



Socket Programming

Chen-Hanson Ting

SVFIG

March 25, 2017



Summary

- Socket Programming
- MicroPython
- Python 3.6
- Access point
- Concerto
- Demo



Socket Programming

- Lowest level of network programming
- Create sockets
- Send packets
- Receive packets
- Process packets



MicroPython

- ESP8266 implementation
- Machine - Pin, PWM
- Network - WLAN
- Socket
- REPL, WebREPL
- File management
- Concerto Players



Python 3.6

- PC resident language IDE
- IDLE
- Socket Programming
- Concerto Director





Access point

- Router of a local network
- Local network configuration
- Servers
- Clients



Concerto Players

- Tone generator
- Packet receiver
- Static sockets
- Packet processor



Tone generator

```
import machine
p14=machine.PWM(machine.Pin(14, 1))
def tone(n):
    if n:
        p14.duty(512)
        p14.freq(n)
    else:
        p14.duty(0)
```



Static Sockets

```
import network
sta=network.WLAN(network.STA_IF)
sta.active(1)
sta.connect('ESPsoftAP','12345678')
sta.ifconfig(('192.168.4.11','255.255.255.
0','192.168.4.1','192.168.4.1'))
newconfig=sta.ifconfig()
print(newconfig)
```



Packet Receiver

```
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.setsockopt(socket.SOL_SOCKET,
             socket.SO_REUSEADDR, 1)
addr=(newconfig[0],10001)
print(addr)
s.bind(addr)
```



Packet Processor

```
def listen():  
    while True:  
        data,address=s.recvfrom(10)  
        tone(int(data))  
        #    print(data,address)  
listen()
```



Concerto Director

- Player sockets
- Tone table
- Music notes
- Music chords
- Director



Player Sockets

```
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
```

```
s.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEADDR, 1)
```

```
def note(array,n,host):
```

```
    freq=f[array[n]]
```

```
    s.sendto(str.encode(str(freq)),host)
```

```
    time.sleep(8*dt)
```



Player Sockets

```
def stop():  
    s.sendto(str.encode(str(0)),HOST1)  
    s.sendto(str.encode(str(0)),HOST2)  
    s.sendto(str.encode(str(0)),HOST3)  
    s.sendto(str.encode(str(0)),HOST4)
```

Das wohltemperier

1. Preludium in

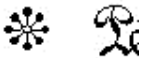
Allegro

legato

Klavier I

The musical score for Klavier I shows the beginning of the 1. Preludium in C major. It is written in common time (C) and consists of two staves: a treble clef staff and a bass clef staff. The tempo is marked 'Allegro' and the articulation is 'legato'. The piece begins with a piano (p) dynamic. The treble clef part starts with a quarter rest, followed by a series of eighth and sixteenth notes. The bass clef part starts with a half note, followed by a series of eighth and sixteenth notes.

Leo.





Tone table

```
f=[round(55*2**(x/12)) for x in  
    range(51)]
```

```
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,
```



Music chords

```
def chord(n):  
    for i in range(1):  
        note(mel,n,HOST3)  
        note(mel,n+1,HOST1)  
        note(mel,n+2,HOST2)  
        note(mel,n+3,HOST2)  
        note(mel,n+4,HOST2)  
        note(mel,n+2,HOST2)  
        note(mel,n+3,HOST2)  
        note(mel,n+4,HOST2)
```



Chord Sequence

```
def play():  
    j=0  
    for i in range(32):  
        chord(j)  
        stop()  
        j=j+5
```



Concerto Director

```
def prelude():  
    play()  
    last()  
    last1()  
    time.sleep(8*dt)  
    stop()  
prelude()
```



Demo

- Access point
- 3 players
- PC as Director
- C Major Prelude (BWV846)