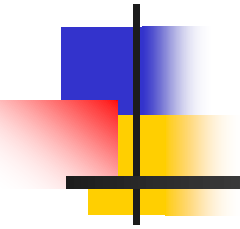
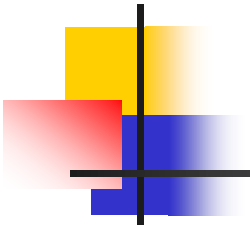


IoT For Fun!

Chen-Hanson Ting

SVFIG

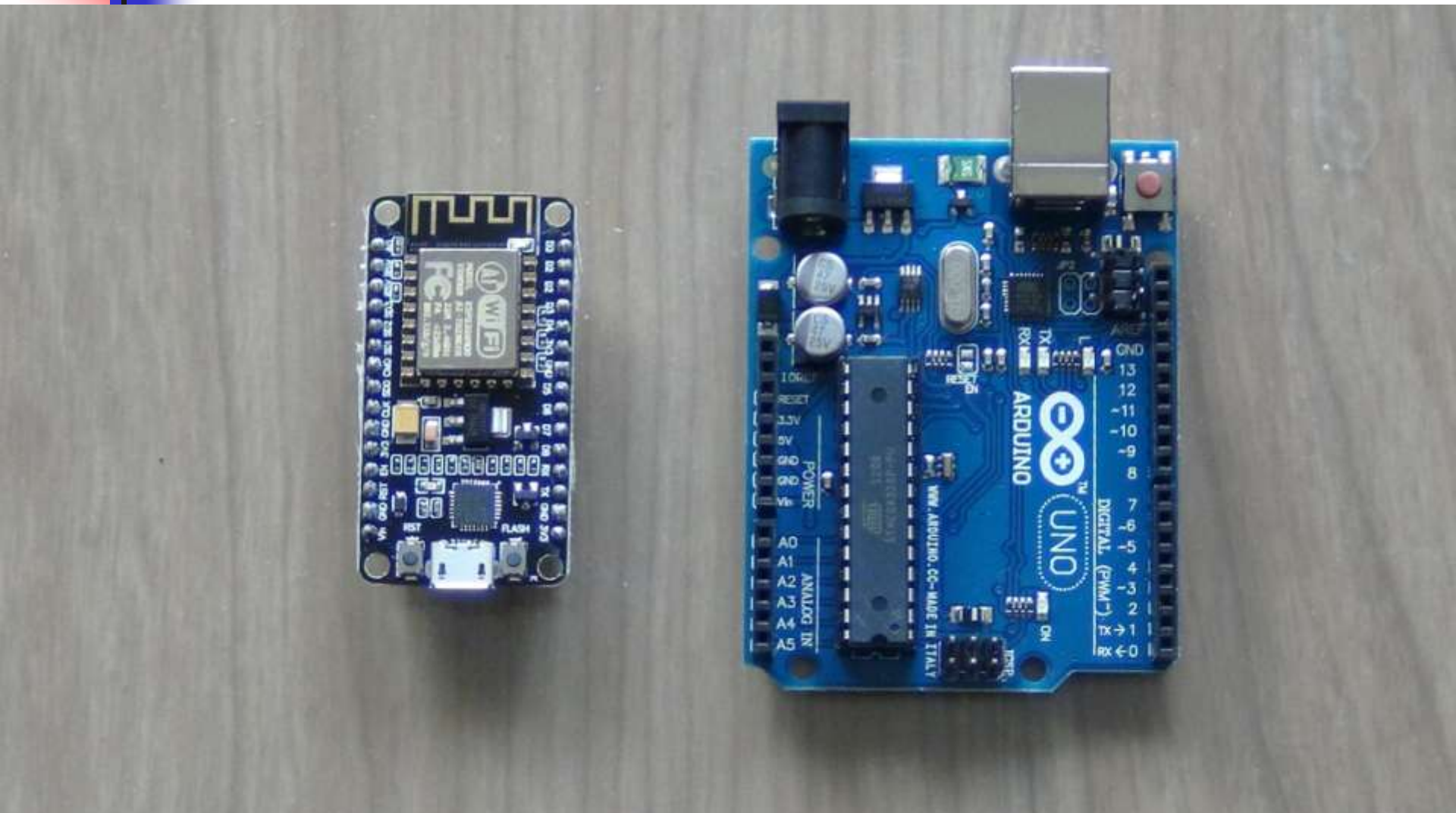
January 28, 2017



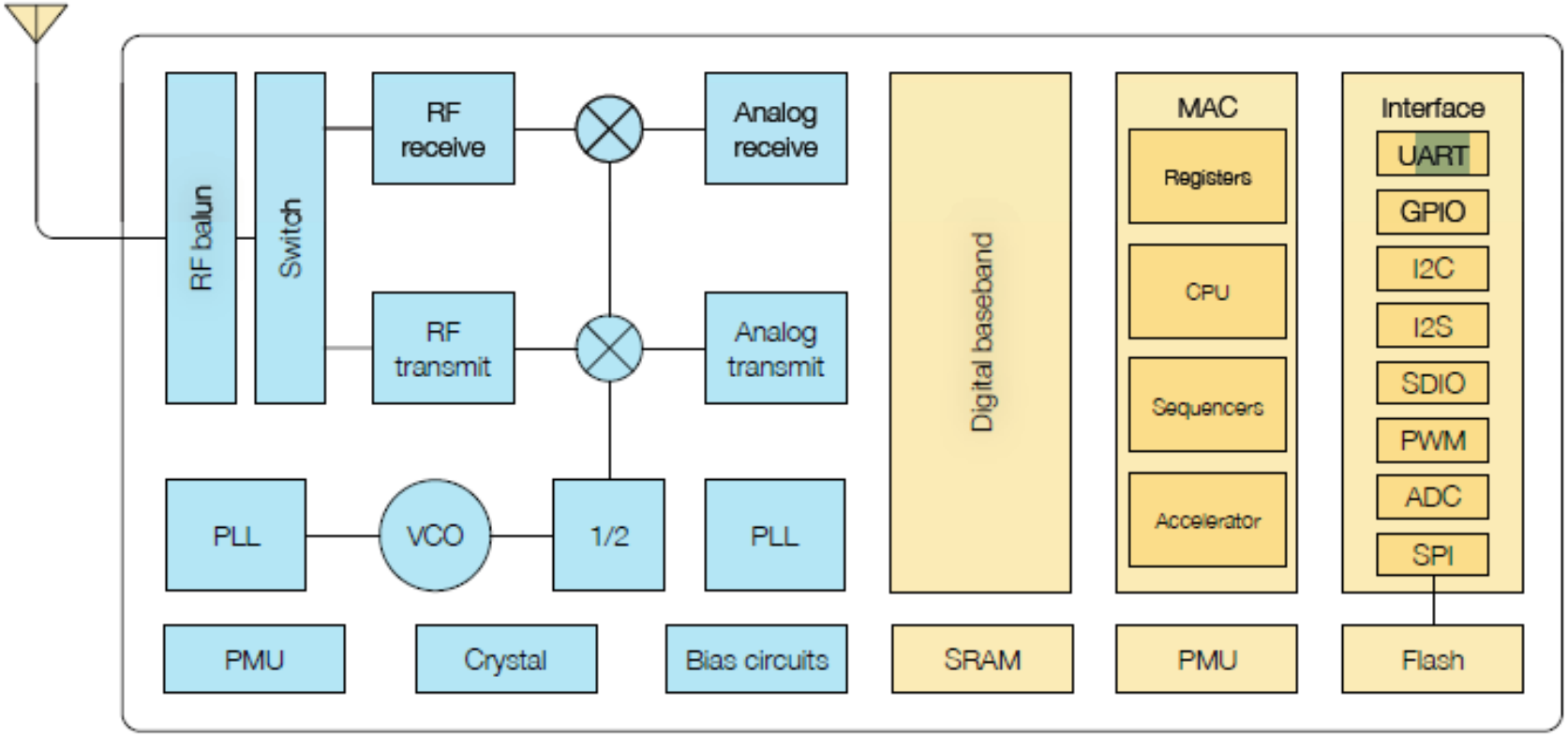
ESP8266

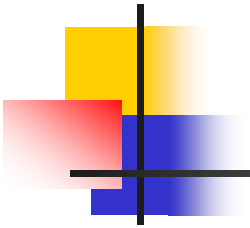
- It looks that ESP8266 12E will replace Arduino Uno, with its WiFi capability, 32-bit processor, and large memories.
- With ESP8266 kits, we can participate in the new IoT revolution, and have lots of fun.

