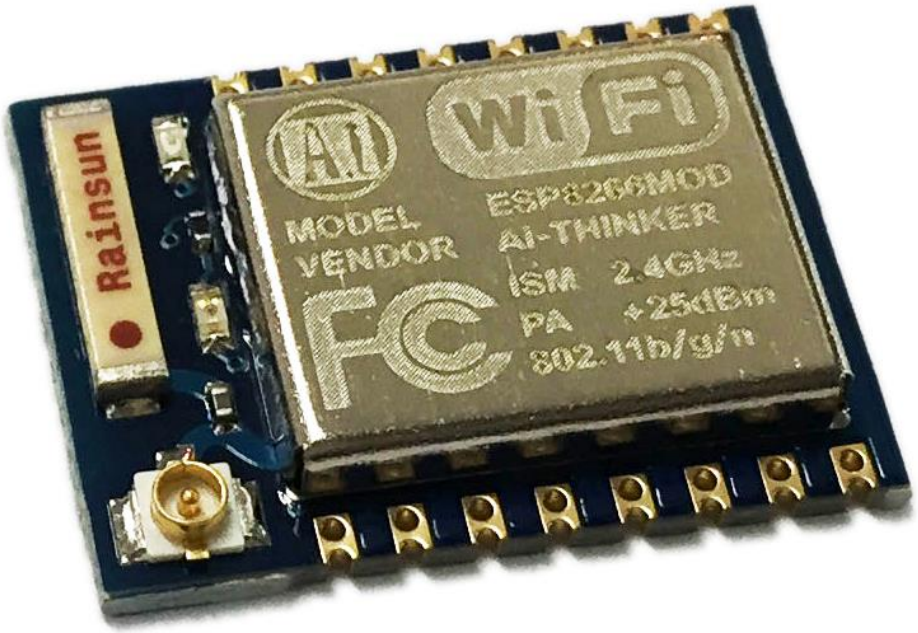
ESP8266

Wi-Fi modul za $3

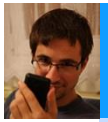
SUT, 28. 4. 2015

Adam Hořčica, Vojtěch Suk

Ahoj, já jsem
ESP8266



Kdo jsme…



Adam

- Bastlíř
- Programátor
- Správce projektu MacGyver

Vojta

- Hračička
- Správce projektu MacGyver

bastlíři SH
MacGyver
macgyver.siliconhill.cz

Silicon Hill

bastlíři SH

MacGyver

macgyver.siliconhill.cz

Silicon Hill

… a co vám přinášíme

- **MQTT – *Protože bez IoT by to dnes nebylo cool & in***
  - Co to je, jak to funguje a jak to použít
  - Demo
- **ESP8266 – *HW***
  - Specifikace, zajímavosti, zapojení
  - Demo (ošmatej si sám)
- **ESP8266 – *The easy way***
  - AT, Lua, uPy, Arduino, …
  - Demo
- **ESP8266 – *Pro opravdové ~~muže~~ vývojáře***
  - SDK
  - **MQTT** klient demo

