



Maker Faire 2017

Chen-Hanson Ting

SVFIG

February 25, 2017



Maker Faire 2017

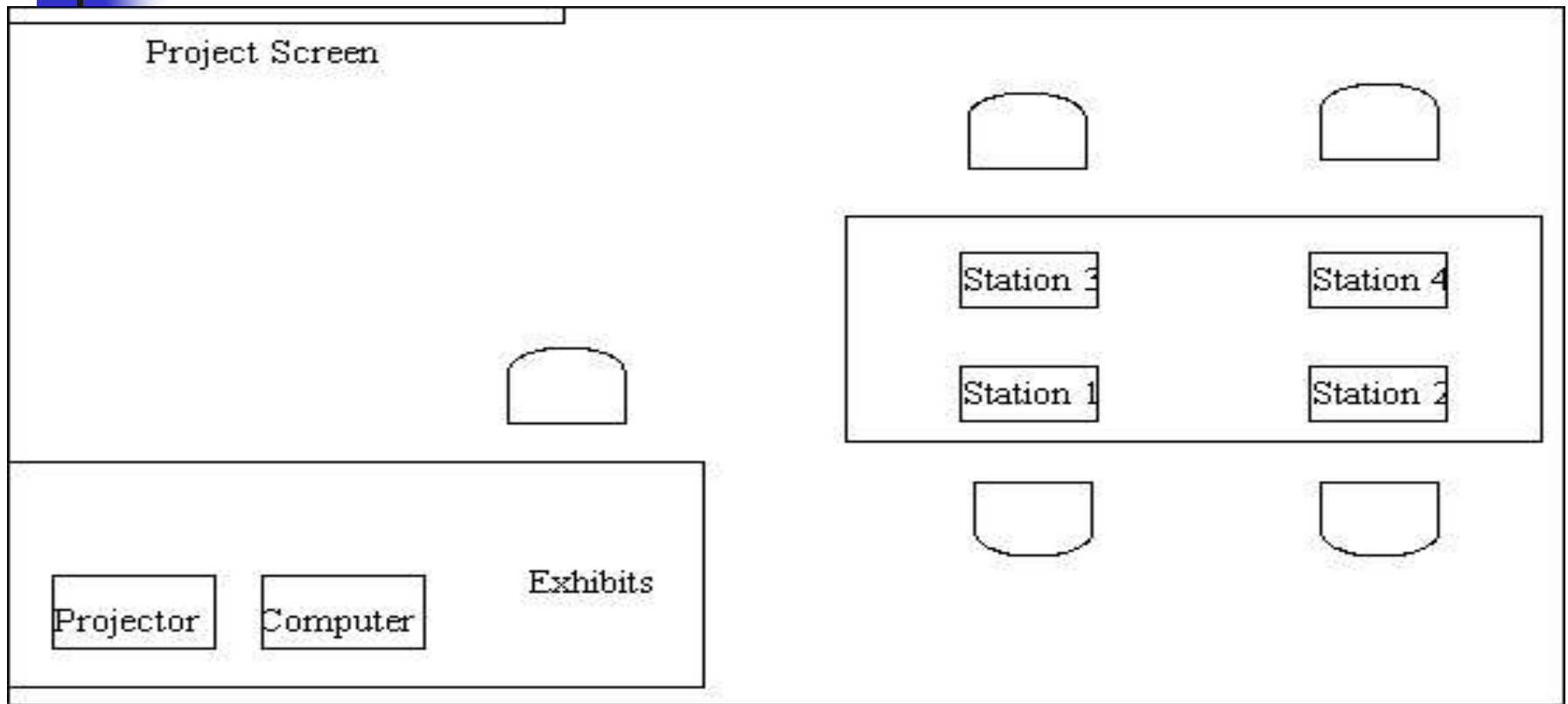
- May 20-21, 10 am to 6 pm
- Theme: IoT For Fun.
- Tutorials and hands-on training
- Need 4 Windows 7 class computers to set up 4 work stations
- I have projector for tutorials and presentations



Maker Faire 2017

- Tutorials and experiments will be focused on ESP8266
- Like to emphasize Forth, but will include Lua, Arduino, and MicroPython.
- A small area will be reserved to show your own projects.
- Take a NodeMCU ESP8266 kit home and start doing something.

Maker Faire Booth





Maker Faire Sign-Ups

- Booth Duty: 2 hours, Saturday or Sunday, morning or afternoon
- There will be 6 (maybe) free tickets (\$50 value) for booth duty.
- Tutorials or Presentations: Scheduled for every hour, select your preferred time and topics. PowerPoint slides are encouraged.