ESP 12E and Arduino Uno



ESP8266


RF balun
Switch
RF receive
RF transmit
Analog receive
Analog transmit
PLL
VCO
1/2
PLL
Digital baseband
MAC
Registers
CPU
Sequencers
Accelerator
Interface
UART
GPIO
I2C
I2S
SDIO
PWM
ADC
SPI
PMU
Crystal
Bias circuits
SRAM
PMU
Flash



ESP8266 Hardware

- 3.3V, ~ 215 mA
- 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
- GPIO, PWM, ADC, UART, I2C, SPI

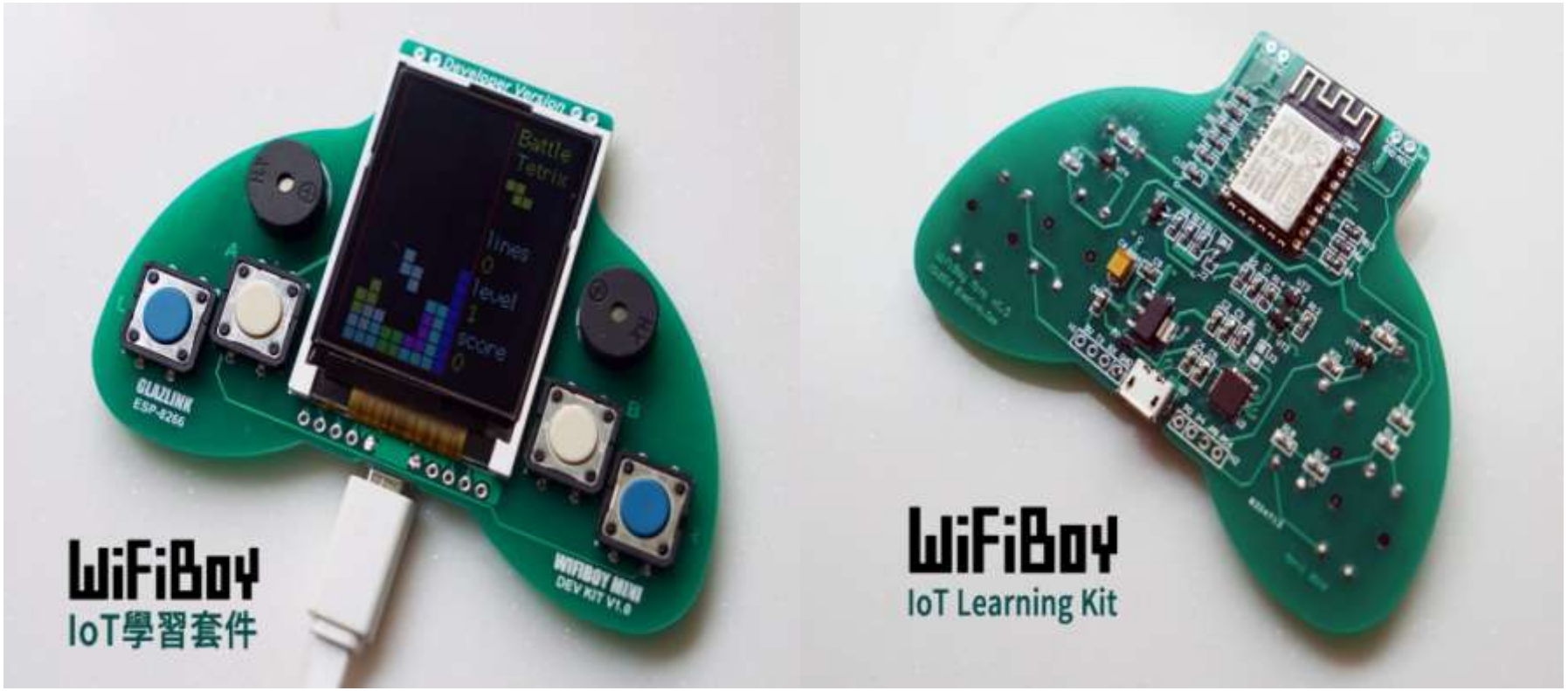
ESP 12E

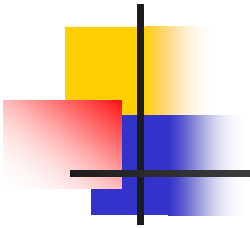

Ai
WiFi
MODEL ESP8266MOD
VENDOR AI-THINKER
ISM 2.4GHz
PA +25dBm
802.11b/g/n
MOORE

NodeMCU


VIA, VIB
VIA, VIB
VIA, VIB
PWM
PWM
PWM
PWM
High Side / Low Side
High Side / Low Side
High Side / Low Side
2.5
2.5
5.0
2.5

WiFiBoy


Developer Version
Battle Tetris
lines 0
level 1
score 0
GLAZLINK
ESP-8266
WiFiBoy
IoT學習套件
WIFIBOY MINI
DEV KIT V1.0
WiFiBoy
IoT Learning Kit

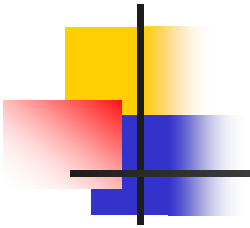


Local Area Network

- 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.

