################################################################################

# \_\_\_\_\_\_\_ #

# /\_\_\_\_\_\_\_\ #

# \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_ #

# | \_\_\_ \ / \_\_\_ \ / \_ \_ \ / \_\_\_ \ #

# | | \ \ | / \ || || || |/ / / | #

# |\_| | || | | || || || || | (\_/ #

# \_\_\_\_/ / | \\_\_\_/ || || || |\ \\_\_\_\_\_ #

# |\_\_\_\_\_/ \\_\_\_\_\_/ |\_||\_||\_| \\_\_\_\_\_\_\ #

# #

################################################################################

# File: HELP-HAMMER #

# Author: MICHAEL SKIBA #

# Date: 04-01-2018 #

# Description: FIXES GUARDIAN #

################################################################################

# Version 0.0.1 - First try

# Version 0.0.2 - Cleaned up some crap

# Version 0.0.3 - Added mega-threading capability to the UDP spammer

# Version 1.0.0 - Ready for action, changed some instruction wording

# Version 1.0.1 - Fixed 'Firmware Update, known IP' bug

################################################################################

# Make an EXE:

# pyinstaller --onefile --icon=gicon.ico --name=HelpHammer helphammer.py

import copy

import sys

import time

import itertools

import threading

import socket

try:

from winsound import Beep

except ImportError:

audio\_handle = file('/dev/audio', 'wb')

def Beep(chime, chlen):

half\_period = int(8000/chime/2)

beep = chr(1)\*half\_period+chr(0)\*half\_period

beep \*= int(chime\*chlen)

audio\_handle.write(beep)

done = False

UDP\_IP = "255.255.0.0"

UDP\_PORT = 9999

# Old logo

"""

def logoprint():

print " "

print " \_\_\_\_\_\_\_"

print " /\_\_\_\_\_\_\_\\"

print " \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_"

print " | \_\_\_ \ / \_\_\_ \ / \_ \_ \ / \_\_\_ \\"

print " | | \ \ | / \ || || || |/ / / |"

print " |\_| | || | | || || || || | (\_/"

print " \_\_\_\_/ / | \\_\_\_/ || || || |\ \\_\_\_\_\_"

print " |\_\_\_\_\_/ \\_\_\_\_\_/ |\_||\_||\_| \\_\_\_\_\_\_\\"

print " "

print " HELP-HAMMER"

print " "

"""

def logoprint():

print " \_\_\_\_\_\_\_\_ \_\_\_\_ \_\_\_ \_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_ \_\_\_\_\_ \_\_\_\_\_\_\_ "

print " / \_\_\_\_\_/| | \/ \_ \\\_\_\_\_\_\_ \\_\_\_\_\_\_ \ | | / \_ \ \ \\ "

print " / \ \_\_\_| | / /\_\ \| \_/| | \| |/ /\_\ \ / | \\ "

print " \ \\_\ \ | / | \ | \| / \ / | \/ | \\"

print " \\_\_\_\_\_\_ /\_\_\_\_\_\_/\\_\_\_\_|\_\_ /\_\_\_\_|\_ /\_\_\_\_\_\_\_ /\_\_\_\\_\_\_\_|\_\_ /\\_\_\_\_|\_\_ /"

print " \/ \/ \/ \/ \/ \/ "

print " "

print " HELP-HAMMER"

print " "

def nevergonna():

quarter = 500

doteighth = 375

eighth = 250

sixteenth = 125

f3 = 349

g3 = 392

a3 = 440

bflat3 = 466

c4 = 523

d4 = 587

eflat4 = 622

e4 = 659

f4 = 698

g4 = 784

a4 = 880

song=(

(f3, sixteenth), (g3, sixteenth), (bflat3, sixteenth), (g3, sixteenth), (d4, doteighth), (d4, doteighth),

(c4, eighth + quarter), (f3, sixteenth), (g3, sixteenth), (bflat3, sixteenth), (g3, sixteenth), (c4, doteighth), (c4, doteighth),

(bflat3, quarter), (a3, sixteenth), (g3, eighth), (f3, sixteenth), (g3, sixteenth), (bflat3, sixteenth), (g3, sixteenth), (bflat3, quarter),

(c4, eighth), (a3, eighth + sixteenth), (g3, sixteenth), (f3, quarter), (f3, eighth), (c4, quarter), (bflat3, quarter + quarter))

for chime, chlen in song:

Beep(chime, chlen)

def clearit(item):

sys.stdout.write(item)

sys.stdout.flush()

sys.stdout.write('\r')

sys.stdout.flush()

def countdown(t):

while t:

mins, secs = divmod(t, 60)

timeformat = '{:02d}:{:02d}'.format(mins, secs)

sys.stdout.write(timeformat)

sys.stdout.flush()

sys.stdout.write('\r')

sys.stdout.flush()

time.sleep(1)

t -= 1

def fanimate():

for c in itertools.cycle(['|', '/', '-', '\\']):

if done:

sys.stdout.write('\r')

break

sys.stdout.write('\rloading ' + c)

sys.stdout.flush()

time.sleep(0.1)

def printbreak():

time.sleep(0.6)

for i in range(40):

print "-",

time.sleep(0.005)

else:

print

def validip(s):

a = s.split('.')

if len(a) != 4:

return False

for x in a:

if not x.isdigit():

return False

i = int(x)

if i < 0 or i > 255:

return False

return True

def firmwareup(beta, UDP\_IP):

if (beta == True):

print("\n HOT FRESH BETA COMING UP!!!")

MESSAGE1 = '{"command":"update\_esp32", "address":"104.236.234.184", "port":"80", "filename":"/G/esp\_beta.bin", "type":0}'

MESSAGE2 = '{"command":"update\_lora", "address":"104.236.234.184", "port":"80", "filename":"/G/st\_beta.bin", "type":0}'

printbreak()

elif (beta ==False):

print("\n Production firmware coming up!!!")

MESSAGE1 = '{"command":"update\_esp32", "address":"104.236.234.184", "port":"80", "filename":"/G/esp\_prod.bin", "type":0}'

MESSAGE2 = '{"command":"update\_lora", "address":"104.236.234.184", "port":"80", "filename":"/G/st\_prod.bin", "type":0}'

printbreak()

print "Sending ESP32 update, please wait"

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

sock.sendto(MESSAGE1, (UDP\_IP, UDP\_PORT))

countdown(30)

printbreak()

print "Sending ST update, please wait"

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

sock.sendto(MESSAGE2, (UDP\_IP, UDP\_PORT))

countdown(30)

def espup(beta, UDP\_IP):

if (beta == True):

MESSAGE1 = '{"command":"update\_esp32", "address": "104.236.234.184", "port":"80", "filename":"/G/esp\_beta.bin", "type":0}'

elif (beta ==False):

MESSAGE1 = '{"command":"update\_esp32", "address":"104.236.234.184", "port":"80", "filename":"/G/esp\_prod.bin", "type":0}'

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

sock.sendto(MESSAGE1, (UDP\_IP, UDP\_PORT))

time.sleep(0.06)

def stup(beta, UDP\_IP):

if (beta == True):

MESSAGE1 = '{"command":"update\_lora", "address": "104.236.234.184", "port":"80", "filename":"/G/st\_beta.bin", "type":0}'

elif (beta ==False):

MESSAGE1 = '{"command":"update\_lora", "address":"104.236.234.184", "port":"80", "filename":"/G/st\_prod.bin", "type":0}'

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

sock.sendto(MESSAGE1, (UDP\_IP, UDP\_PORT))

time.sleep(0.06)

def wificonn():

print ("""

Please connect your PC to the Guardian hotspot

It will be displayed as Guardian-XXXX, where

XXXX is the device PIN

""")

raw\_input("Press Enter when you're connected")

print ("""

Please enter the Wi-Fi network name

(Case sensitive!!!)

""")

ssid = raw\_input(" >>> ")

printbreak()

print ("""

Please enter the Wi-Fi password

(Case Sensitive!!!)

""")

password = raw\_input(" >>> ")

printbreak()

print "Please wait..."

UDP\_IP = "192.168.4.1"

UDP\_PORT = 9999

wifimessage = '{"command":"set\_WIFI\_station","type":0,"SSID":"' + ssid + '","' + 'PASS":"' + password + '", "connect":0}'

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

sock.sendto(wifimessage, (UDP\_IP, UDP\_PORT))

time.sleep(5)

apmessage = '{"command":"set\_WIFI\_ap", "option":0 ,"type":0}'

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

sock.sendto(apmessage, (UDP\_IP, UDP\_PORT))

time.sleep(5)

printbreak()

menu0()

def menu0():

ans=True

while ans:

print ("""

Please connect your PC to the same Wi-Fi

network as the Guardian valve controller.

