ESP32 Workshop

C. H. Ting

May 19, 2019

SVFIG

Welcome!

- San Mateo is the best town in the whole world.
- Not too hot, not too cold, not too wet, not too dry.
- We have the best Chinese restaurants outside of San Francisco.
- We have Maker Faire!

Chinese Restaurants on the 25th Avenue

- **Little Shanghai, 17 E 25th Ave.**
- **Fortune Star, 173 W 25th Ave.**
- **Chef Zhao Bistro, 2450 El Camino Real. Can be really hot!**
- **Chairman Mao’s Kitchen (The Noodle Shop) 164 W 25th Ave.**

Summary

- Affordable microcontroller kits
- NodeMCU ESP32S kit
- Arduino IDE
- Plan-B robot
- Serial Monitor interface
- Web Browser interface
- Bach benchmark

Affordable Microcontroller Kits

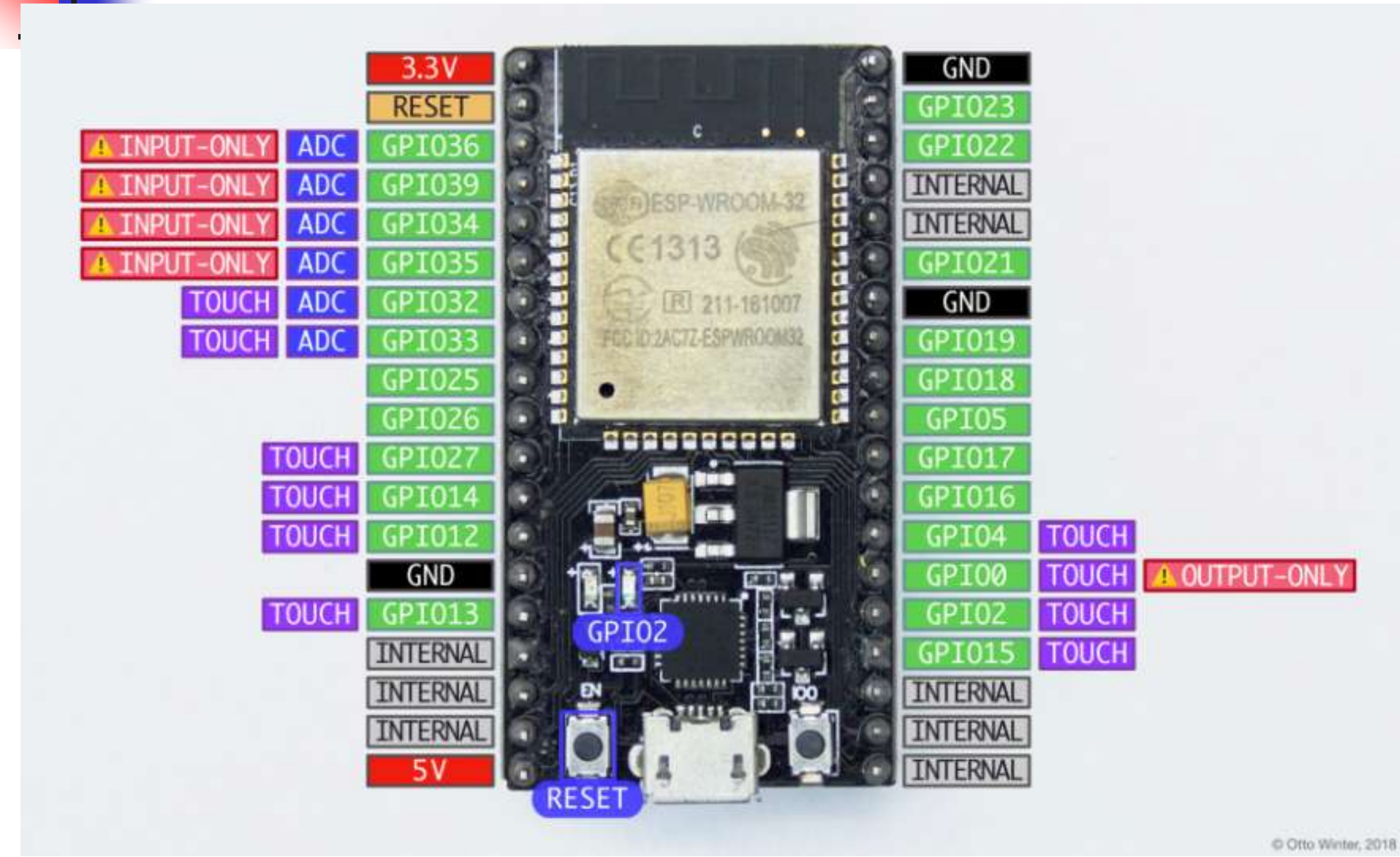
- 2005 ADuC7020
- 2009 STM8 Discovery
- 2011 TI LaunchPad MSP430
- 2014 STM32 Discovery
- 2015 Arduino Uno ATmega328P
- 2017 NodeMCU ESP8266
- 2019 NodeMCU ESP32