MicroPython

Chen-Hanson Ting

SVFIG

February 25, 2017



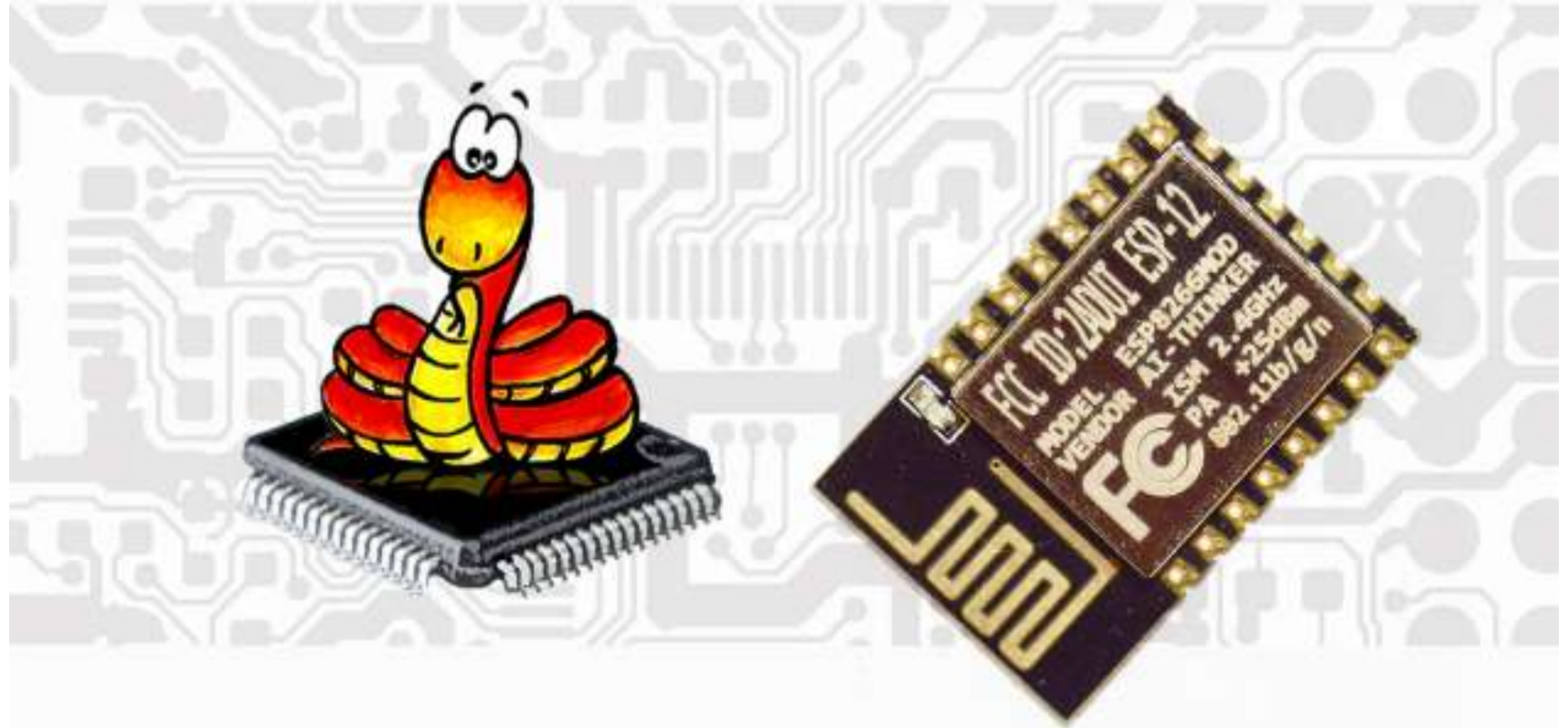
ESP8266

- It looks that ESP8266 12E will replace Arduino Uno, with its WiFi capability, 32-bit processor, and large memories.
- With ESP8266 kits, we can participate in the new IoT revolution, and have lots of fun.

NodeMCU and Arduino Uno



Micropython



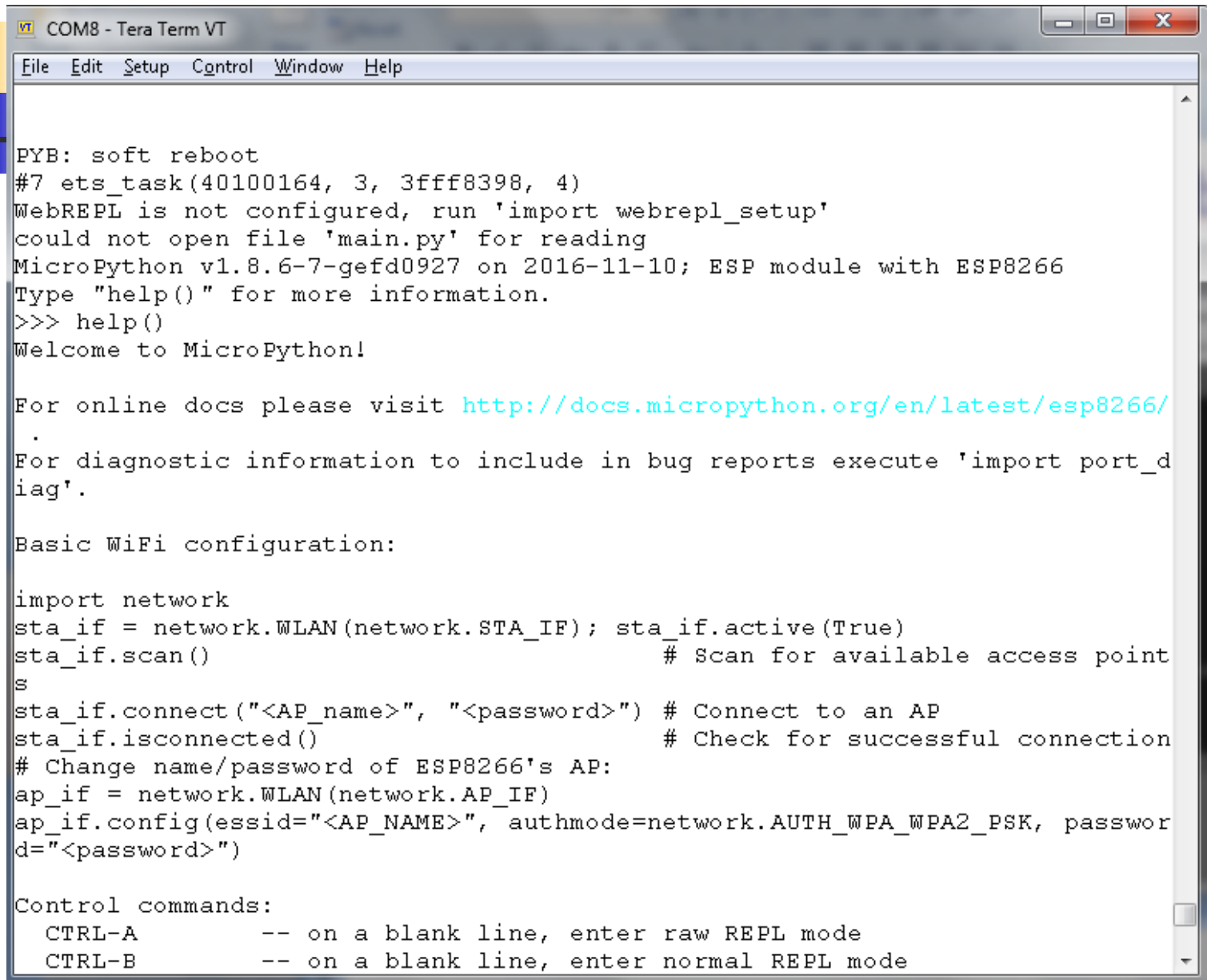
MicroPython on the ESP8266



Micropython

- Micropython is a powerful interpreting language for ESP8266.
- WebREPL (Read-Evaluate-Print Loop) is a graphic interface on PC to control ESP8266 remotely.
- A complete IDE for firmware engineering.