Soft Access Point

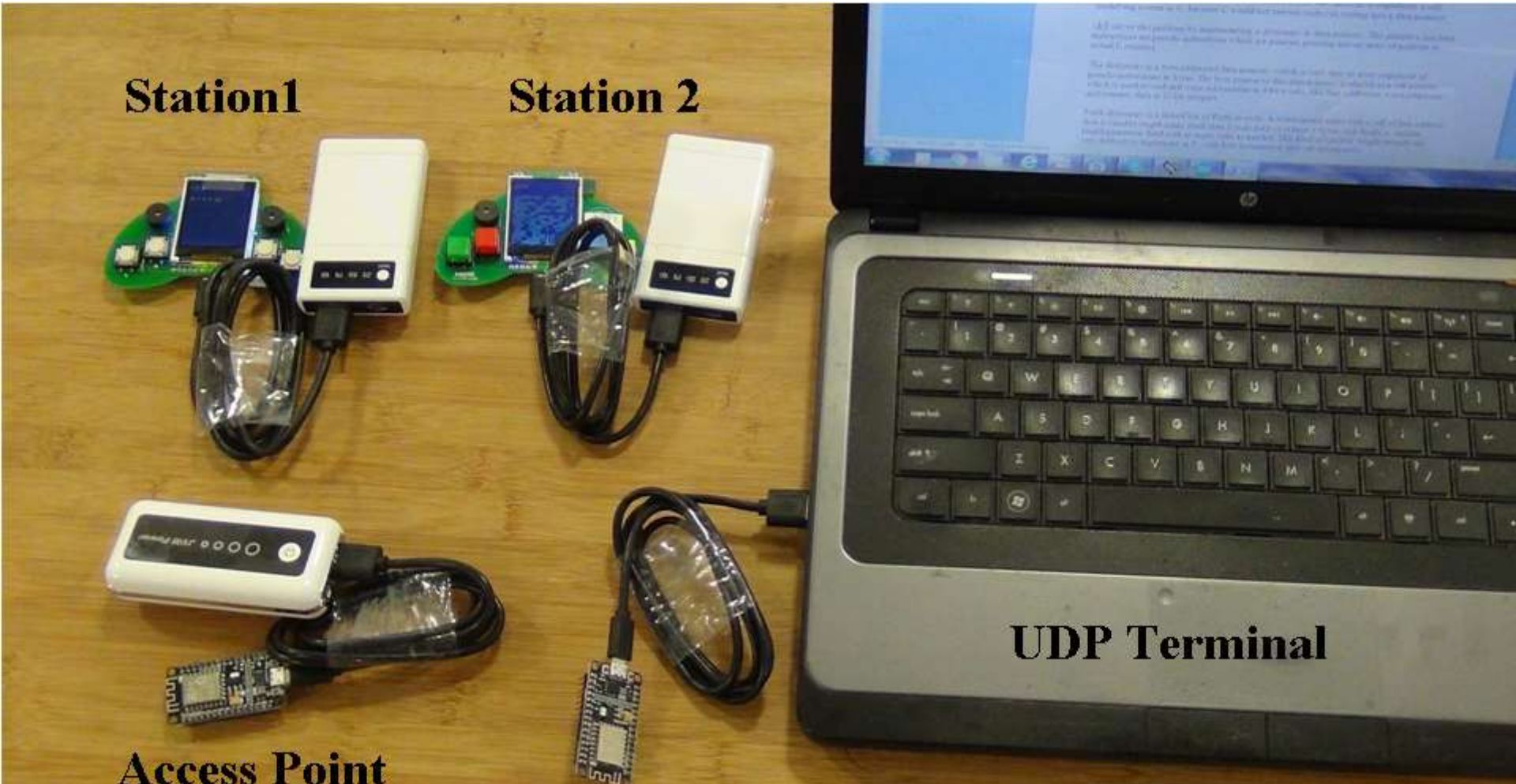
ESP8266 operating in the **Soft Access Point** mode



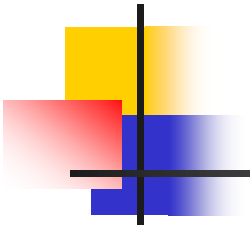
Demo

- 1 ESP8266 set up as an Access Point.
- 3 ESP8266 set up as Stations to receive Forth commands.
- PC serves as a UDP terminal, sends out Forth commands to Stations.

Local Area Network



Access Point



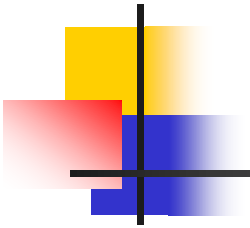
espForth on a Station

- On power-up, connect to LAN Access Point and display IP address and port number.
- Receive commands from Serial Terminal and UDP packets.
- Send text strings to Serial Terminal, UDP client, and optional LCD display.