ESP32

- Dual 32-bit Xtensa LX106, 240 MHz
- 520 KB SRAM, 4 MB flash
- 28 GPIO pins, 16 PWM channels
- 8 ADC, 2 DAC, 3 UART, 2 SPI, 3 I2C
- WiFi: IEEE 802.11 b/g/n/e/I
- Bluetooth

NodeMCU ESP32S

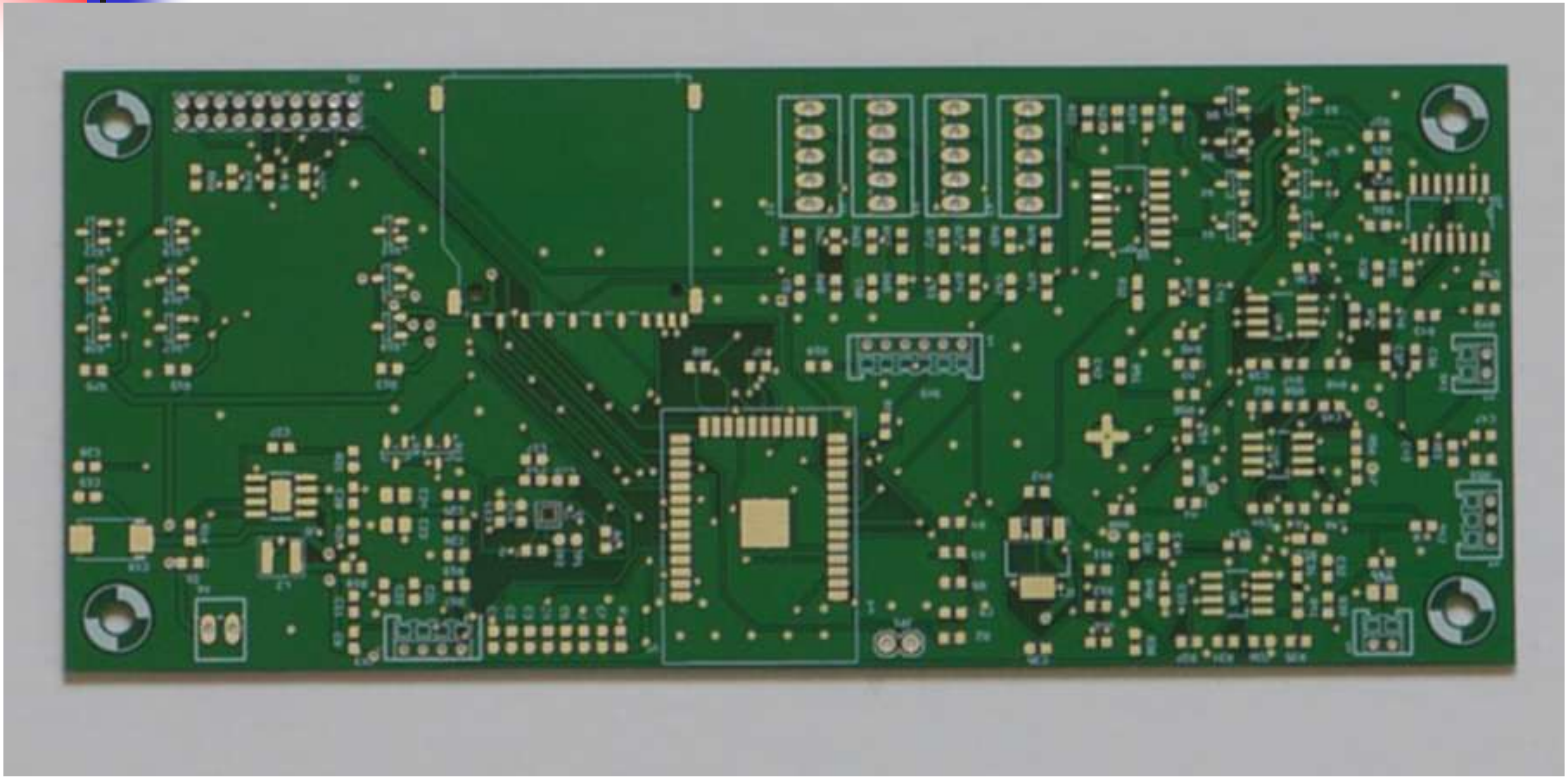
NodeMCU ESP32S



© Otto Winter, 2018

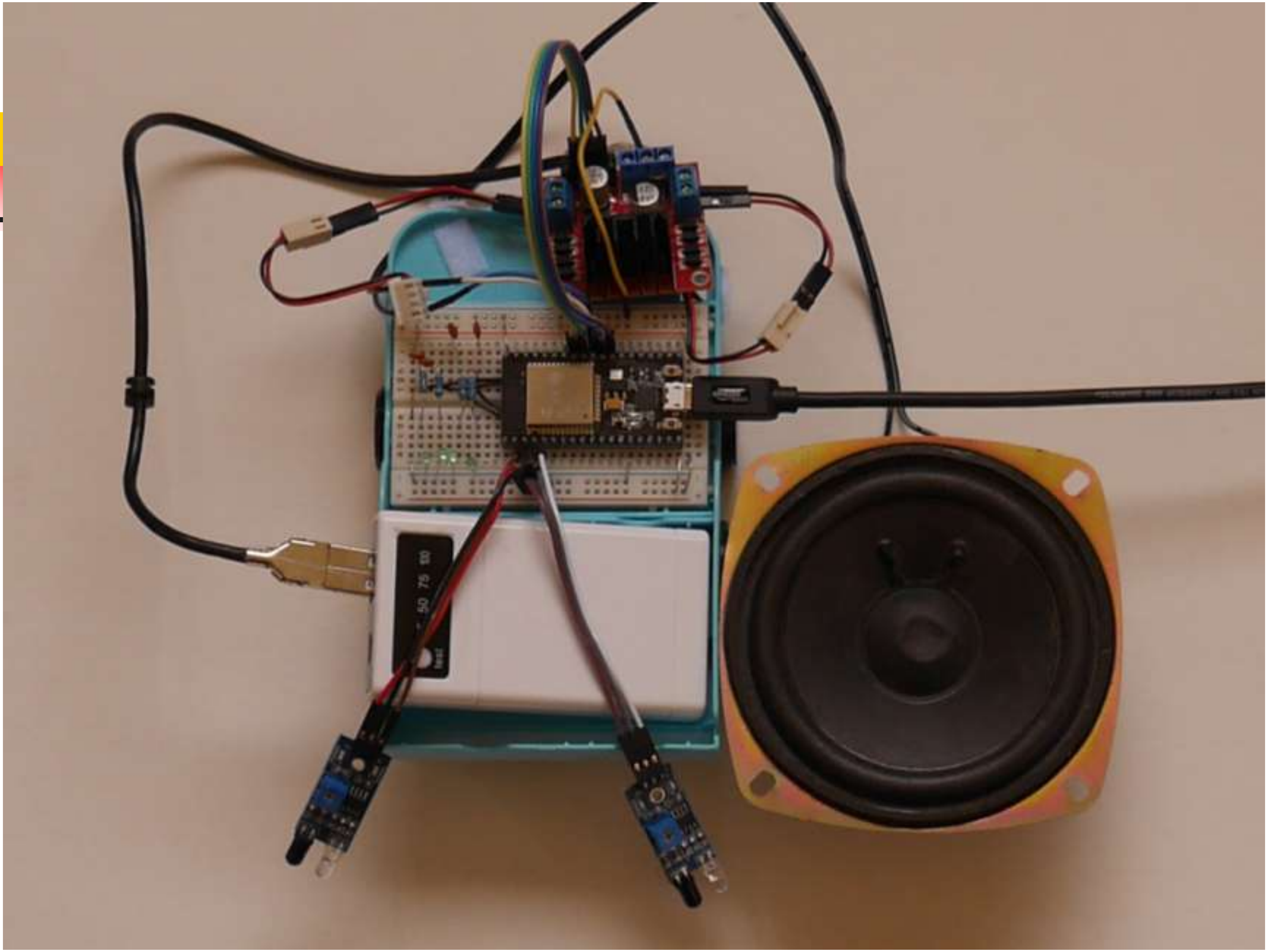
AIR-AI Robot

- Ron Golding is building his AIR Robot for Maker Faire.
- So far, I have seen only a bare PC board.



Plan-B Robot

- I got a number of robot car kits from Taiwan FIG.
- I assembled two robot car kits for Maker Faire.
- Both use NodeMCU ESP32S kits as controllers.
- I tested motors, obstacle sensors, and speaker.