ADAM

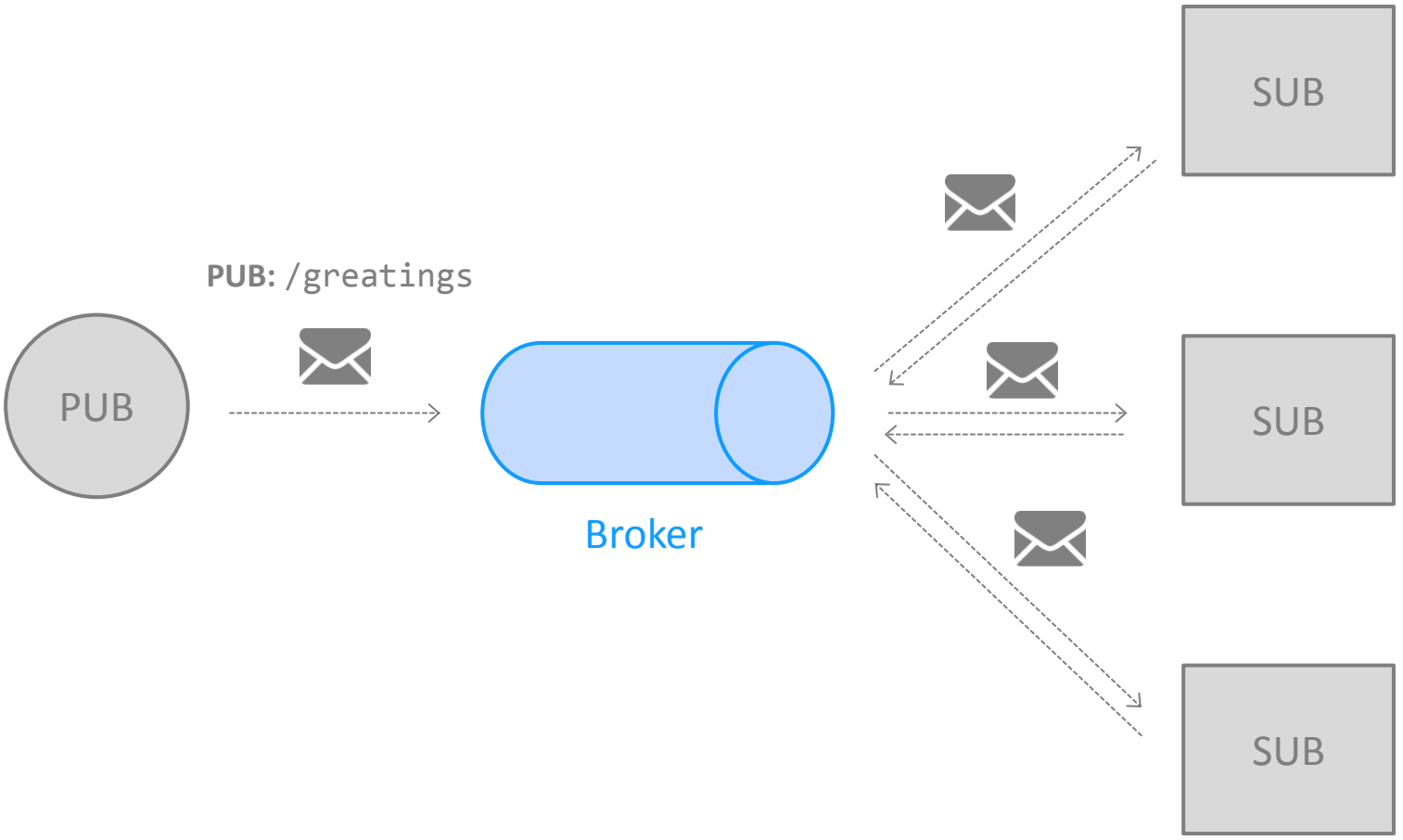
Protokol pro IoT

MQTT

MQTT

- Pub-Sub
- Broker x Client
  - Publikace zprávy s předmětem (topic)
  - Odebírání všech zpráv s daným předmětem
  - Obsah zprávy je TXT
- Malé datové nároky
- Postavené nad TCP/IP

Pub/Sub



Předmět (topic)

- Publikace

  `/hierarchická/struktura/xyz/123`

- Odebírání
  - Přesný předmět:

    `/hierarchická/struktura/xyz`

  - Wildchar:

    `/hierarchická/+/+/xyz`
    `/hierarchická/struktura/#`
    `/+/struktura/#`

Předmět (topic)

```
/sensory/budovaA/mistnost123/teplota

/sensory/budovaA/mistnost123/+
/sensory/budovaA/#
/sensory/budovaA/+/teplota
/sensory/+/+/teplota
```

Quality of Service

- Co se stane se zprávou v případě poruchy:

QoS 0 … nemusí být doručena **vůbec**

QoS 1 … může být doručena **vícekrát**

QoS 2 … bude doručena **právě jednou**

Retain

- Příznak trvalé zprávy
- Broker si ji zapamatuje
- První co dostanu po *sub*

- Je možné smazat zasláním `null` zprávy

„Poslední vůle“ (will)

- *Co se stane, když mě někdo ~~zabije~~ odpojí*
- Zpráva, kterou broker zašle při nečekaném odpojení klienta

- `will-topic`
- `will-payload`
- `will-qos`
- `will-retain`

Mosquitto

- Broker (mosquitto)
- C++, multiplatformní

- Klient:
  - mosquitto_pub
  - mosquitto_sub

- http://mosquitto.org
- Demo: iot.eclipse.org:1883

Node Red

- Grafický jazyk + runtime
- Tok zpráv od vstupu do výstupu
  - API, HW, online service
- Implementace v node.js