REPL on Tera Term



The screenshot shows a Tera Term window titled "COM8 - Tera Term VT". The window contains the following text:

```
File Edit Setup Control Window Help

PYB: soft reboot
#7 ets_task(40100164, 3, 3fff8398, 4)
WebREPL is not configured, run 'import webrepl_setup'
could not open file 'main.py' for reading
MicroPython v1.8.6-7-gefd0927 on 2016-11-10; ESP module with ESP8266
Type "help()" for more information.
>>> help()
Welcome to MicroPython!

For online docs please visit http://docs.micropython.org/en/latest/esp8266/
.
For diagnostic information to include in bug reports execute 'import port_d
iag'.

Basic WiFi configuration:

import network
sta_if = network.WLAN(network.STA_IF); sta_if.active(True)
sta_if.scan() # Scan for available access point
s
sta_if.connect("<AP_name>", "<password>") # Connect to an AP
sta_if.isconnected() # Check for successful connection
# Change name/password of ESP8266's AP:
ap_if = network.WLAN(network.AP_IF)
ap_if.config(essid="<AP_NAME>", authmode=network.AUTH_WPA_WPA2_PSK, passwor
d="<password>")

Control commands:
CTRL-A -- on a blank line, enter raw REPL mode
CTRL-B -- on a blank line, enter normal REPL mode
```

WebREPL

ws://192.168.4.1:8266/ Disconnect

```
>>>
>>> Disconnected
Welcome to MicroPython!
Password:
WebREPL connected
>>> import bwv946
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: no module named 'bwv946'
>>> import bwv846
>>> bwv846.last()
>>> bwv846.last()
>>> Disconnected
Welcome to MicroPython!
Password:
WebREPL connected
>>> import os
>>> os.listdir()
['boot.py', 'webrepl_cfg.py', 'silentNight.py', 'melody.py', 'bwv846.py']
>>> import bwv846
>>> bwv846.prelude()
>>>
>>>
>>> kk1
```

Send a file

Choose File bwv846.py
bwv846.py - 1489 bytes
Send to device

Get a file

Get from device

Sent bwv846.py, 1489 bytes

*Terminal widget should be focused (text cursor visible) to accept input. Click on it if not.
To paste, press Ctrl+A, then Ctrl+V*

NodeMCU (\$3.20)

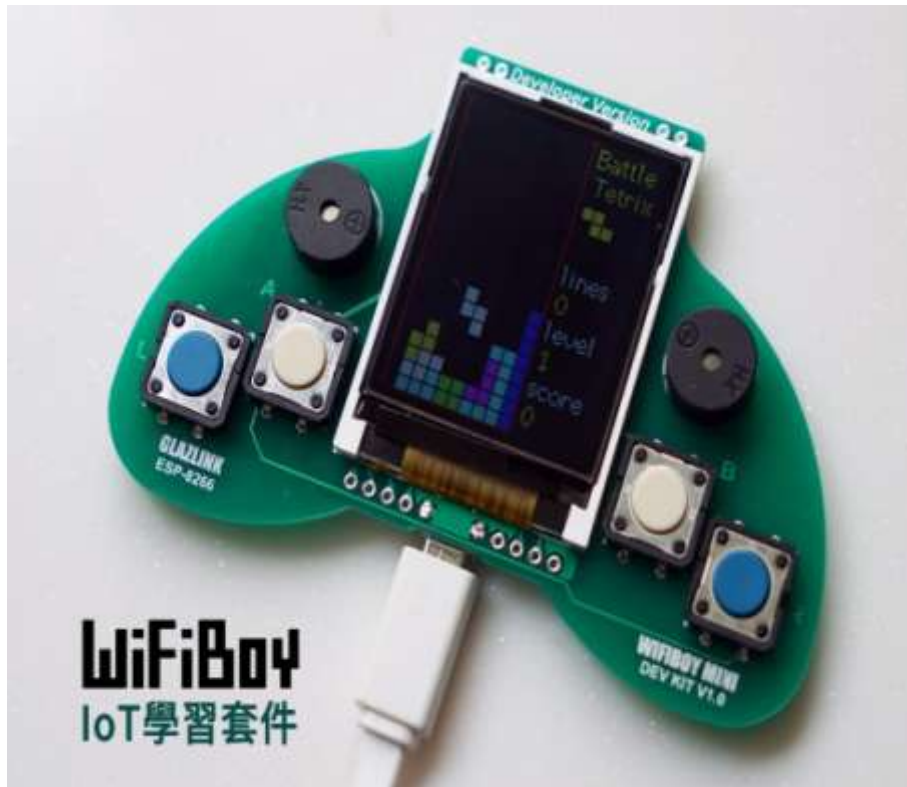




ESP8266 NodeMCU Kit

- A complete IoT firmware engineering platform
- Xtensa LX3 CPU, 32-bit, 160 MHz
- RAM 32Kb, DRAM 80Kb, Flash 4 Mb
- Wi-Fi 802.11 b/g/n 2.4 GHz radio
- MicroUSB connector
- GPIO, PWM, ADC, UART, I2C, SPI

WiFiBoy





Firmware Engineering

- IDE: Integrated Development Environment
- Programming tools
- Flashing tools
- Power up configuration
- I/O interfacing



MicroPython

- REPL: Read-Eval-Process-Loop
- Remote Monitor: WebREPL
- Flash file system
- File download
- Extensive libraries