100
75
50
test

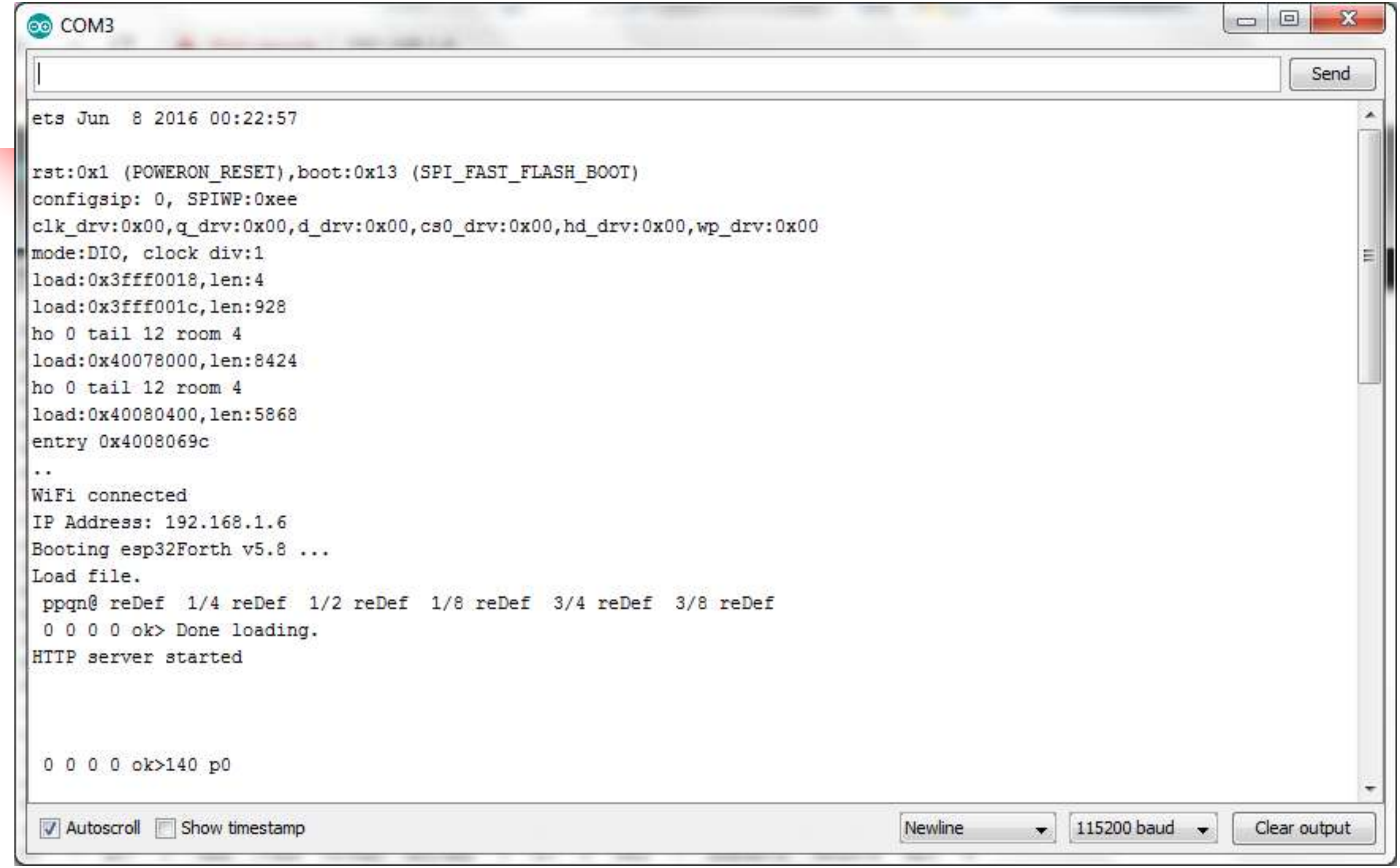
ESP32forth

- ESP32forth emulates eP32, a 32-bit Forth microcontroller.
- Virtual Forth Machine executes byte code, and is written in C as a sketch.
- ESP32forth Finite State Machine:

```
{primitives[char cData[P++]]();}
```

Serial Monitor Interface

- NodeMCU ESP32S is connected to Arduino IDE through a microUSB cable.
- User test and program ESP32 through the Serial Monitor at 11520 baud.
- Turnkey application is placed in load.txt file stored in flash.



```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:928
ho 0 tail 12 room 4
load:0x40078000,len:8424
ho 0 tail 12 room 4
load:0x40080400,len:5868
entry 0x4008069c
..
WiFi connected
IP Address: 192.168.1.6
Booting esp32Forth v5.8 ...
Load file.
 ppqn@ reDef  1/4 reDef  1/2 reDef  1/8 reDef  3/4 reDef  3/8 reDef
 0 0 0 0 ok> Done loading.
HTTP server started



 0 0 0 0 ok>140 p0
```

☑ Autoscroll □ Show timestamp

Newline

115200 baud

Robot Control

`FORE`	**Drive robot forward**
`BACK`	**Drive robot backward**
`LEFT`	**Drive robot left**
`RIGHT`	**Drive robot right**
`STOP`	**Stop robot**
`LED`	**Turn on blue LED**

Lesson 1

```
( Lesson 1. The Universal Greeting )
: HELLO CR ." Hello, world!" ;
```

Lesson 2

```
( Lesson 2. The Big F )
: bar   CR ." *****" ;
: post  CR ." *    " ;
: F     bar post bar post post post ;
```