Station 1

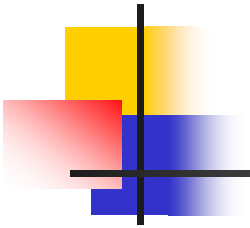

Game Developer Edition
words
IMMEDIATE COMPILE-O
CONSTANT VARIABLE C
REATE CODE "
ABORT" WHILE WHEN
ELSE AFT
REPEAT AHEAD IF A
GAIN UNTIL NEXT BE
GIN FOR THEN COLD
VDP FORGET WORDS
>NAME P
DUMP
OVERT $COMPILE ,COMP
ILE [COMPILE]
?UNIQUE
NAME
2393
.4.5
GLAZLINK
GL-ESP-8266
test
25 50 75 100

Station 1, Close-up



ESP8266 IDE

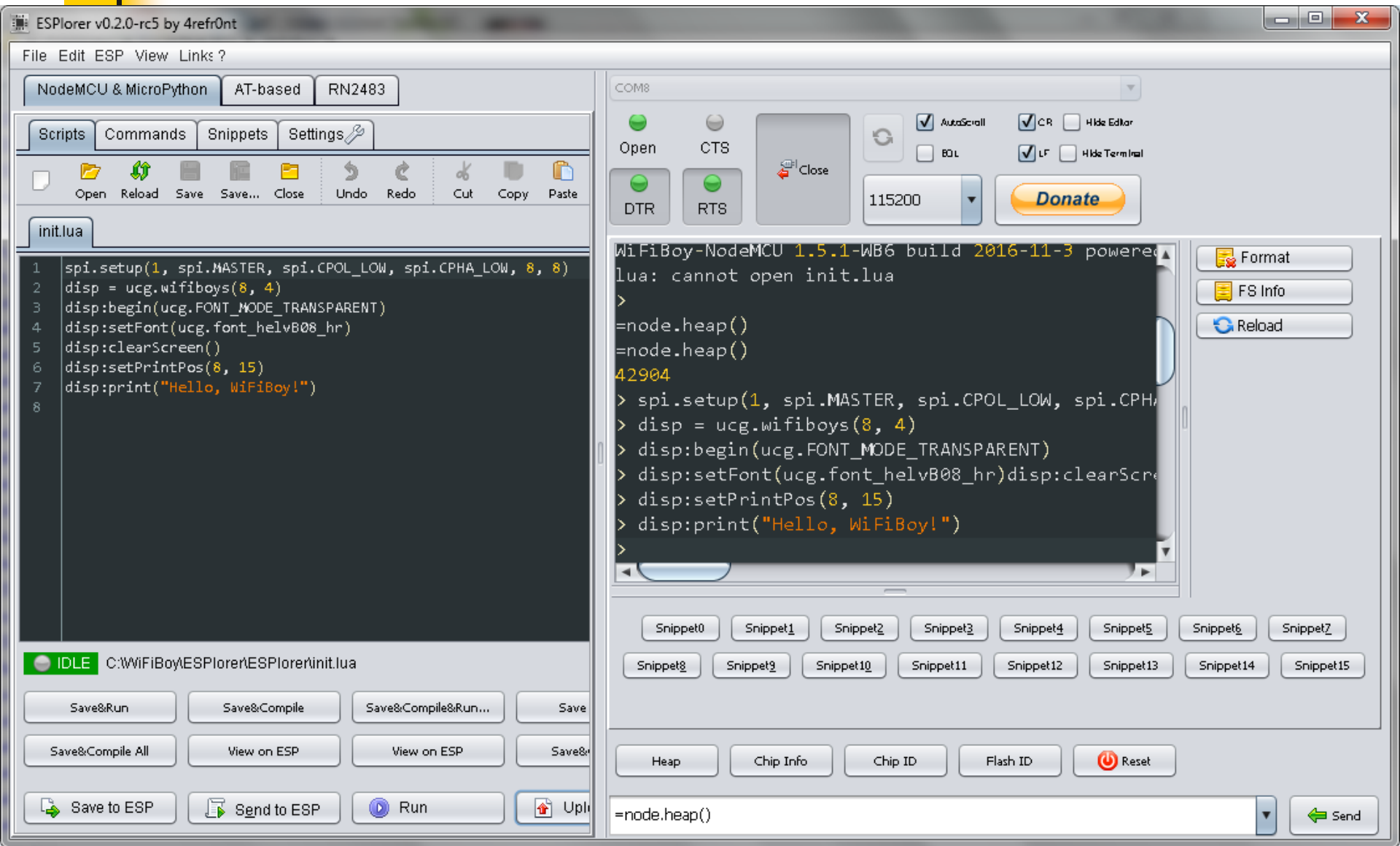
- Lua programming language with ESPlorer IDE
- Arduino IDE with C-like language and flashing tool
- Micropython with esptool and WebREPL interpreter

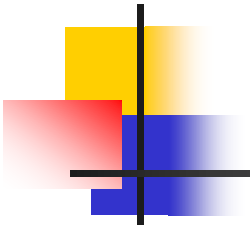


ESPlorer IDE

- You can send Lua commands directly to ESP8266.
- You can edit a file in the Editor panel.
- You can upload a file to ESP8266 flash memory.
- You can run a file in flash memory.