MicroPython Libraries

- machine
 - Pin, PWM
- Flash file system
 - os, open
- network
 - STA_IF
 - AP_IF

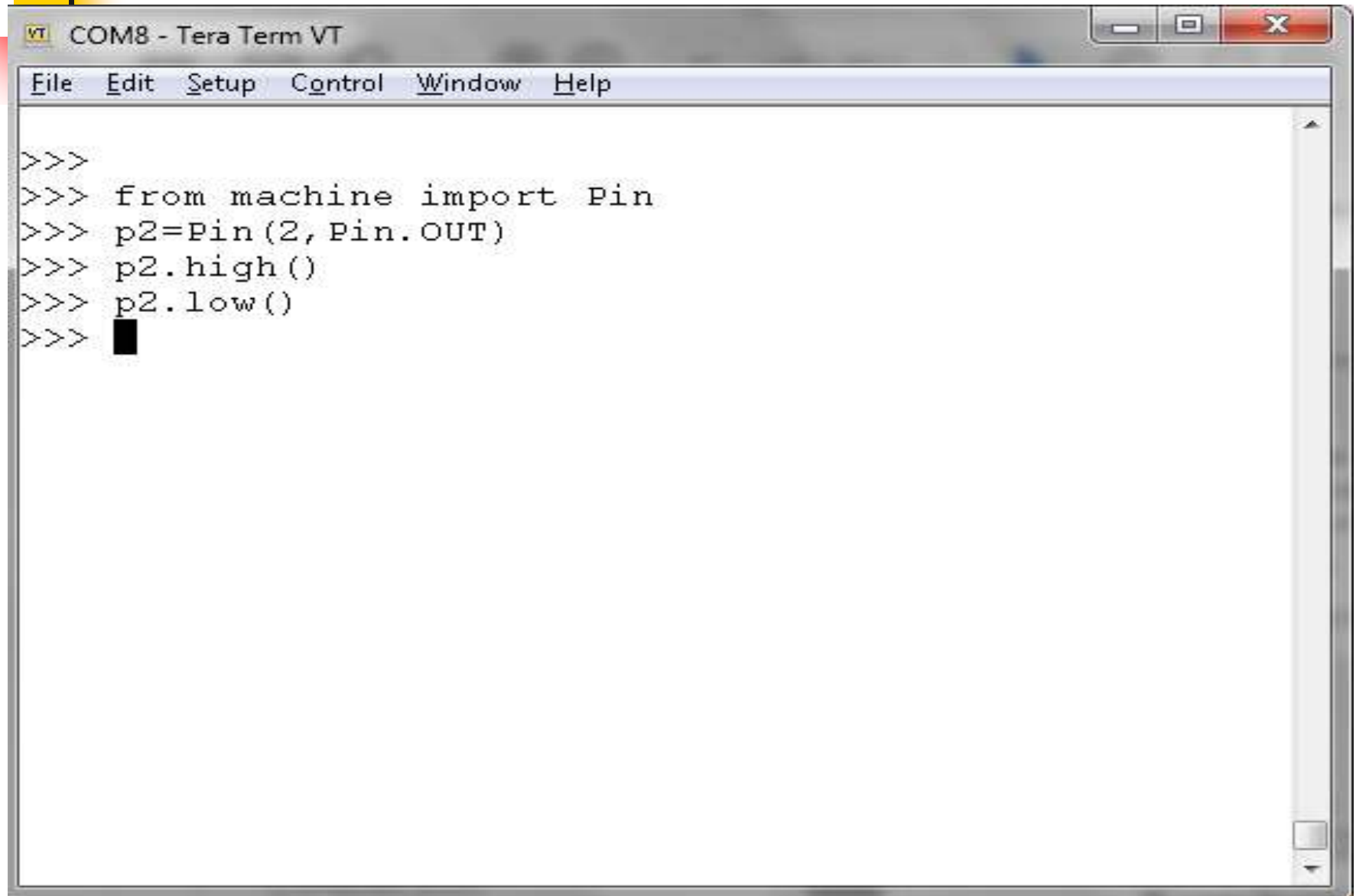
Machine Library

VT COM8 - Tera Term VT

File Edit Setup Control Window Help

```
>>> import machine
>>> dir(machine)
['__name__', 'mem8', 'mem16', 'mem32', 'freq', 'reset',
 'reset_cause', 'unique_id', 'idle', 'sleep', 'deepsleep',
 'disable_irq', 'enable_irq', 'time_pulse_us', 'RTC', 'Timer',
 'WDT', 'Pin', 'PWM', 'ADC', 'UART', 'I2C', 'SPI', 'DEEPSLEEP',
 'PWRON_RESET', 'HARD_RESET', 'DEEPSLEEP_RESET', 'WDT_RESET',
 'SOFT_RESET']
>>> █
```

Pin Function



```
COM8 - Tera Term VT
File Edit Setup Control Window Help
>>>
>>> from machine import Pin
>>> p2=Pin(2, Pin.OUT)
>>> p2.high()
>>> p2.low()
>>> █
```

PWM Function

```
VT COM8 - Tera Term VT
File Edit Setup Control Window Help
>>>
>>> from machine import Pin
>>> p2=Pin(2, Pin.OUT)
>>> p2.high()
>>> p2.low()
>>>
>>>
>>> from machine import Pin, PWM
>>> p14=PWM(14)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: expecting a pin
>>> p14=PWM(Pin(14))
>>> p14.freq(440)
>>> p14.duty(512)
>>> p14.duty(0)
>>>
```

os Library

VT COM8 - Tera Term VT

File Edit Setup Control Window Help

```
>>> import os
>>> os.listdir()
['boot.py', 'webrepl_cfg.py', 'SilentNight.py', 'melody.py', 'bwv846.py']
>>> help(os)
object <module 'uos'> is of type module
  __name__ -- uos
  uname -- <function>
  urandom -- <function>
  dupterm -- <function>
  dupterm_notify -- <function>
  VfsFat -- <class 'VfsFat'>
  listdir -- <function>
  mkdir -- <function>
  rmdir -- <function>
  chdir -- <function>
  getcwd -- <function>
  remove -- <function>
  rename -- <function>
  stat -- <function>
  statvfs -- <function>
  umount -- <function>
```