Lesson 3

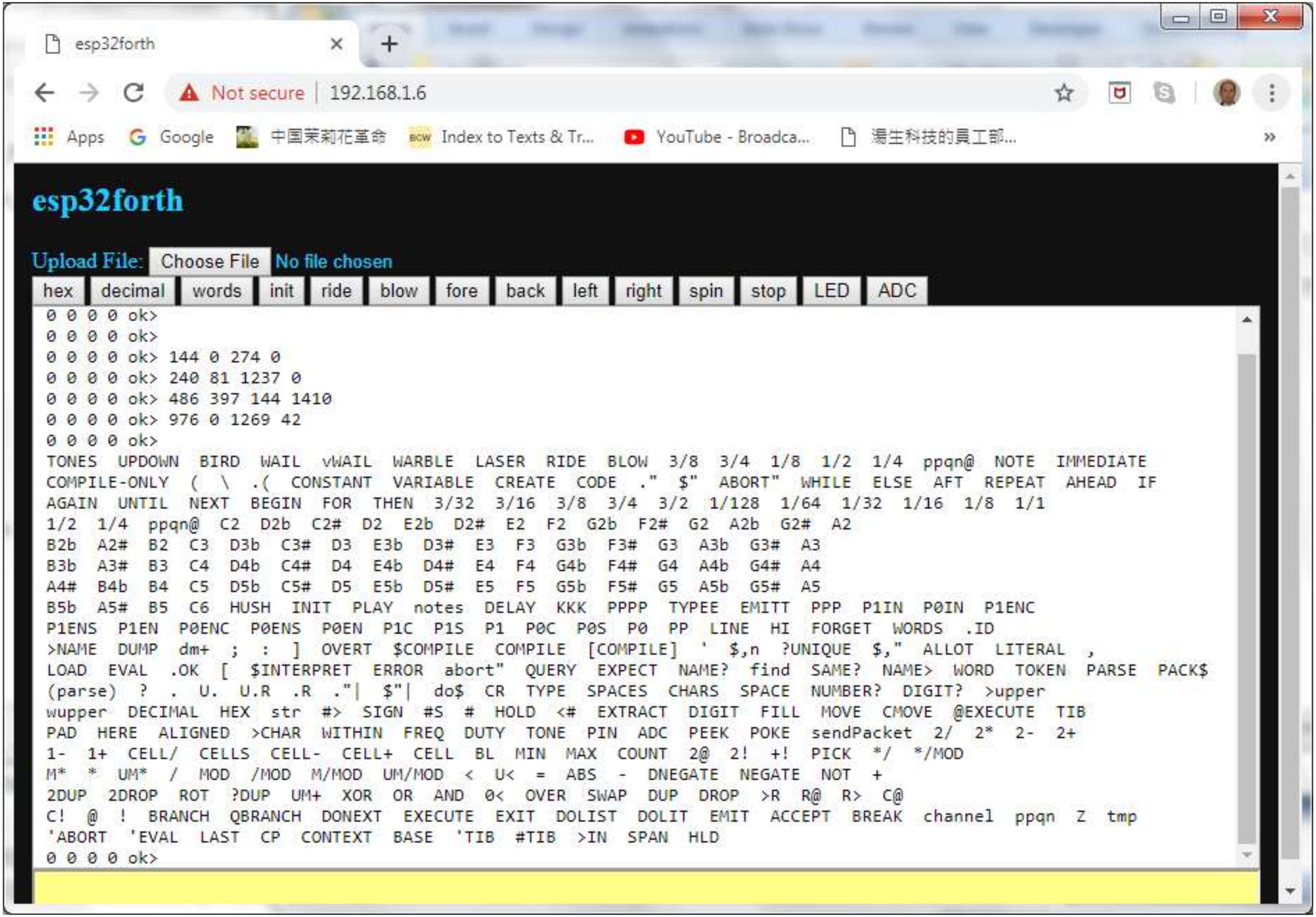
```
( Lesson 3.      FIG, Forth Interest Group )
: center CR ."   *  " ;
: sides  CR ." *   *" ;
: triad1 CR ." * * *" ;
: triad4 CR ."  *** " ;
: right  CR ." * ***" ;
: bigI  center center center center center center
  center ;
: bigG  triad4 sides post right triad1 sides triad4 ;
: FIG  F bigI bigG ;
```

Web Browser Interface

- I was using very simple HTTP protocol to send Forth commands to the robot.
- Brad Nelson changed it to Web server and implemented a true browser to send commands and to download text files.