- http://nodered.org/
- https://learn.adafruit.com/raspberry-pi-hosting-node-red/what-is-node-red

Demo

- http://iot.siliconhill.cz
- Mosquitto konzolový klient
- MQTT Spy
- MQTT Hive web client
- NodeRed
- *Like-o-meter*

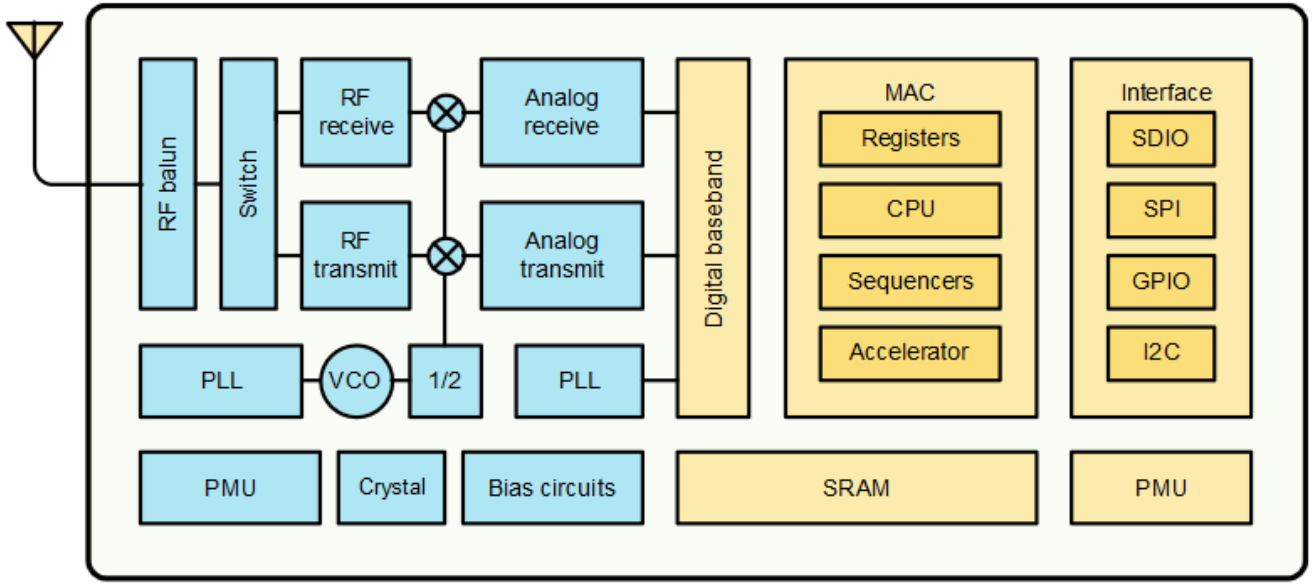
VOJTA

Železo

ESP8266 HW

ESP8266

- WiFi SoC
- 32 bit RISC CPU
- GPIO, I2C, ADC, SPI, PWM, UART, RTC, JTAG



ESP8266

- Vyrábí Espressif Systems
- Postaveno na xTensa od Tensilica
- Moduly od od různých výrobců

ESP8266

- Vyžaduje 3,3 V napájení
- GPIO jsou 5 V tolerantní
- Při příjmu odběr 60-62 mA
- Při vysílání až 215 mA

CPU

- 80 MHz
- 96 kB data RAM
- 64 kB instuction ROM (bootloader od výrobce)
- 64 kB instruction RAM (cache pro SPI flash)
- 4 sleep módy

CPU

- Boot módy
- Na daných pinech nutný PullUp

MTDO	GPIO0	GPIO2	Mode	Description
L	L	H	UART	Download code from UART
L	H	H	Flash	Boot from SPI Flash
H	x	x	SDIO	Boot from SD-card

WiFi

- Podpora 802.11b/g/n (2.4 GHz)
- WPA/WPA2 (pouze PSK)
- Až 20 dBm (při 802.11b, jinak 16 dBm)
- Antena diversity