ESPlorer IDE

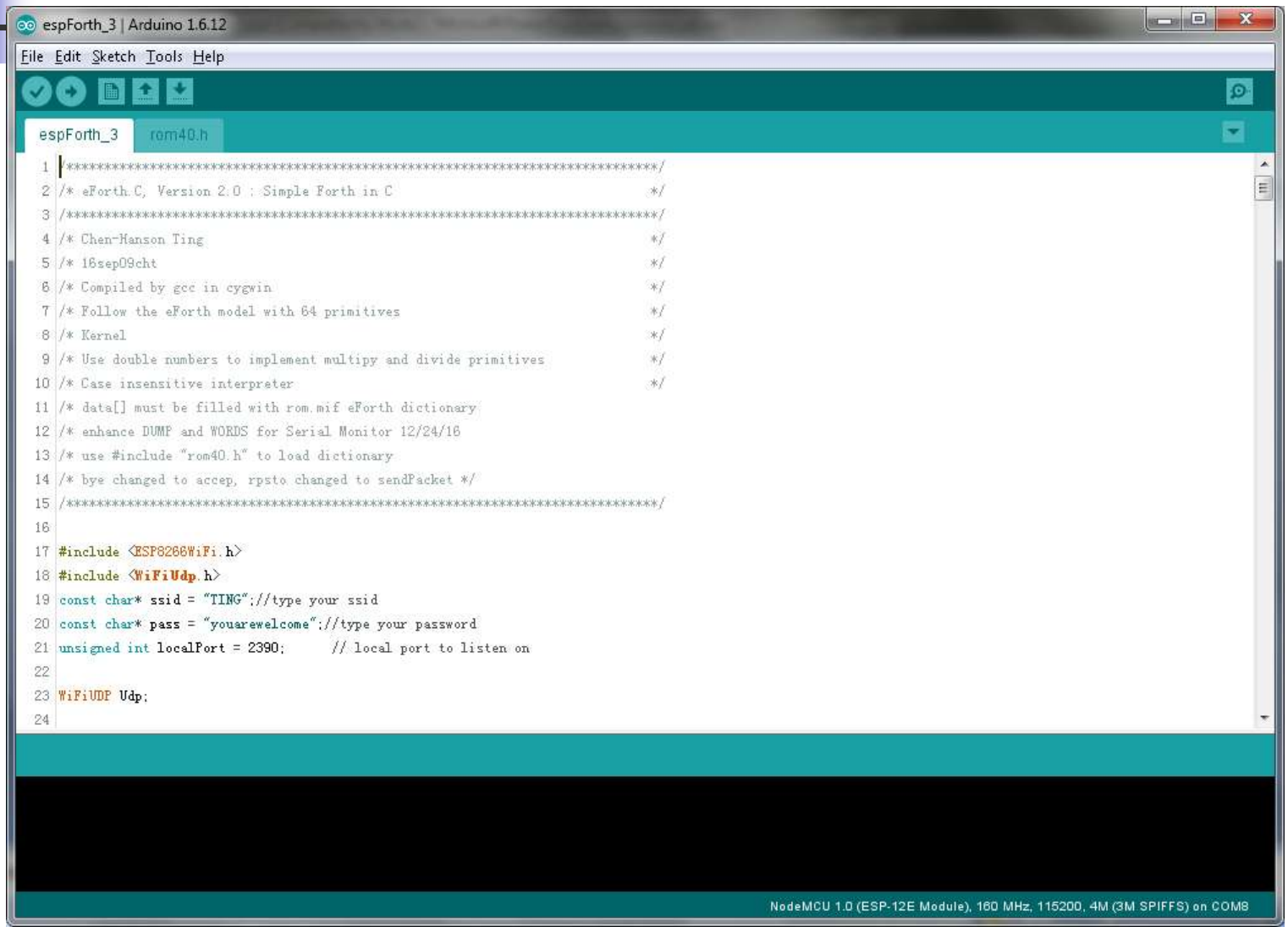


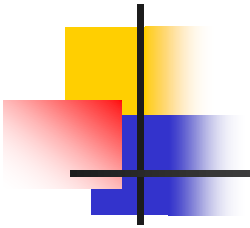


Arduino IDE

- You can load a sketch into editor window.
- A sketch is a C-like file with setup() and loop() routines.
- You can compile a sketch and upload it to ESP8266, and it gets executed.