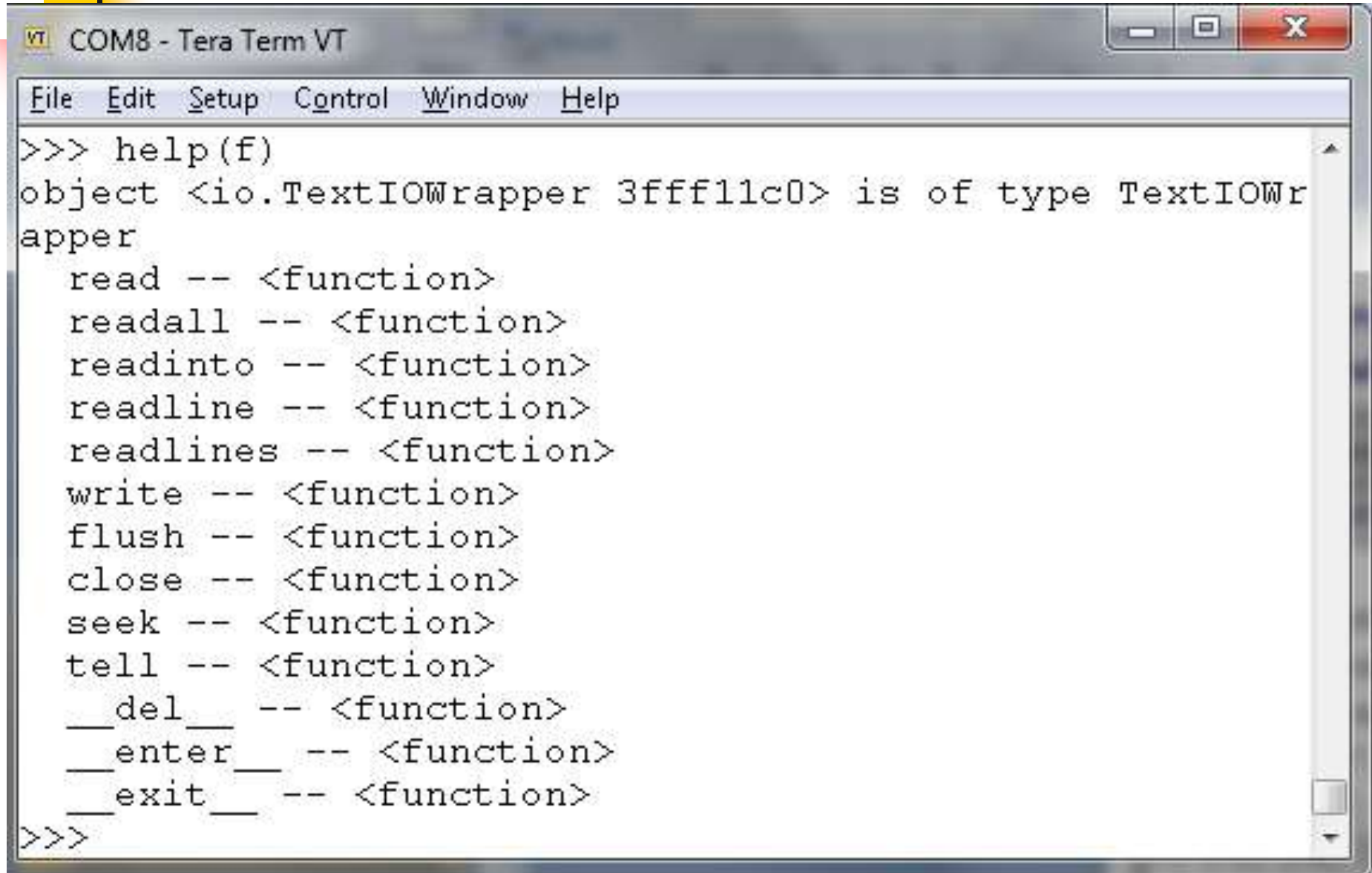
Open() Build-in FileSystem

COM8 - Tera Term VT

File Edit Setup Control Window Help

```
>>>
>>> f=open('bwv846.py','r')
>>> f.read()
'import machine, time\r\nsnd=machine.PWM(machine.Pin(1
4, 1))\r\nenv=[128,64,32,16,8,4,2,1]\r\nf=[round(55*2*
*(x/12)) for x in range(51)]\r\ndt=0.025\r\nmel=[\r\n2
7,31,34,39,43,\r\n27,29,36,41,44,\r\n26,29,34,41,44,\r
\n27,31,34,39,43,\r\n27,31,36,43,48,\r\n27,29,33,36,41
,\r\n26,29,34,41,46,\r\n26,27,31,34,39,\r\n24,27,31,34
,39,\r\n17,24,29,33,39,\r\n22,26,29,34,38,\r\n22,25,31
,34,40,\r\n20,24,29,36,41,\r\n20,23,29,32,38,\r\n19,22
,27,34,39,\r\n19,20,24,27,32,\r\n17,20,24,27,32,\r\n10
,17,22,26,32,\r\n15,19,22,27,31,\r\n15,22,25,27,31,\r\n
n8,20,24,27,31,\r\n9,15,24,27,30,\r\n10,18,26,27,30,\r
\n11,20,26,27,29,\r\n10,20,22,26,29,\r\n10,19,22,27,31
,\r\n10,17,22,27,32,\r\n10,17,22,26,32,\r\n10,18,24,27
,33,\r\n10,19,22,27,34,\r\n10,17,22,27,32,\r\n10,17,22
```