If you need to connect the Guardian valve

controller to Wi-Fi, select '5'

1. Firmware update with IP

2. Beta firmware update with IP

3. Firmware update, IP unknown

4. Beta firmware update, IP unknown

5. Connect Guardian to Wi-Fi

6. Beta Hammer (legacy mode)

7. Exit

""")

ans=raw\_input(" What will it be, buddy?: ")

if ans=="1":

print("\n Ok hold on")

printbreak()

menu01(False)

elif ans=="2":

print("\n Sure, one sec")

printbreak()

menu01(True)

elif ans == "3":

print("\n Ok hold on")

printbreak()

menu02(False, False)

elif ans == "4":

print("\n Sure, one sec")

printbreak()

menu02(True, False)

elif ans == "6":

print("\n Sure, one sec")

printbreak()

menu02(True, True)

elif ans == "5":

print("\n You got it!")

printbreak()

wificonn()

elif ans=="7":

printbreak()

print("\n BYE BYE!!")

time.sleep(2)

exit(0)

elif ans !="":

print("\n Not a valid choice, try again!!")

printbreak()

def menuXX():

ans=True

while ans:

print ("""

Did it work?

""")

ans=raw\_input(" Y/N: ")

if (ans=="N" or ans=="n"):

print("\n OH NO!")

printbreak()

menu0()

elif (ans=="Y" or ans=="y"):

print("\n OK!!")

printbreak()

time.sleep(2)

sys.exit(0)

elif ans !="":

print("\n Not a valid choice, try again!!")

printbreak()

def menu01(beta):

print ("""

Please connect to the same Wi-Fi

network as the Guardian

""")

raw\_input("Press Enter when you're connected")

ans=True

while ans:

print ("""

IP of Guardian?

""")

UDP\_IP = raw\_input("IP: ")

ans = validip(UDP\_IP)

if ans:

firmwareup(beta, UDP\_IP)

printbreak()

menuXX()

elif not ans:

print("\n Not a valid IP")

time.sleep(2)

printbreak()

ans = True

def menu02(beta, compat):

print ("""

Please connect to the same Wi-Fi

network as the Guardian

""")

raw\_input("Press Enter when you're connected")

printbreak()

ans=True

while ans:

goodip = str(socket.gethostbyname(socket.gethostname()))

UDP\_IP = copy.copy(goodip)

ans = validip(UDP\_IP)

subnettemp = UDP\_IP

subnetsplit = subnettemp.split('.')

subnet = subnetsplit[0]+'.'+subnetsplit[1]+'.'+subnetsplit[2]+'.'

if ans:

print "HOT FRESH BETA COMING UP!!!"

time.sleep(1)

printbreak()

print "Performing update 1 of 2..."

if compat:

for i in range(1,255):

items = str(i) + "/255"

clearit(items)

TDP\_IP = subnet + str(i)

espup(beta, TDP\_IP)

clearit(" ")

print "Preparing payload, please wait..."

countdown(30)

printbreak()

print "Performing update 2 of 2..."

for i in range(1,255):

items = str(i) + "/255"

clearit(items)

TDP\_IP = subnet + str(i)

stup(beta, TDP\_IP)

clearit(" ")

print "Clearing cache, please wait..."

countdown(30)

clearit(" ")

printbreak()

menuXX()

else:

for i in range(1,255):

items = str(i) + "/255"

clearit(items)

TDP\_IP = subnet + str(i)

t = threading.Thread(target = espup, args = (beta, TDP\_IP))

t.start()

if i == 254:

t.join()

clearit(" ")

print "Preparing payload, please wait..."

countdown(30)

printbreak()

print "Performing update 2 of 2..."

for i in range(1,255):

items = str(i) + "/255"

clearit(items)

TDP\_IP = subnet + str(i)

p = threading.Thread(target = stup, args = (beta, TDP\_IP))

p.start()

if i == 254:

p.join()

clearit(" ")

print "Clearing cache, please wait..."

countdown(30)

clearit(" ")

printbreak()

menuXX()

elif not ans:

print("\n Not a valid IP")

time.sleep(2)

printbreak()

ans = True

if \_\_name\_\_=="\_\_main\_\_":

logoprint()

t = threading.Thread(target=fanimate)

t.start()

nevergonna()

done = True

sys.stdout.write('\r')

sys.stdout.flush()

print " "

printbreak()

menu0()