Periferie

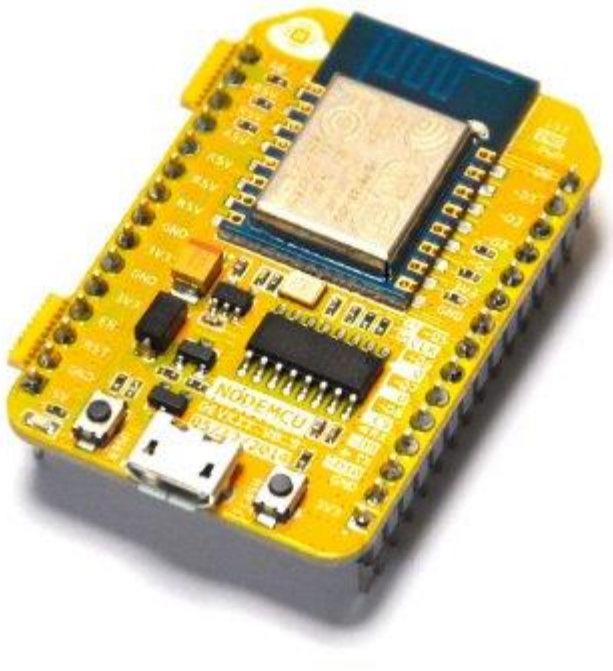
- Až 16 GPIO pinů
- 10 bitový ADC
- SPI (použitá pro externí FLASH paměť)
- UART
- RTC
- SDIO 2.0

Dostupný HW

- eBay, Alibaba,…
- Liší se jen počtem GPIO a anténou

Dostupný HW

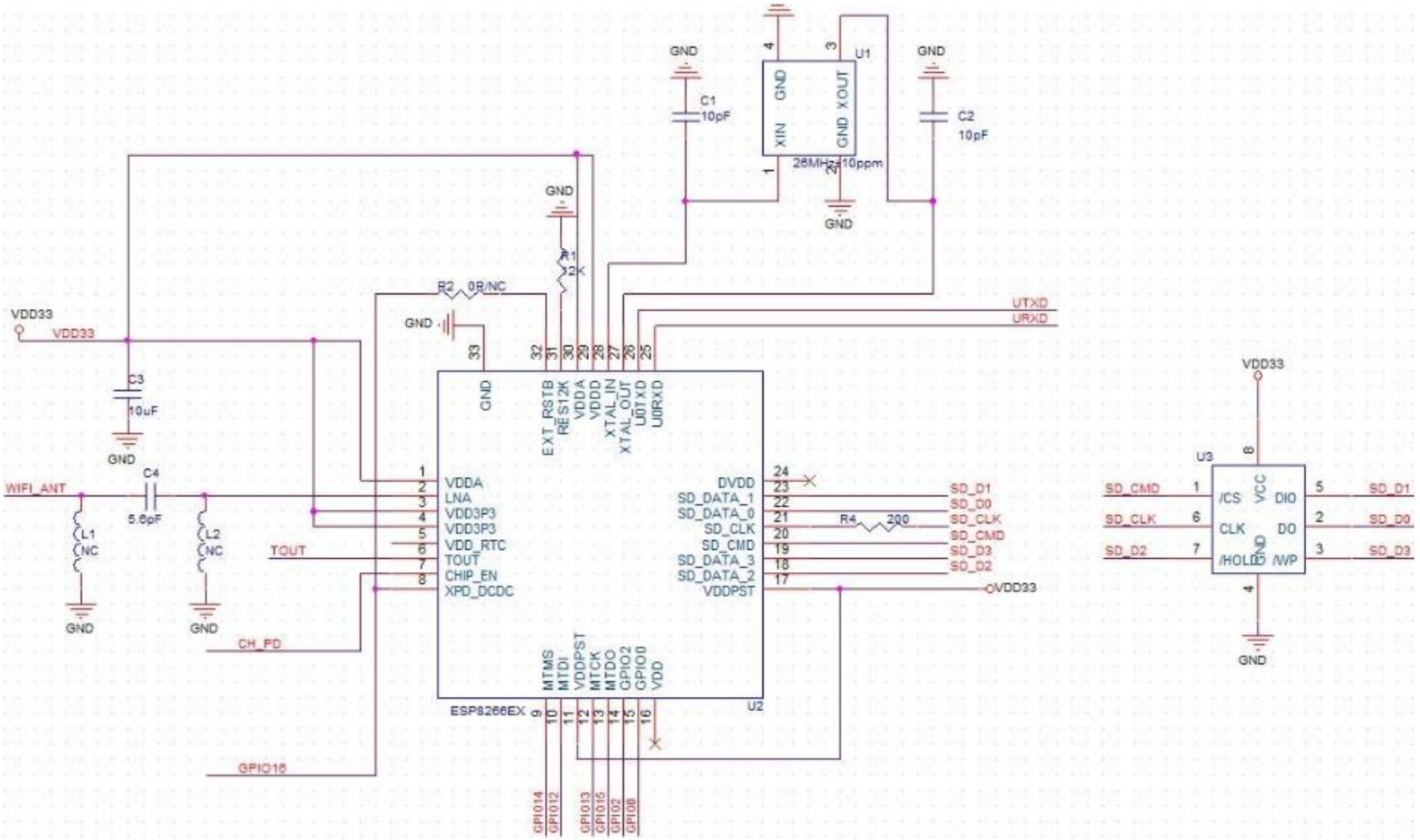
- U většiny modulů je potřeba další PCB
- Nebo použít ESP-12 development board