Open() Build-in FileSystem



```
VT COM8 - Tera Term VT
File Edit Setup Control Window Help
>>> help(f)
object <io.TextIOWrapper 3fff11c0> is of type TextIOWr
apper
  read -- <function>
  readall -- <function>
  readinto -- <function>
  readline -- <function>
  readlines -- <function>
  write -- <function>
  flush -- <function>
  close -- <function>
  seek -- <function>
  tell -- <function>
  __del__ -- <function>
  __enter__ -- <function>
  __exit__ -- <function>
>>>
```


Bach C Major Prelude

bwv846.py - Notepad

File Edit Format View Help

```
import machine, time
snd=machine.PWM(machine.Pin(14, 1))

env=[128, 64, 32, 16, 8, 4, 2, 1]
f=[round(55*2**(x/12)) for x in range(51)]
dt=0.025

mel=[
27, 31, 34, 39, 43,
27, 29, 36, 41, 44,
26, 29, 34, 41, 44,
27, 31, 34, 39, 43,
27, 31, 36, 43, 48,
27, 29, 33, 36, 41,
26, 29, 34, 41, 46,
26, 27, 31, 34, 39,
24, 27, 31, 34, 39,
17, 24, 29, 33, 39,
22, 26, 29, 34, 38,
22, 25, 31, 34, 40,
```

Bach C Major Prelude

```
bwv846.py - Notepad
File Edit Format View Help

def note(array, n):
    snd.freq(f[array[n]])
    for i in range(7):
        snd.duty(env[i])
        time.sleep(dt)

def chord(n):
    for i in range(1):
        note(mel, n)
        note(mel, n+1)
        note(mel, n+2)
        note(mel, n+3)
        note(mel, n+4)
        note(mel, n+2)
        note(mel, n+3)
        note(mel, n+4)

def play():
    j=0
```

Bach C Major Prelude

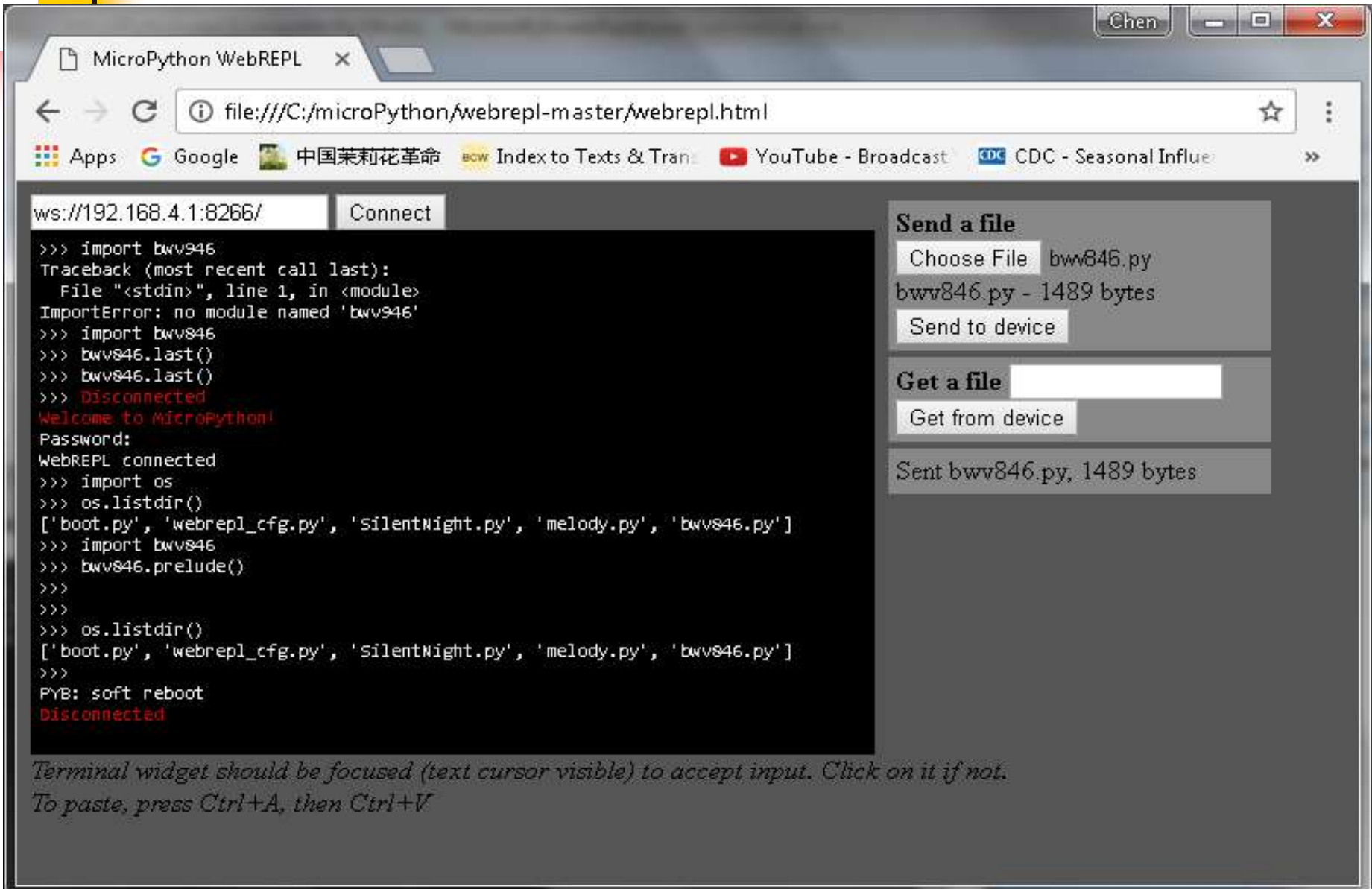
```
bwv846.py - Notepad
File Edit Format View Help
def play():
    j=0
    for i in range(32):
        chord(j)
        snd.duty(0)
        time.sleep(4*dt)
        j=j+5

def last():
    for i in range(len(end)):
        note(end,i)

def prelude():
    play()
    last()
    time.sleep(16*dt)
    snd.duty(0)

prelude()
```