Web Browser Interface

- ESP32forth is extended so that user can operate ESP32 through HTTP web browser.
- All interactive interpreter and compiler functions
- Direct robot control
- Source code file download



```
0 0 0 0 ok>
0 0 0 0 ok>
0 0 0 0 ok> 144 0 274 0
0 0 0 0 ok> 240 81 1237 0
0 0 0 0 ok> 486 397 144 1410
0 0 0 0 ok> 976 0 1269 42
0 0 0 0 ok>
TONES  UPDOWN  BIRD  WAIL  vWAIL  WARBLE  LASER  RIDE  BLOW  3/8  3/4  1/8  1/2  1/4  ppqn@  NOTE  IMMEDIATE
COMPILE-ONLY  (  \  .(  CONSTANT  VARIABLE  CREATE  CODE  ."  $"  ABORT"  WHILE  ELSE  AFT  REPEAT  AHEAD  IF
AGAIN  UNTIL  NEXT  BEGIN  FOR  THEN  3/32  3/16  3/8  3/4  3/2  1/128  1/64  1/32  1/16  1/8  1/1
1/2  1/4  ppqn@  C2  D2b  C2#  D2  E2b  D2#  E2  F2  G2b  F2#  G2  A2b  G2#  A2
B2b  A2#  B2  C3  D3b  C3#  D3  E3b  D3#  E3  F3  G3b  F3#  G3  A3b  G3#  A3
B3b  A3#  B3  C4  D4b  C4#  D4  E4b  D4#  E4  F4  G4b  F4#  G4  A4b  G4#  A4
A4#  B4b  B4  C5  D5b  C5#  D5  E5b  D5#  E5  F5  G5b  F5#  G5  A5b  G5#  A5
B5b  A5#  B5  C6  HUSH  INIT  PLAY  notes  DELAY  KKK  PPPP  TYPEE  EMITT  PPP  P1IN  P0IN  P1ENC
P1ENS  P1EN  P0ENC  P0ENS  P0EN  P1C  P1S  P1  P0C  P0S  P0  PP  LINE  HI  FORGET  WORDS  .ID
>NAME  DUMP  dm+  ;  :  ]  OVERT  $COMPILE  COMPILE  [COMPILE]  '  $,n  ?UNIQUE  $,"  ALLOT  LITERAL  ,
LOAD  EVAL  .OK  [  $INTERPRET  ERROR  abort"  QUERY  EXPECT  NAME?  find  SAME?  NAME>  WORD  TOKEN  PARSE  PACK$
(parse)  ?  .  U.  U.R  .R  ."|  $"|  do$  CR  TYPE  SPACES  CHARS  SPACE  NUMBER?  DIGIT?  >upper
wupper  DECIMAL  HEX  str  #>  SIGN  #S  #  HOLD  <#  EXTRACT  DIGIT  FILL  MOVE  CMOVE  @EXECUTE  TIB
PAD  HERE  ALIGNED  >CHAR  WITHIN  FREQ  DUTY  TONE  PIN  ADC  PEEK  POKE  sendPacket  2/  2*  2-  2+
1-  1+  CELL/  CELLS  CELL-  CELL+  CELL  BL  MIN  MAX  COUNT  2@  2!  +!  PICK  */  */MOD
M*  *  UM*  /  MOD  /MOD  M/MOD  UM/MOD  <  U<  =  ABS  -  DNEGATE  NEGATE  NOT  +
2DUP  2DROP  ROT  ?DUP  UM+  XOR  OR  AND  0<  OVER  SWAP  DUP  DROP  >R  R@  R>  C@
C!  @  !  BRANCH  QBRANCH  DONEXT  EXECUTE  EXIT  DOLIST  DOLIT  EMIT  ACCEPT  BREAK  channel  ppqn  Z  tmp
'ABORT  'EVAL  LAST  CP  CONTEXT  BASE  'TIB  #TIB  >IN  SPAN  HLD
0 0 0 0 ok>
```

Robot Control Buttons

INIT	Initialize speaker
RIDE	Speaker demo
FORE	Drive robot forward
BACK	Drive robot backward
LEFT	Drive robot left
RIGHT	Drive robot right
STOP	Stop robot
LED	Turn on blue LED
ADC	Test 4 analog inputs