Arduino IDE

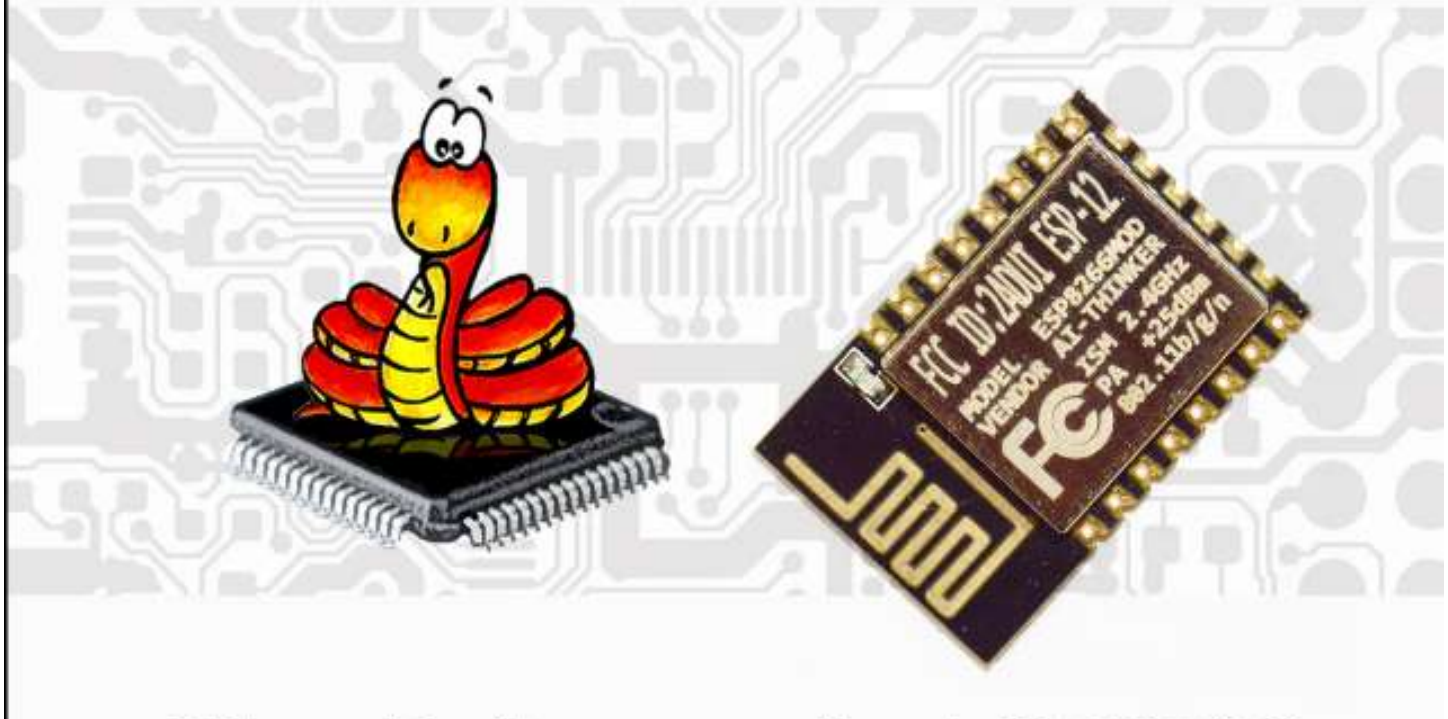

espForth_3 | Arduino 1.6.12
File Edit Sketch Tools Help
espForth_3
rom40.h
/***/
/* eForth.C, Version 2.0 : Simple Forth in C */
/***/
/* Chen-Hanson Ting */
/* 16sep09cht */
/* Compiled by gcc in cygwin */
/* Follow the eForth model with 64 primitives */
/* Kernel */
/* Use double numbers to implement multipy and divide primitives */
/* Case insensitive interpreter */
/* data[] must be filled with rom.mif eForth dictionary
/* enhance DUMP and WORDS for Serial Monitor 12/24/16
/* use #include "rom40.h" to load dictionary
/* bye changed to accep, rpsto changed to sendPacket */
/***/
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
const char* ssid = "TING";//type your ssid
const char* pass = "youarewelcome";//type your password
unsigned int localPort = 2390; // local port to listen on
WiFiUDP Udp;
NodeMCU 1.0 (ESP-12E Module), 160 MHz, 115200, 4M (3M SPIFFS) on COM8