WebREPL Download / Run



MicroPython WebREPL

file:///C:/microPython/webrepl-master/webrepl.html

ws://192.168.4.1:8266/ Connect

```
>>> import bww846
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: no module named 'bww846'
>>> import bww846
>>> bww846.last()
>>> bww846.last()
>>> Disconnected
Welcome to MicroPython!
Password:
WebREPL connected
>>> import os
>>> os.listdir()
['boot.py', 'webrepl_cfg.py', 'silentnight.py', 'melody.py', 'bww846.py']
>>> import bww846
>>> bww846.prelude()
>>>
>>>
>>> os.listdir()
['boot.py', 'webrepl_cfg.py', 'silentnight.py', 'melody.py', 'bww846.py']
>>>
PYB: soft reboot
Disconnected
```

Send a file

Choose File bww846.py
bww846.py - 1489 bytes

Send to device

Get a file

Get from device

Sent bww846.py, 1489 bytes

*Terminal widget should be focused (text cursor visible) to accept input. Click on it if not.
To paste, press Ctrl+A, then Ctrl+V*



Local Area Network

- A local area network (LAN) with a number of ESP8266 running Forth. They can communicate with a host computer.
- A host computer sends out Forth commands to each ESP8266 to accomplish certain task.

Network Library

VT COM8 - Tera Term VT

File Edit Setup Control Window Help

```
Traceback (most recent call last):
  File "<stdin>", line 2
SyntaxError: invalid syntax
>>>
>>> import network
>>> sta=network.WLAN(network.STA_IF)
>>> ap=network.WLAN(network.AP_IF)
>>> sta.active()
True
>>> ap.active()
True
>>> ap.ifconfig()
('192.168.4.1', '255.255.255.0', '192.168.4.1', '192.168.1.1')
>>>
```