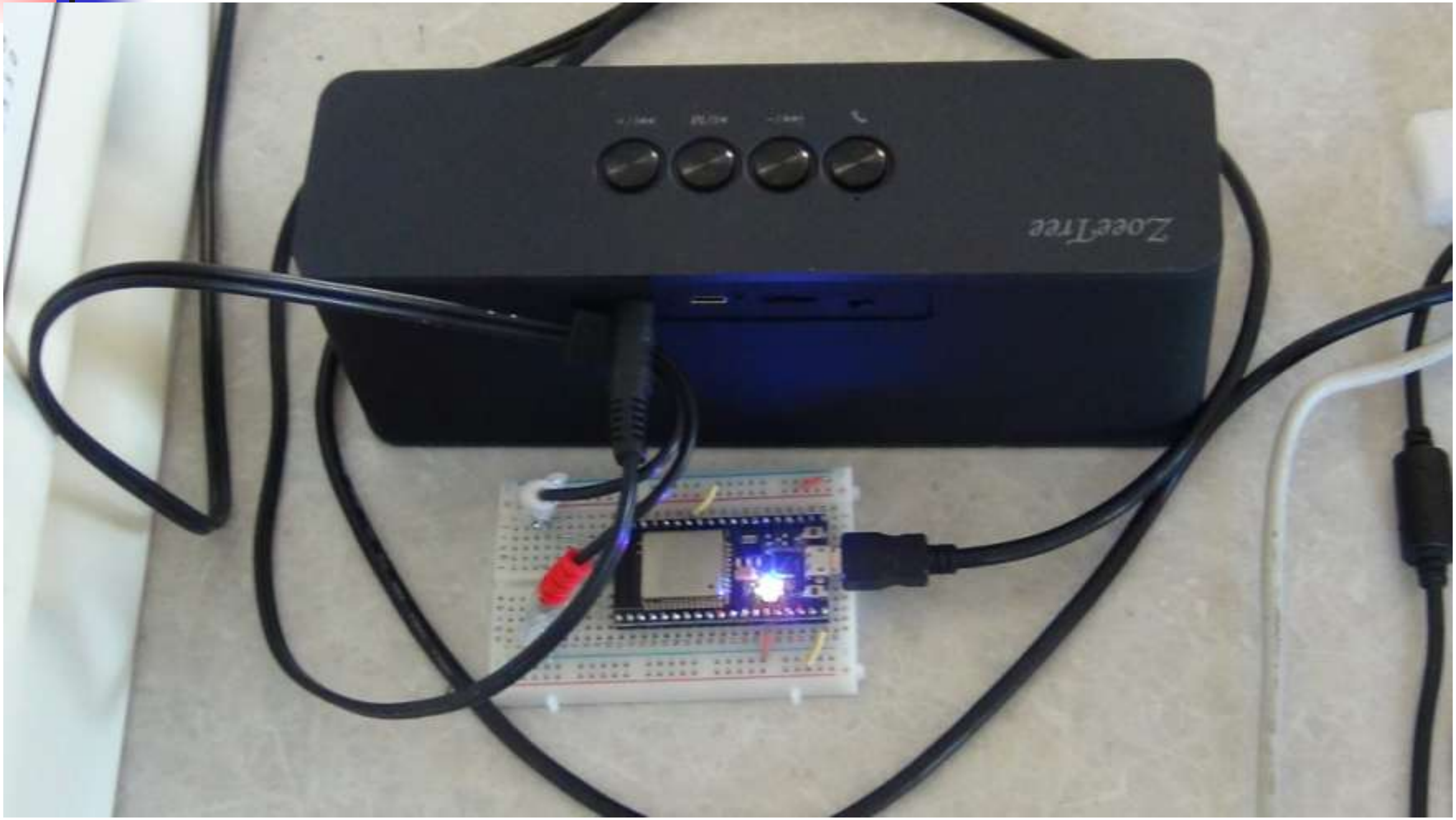
Bach Benchmark

- In 1984, a friend in Taiwan FIG gave me an PC IO Card with 4 8253s and 4 8255s, for machine automation.
- I built a 12 channel electronic organ, and played many organ pieces by Bach.

