IPsec tunnel UP through the ETH interface.

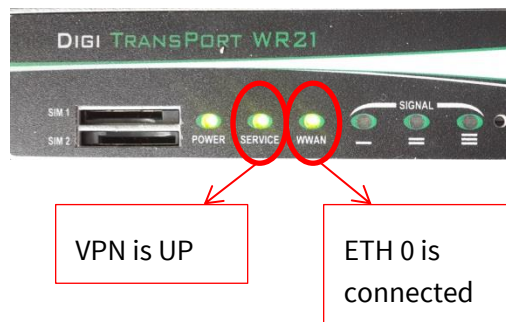
4.1 ETH 0 UP & VPN UP

With the ETH 0 connected and the VPN UP:

```
eth 0 status
Activation Status:      Active
Connection Status:     Connected
Link:                   100Base-T Full-Duplex
Physical Port:         ETH 0
MAC:                    00 04 2D 04 04 B1
IP Address:             192.168.10.202
Mask:                   255.255.255.0
Gateway:                192.168.10.1
OK

sastat
IPsec SAs (total:1). Eroute 0 -> 4
Outbound V1 SAs
  SPI Eroute      Peer IP      Rem. subnet      Loc.
subnet  TTL      KBytes Left      VIP
f8235271    0    192.168.10.1    10.104.1.109/24
192.168.4.0/24 28666          0                N/A
Inbound V1 SAs
  SPI Eroute      Peer IP      Rem. subnet      Loc.
subnet  TTL      KBytes Left      VIP
593488f0    0    192.168.10.1    10.104.1.109/24
192.168.4.0/24 28666          0                N/A
```

Run the script (if it is not running) and check the LEDs, should be both ON like the following picture:



QN How to make Leds show ETH and Eroute status on TransPort WR21

4.2 ETH 0 DOWN & VPN DOWN

Disconnect the cable of ETH 0 and check the LEDs (the one related to the VPN will take more to update):

```
eth 0 status
Activation Status:      Active
Connection Status:    Disconnected
Link:                  10Base-T Half-Duplex
Physical Port:        ETH 0
MAC:                  00 04 2D 04 04 B1
IP Address:           192.168.10.202
Mask:                 255.255.255.0
Gateway:              192.168.10.1
OK

sastat
IPsec SAs (total:0). Eroute 0 -> 4
Outbound V1 SAs
    List Empty
Inbound V1 SAs
    List Empty
```

Run the script (if it is not running) and check the LEDs, should be both ON like the following picture:



QN How to make Leds show ETH and Eroute status on TransPort WR21

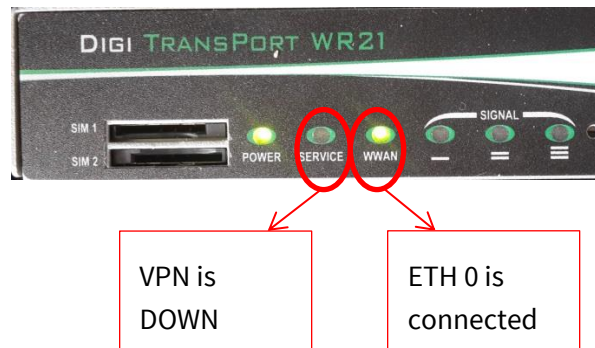
4.3 ETH 0 UP & VPN DOWN

Reconnect the ETH cable but first disable (for example) the IPsec under the ETH 0 interface:

```
eth 0 status
Activation Status:      Active
Connection Status:    Connected
Link:                  100Base-T Full-Duplex
Physical Port:        ETH 0
MAC:                   00 04 2D 04 04 B1
IP Address:           192.168.10.202
Mask:                  255.255.255.0
Gateway:              192.168.10.1
OK

sastat
IPsec SAs (total:0). Eroute 0 -> 4
Outbound V1 SAs
    List Empty
Inbound V1 SAs
    List Empty
```

Run the script (if it is not running) and check the LEDs, should be both ON like the following picture:



5 PYTHON SCRIPT

```

=====
from BaseWizard import write_cli
import time
from time import sleep

# Commands to assign the LEDs you want to control to bank 3, the syntax is ledbnk
<lednumber> <banknumber>
# In our case we have set led 2 (WWAN) and 1 (SERVICE) to bank 3

write_cli('ledbnk 2 3')
write_cli('ledbnk 1 3')

# With while TRUE and TRUE = 1 the script will run forever as soon as it is started

TRUE = 1
while TRUE :

    #Check status of eth 1
    result = write_cli('eth 1 status')
    i = result.find('Connection Status:      Connected',0)
    if i > 0:
        #LED command to turn ON the appropriate LED, syntax is ledn <lednumber>
        <ledbank> <ledstate>
        #In that case we set led 2 (WWAN) to switch ON (statenumber=2)

        write_cli('ledn 2 3 2')
    else:
        #LED command to turn OFF the appropriate LED, syntax is ledn <lednumber>
        <ledbank> <ledstate>
        #In that case we set led 2 (WWAN) to switch OFF (statenumber=1)

        write_cli('ledn 2 3 1')

    #Check status of Eroute 0
    result = write_cli('sastat 0 0')
    i = result.find('SPI Eroute',0)
    if i > 0:
        #LED command to turn ON the appropriate LED, syntax is ledn <lednumber>
        <ledbank> <ledstate>
        #In that case we set led 1 (service) to switch ON (statenumber=2)
        write_cli('ledn 1 3 2')
    else:
        #LED command to turn OFF the appropriate LED, syntax is ledn <lednumber>
        <ledbank> <ledstate>
        #In that case we set led 1 (service) to switch OFF (statenumber=1)
        write_cli('ledn 1 3 1')

    time.sleep(0.25) #Run check every 250 ms
=====
=====

```