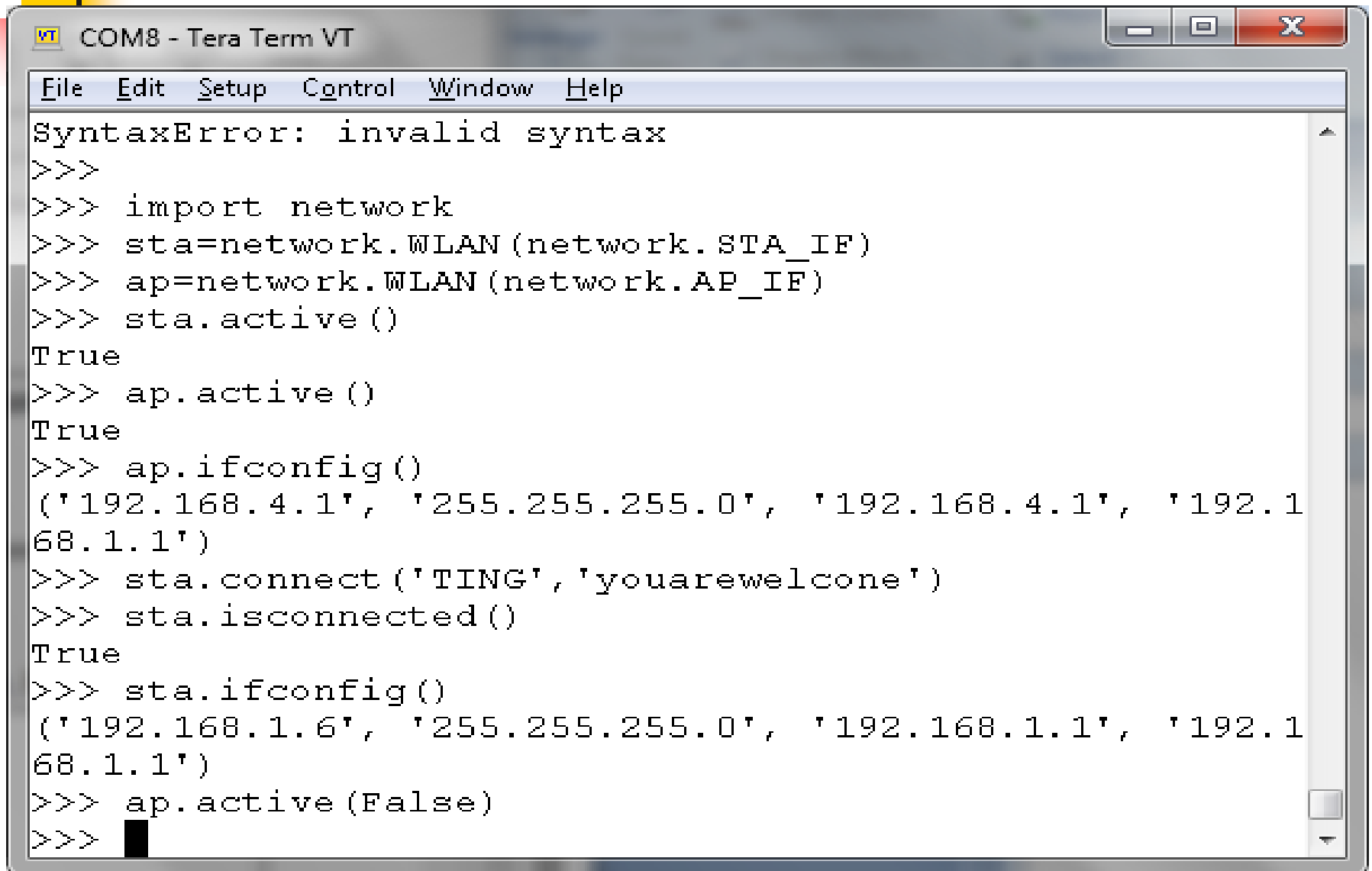
Network Library

VT COM8 - Tera Term VT

File Edit Setup Control Window Help

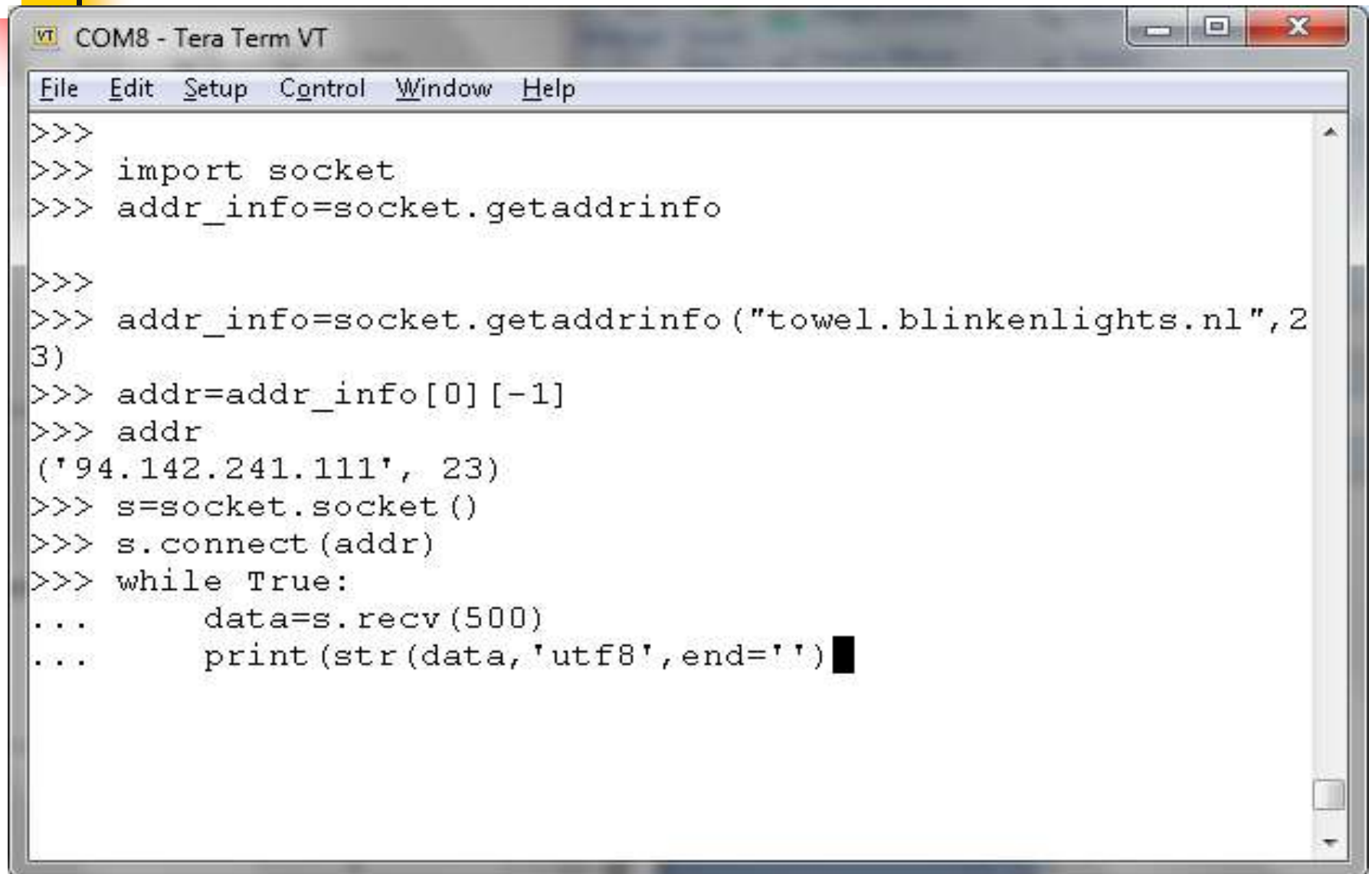
```
Traceback (most recent call last):
  File "<stdin>", line 2
SyntaxError: invalid syntax
>>>
>>> import network
>>> sta=network.WLAN(network.STA_IF)
>>> ap=network.WLAN(network.AP_IF)
>>> sta.active()
True
>>> ap.active()
True
>>> ap.ifconfig()
('192.168.4.1', '255.255.255.0', '192.168.4.1', '192.168.1.1')
>>>
```

Station Connected



```
COM8 - Tera Term VT
File Edit Setup Control Window Help
SyntaxError: invalid syntax
>>>
>>> import network
>>> sta=network.WLAN(network.STA_IF)
>>> ap=network.WLAN(network.AP_IF)
>>> sta.active()
True
>>> ap.active()
True
>>> ap.ifconfig()
('192.168.4.1', '255.255.255.0', '192.168.4.1', '192.168.1.1')
>>> sta.connect('TING', 'youarewelcone')
>>> sta.isconnected()
True
>>> sta.ifconfig()
('192.168.1.6', '255.255.255.0', '192.168.1.1', '192.168.1.1')
>>> ap.active(False)
>>>
```


TCP Socket

A screenshot of a Tera Term VT window titled "COM8 - Tera Term VT". The window has a menu bar with "File", "Edit", "Setup", "Control", "Window", and "Help". The main area contains Python code for a TCP socket connection. The code starts with three blank lines, followed by "import socket" and "addr_info=socket.getaddrinfo". There are two blank lines, then "addr_info=socket.getaddrinfo('towel.blinkenlights.nl', 23)", "addr=addr_info[0][-1]", and "addr". This is followed by "s=socket.socket()", "s.connect(addr)", and a "while True:" loop. Inside the loop, there are two lines: "... data=s.recv(500)" and "... print(str(data, 'utf8', end=' '))". The cursor is at the end of the last line.

```
>>>
>>> import socket
>>> addr_info=socket.getaddrinfo

>>>
>>> addr_info=socket.getaddrinfo("towel.blinkenlights.nl", 2
3)
>>> addr=addr_info[0][-1]
>>> addr
('94.142.241.111', 23)
>>> s=socket.socket()
>>> s.connect(addr)
>>> while True:
...     data=s.recv(500)
...     print(str(data, 'utf8', end=' '))
```



Closing Remarks

- ESP8266 is IoT ready.
- Are we ready?
- There are many IDE's already available for ESP8266.
- Will Forth play a role in this ESP8266 revolution?