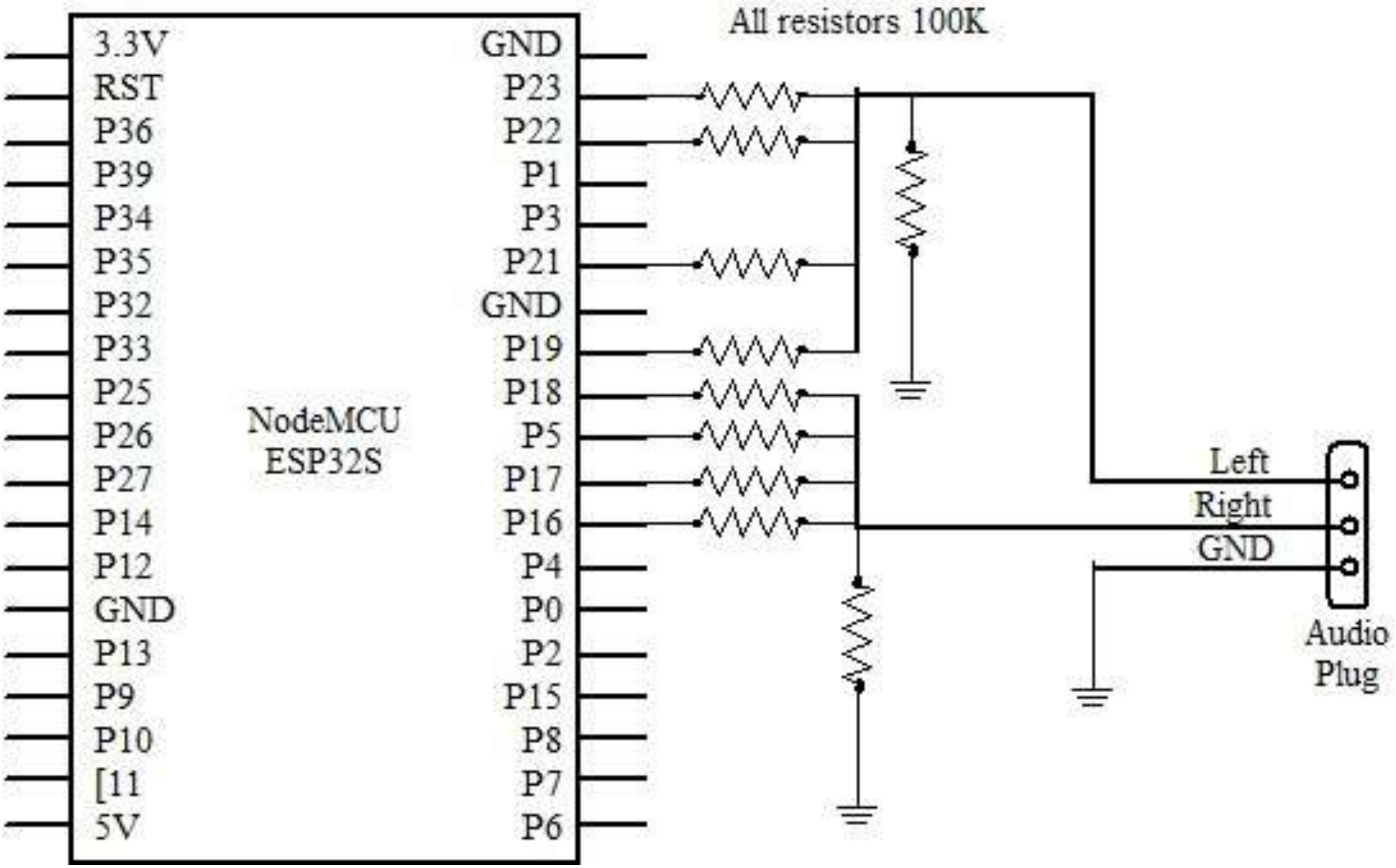
Bach Benchmark

- I tried to play Bach organ music on most of the microcontrollers I worked with.
- Only Arduino Uno could play three-voice organ pieces.
- ESP32 can play 8 voices, and passes my Bach Benchmark.

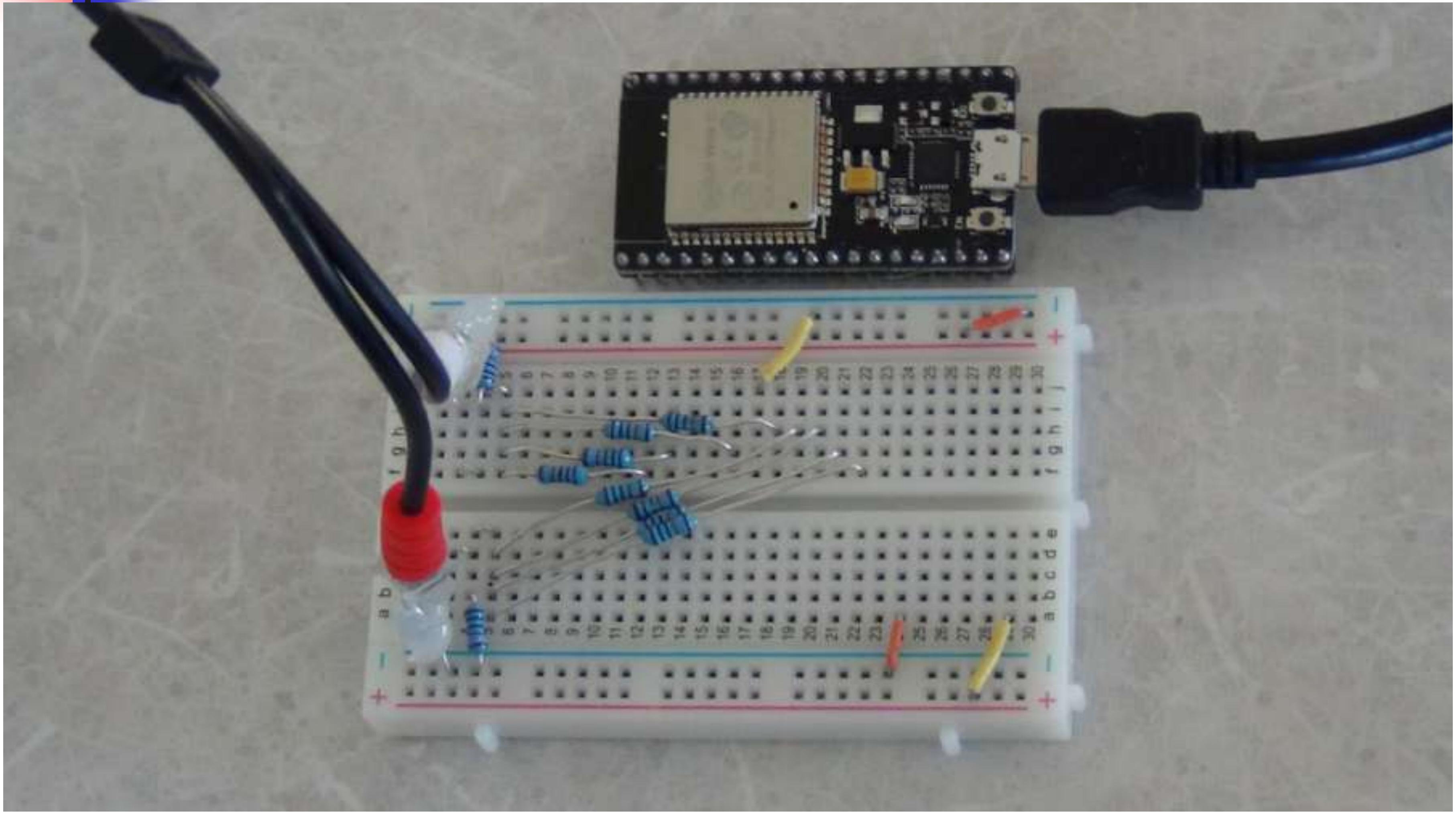
Electronic Organ



Electronic Organ



Electronic Organ



Organ Demo

- 8 Channel electronic organ
- 8 Digital outputs are summed to a left voice and a right voice through an array of 100KΩ resistors.
- Left and right voices are amplified by a speaker.
- A musette dance and a fugue.

Questions?

Thank you.