Micropython/WebREPL

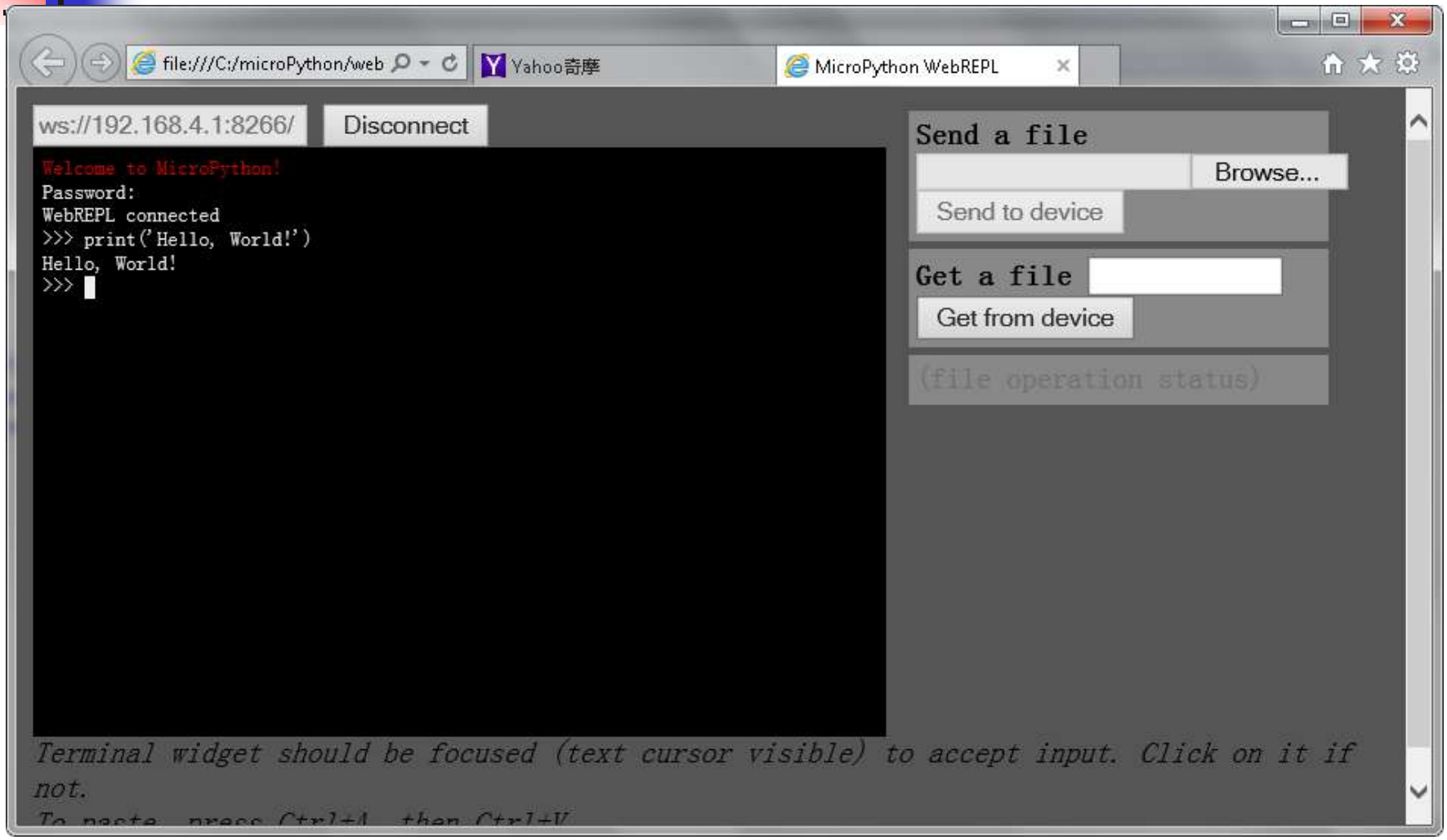
- Micropython is a powerful interpreter for ESP8266.
- WebREPL (Read-Evaluate-Print Loop) is a graphic interface on PC for ESP8266.
- You can download/upload files to/from ESP8266, and interpret them in ESP8266.

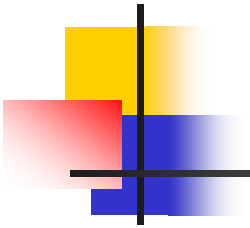
Micropython



MicroPython on the ESP8266

Micropython/WebREPL





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?