Vlastní HW

- Stačí jen několik málo součástek
- QFN32 pouzdro
- Impedanční přizpůsobení

Vlastní HW

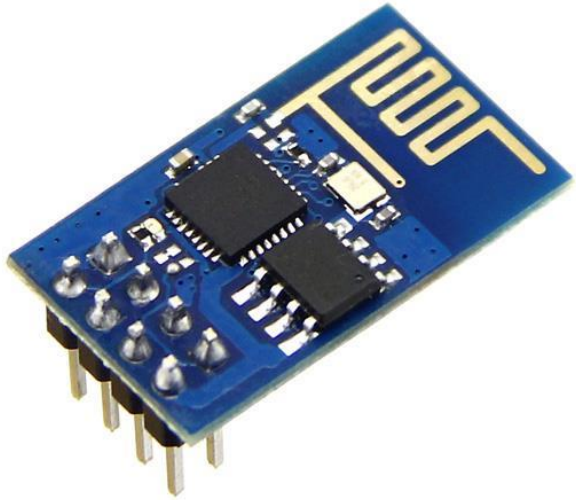

GND
C1
10pF
U1
GND
XIN
GND
XOUT
26MHz/10ppm
C2
10pF
R2 0R/NC
UTXD
URXD
VDD33
C3
10uF
C4
5.6pF
WIFI_ANT
L1
NC
L2
NC
TOUT
CH_PD
GPIO16
VDDA
LNA
VDD3P3
VDD3P3
VDD_RTC
TOUT
CHIP_EN
XPD_DCDC
EXT_RSTB
RES12K
VDDA
VDDD
XTAL_IN
XTAL_OUT
U0TXD
U0RXD
DVDD
SD_DATA_1
SD_DATA_0
SD_CLK
SD_CMD
SD_DATA_3
SD_DATA_2
VDDPST
R4 200
SD_D1
SD_D0
SD_CLK
SD_CMD
SD_D3
SD_D2
MTMS
MTDI
VDDPST
MTCK
MTDO
GPIO2
GPIO0
VDD
ESP8266EX
U2
GPIO14
GPIO12
GPIO13
GPIO15
GPIO2
GPIO0
U3
/CS
VCC
DIO
CLK
DO
/HOLD
/WP
SD_CMD
SD_CLK
SD_D2
SD_D1
SD_D0
SD_D3

Zapojení

- Pro boot do aplikace
  - MDTO -> Log0, GPIO0 -> Log1, GPIO2 -> Log1
  - CH_PD na Log1
- Pro bootloader
  - MDTO -> Log0, **GPIO0 -> Log0**, GPIO2 -> Log1
  - CH_PD na Log1

Demo

- Nechat kolovat…



ESP-01

ESP-07

ADAM

The easy way

ESP8266

Jak s modulem komunikovat

- Sériová linka
  - Bootloader má 76923 Bd
  - Aplikace typicky 9600 nebo 115200 Bd
- Ale co dál…
  - AT
  - Terminál (Lua, uPy, Frankenstein)
  - Programování (Arduino, C SDK)

AT příkazy

- Je tam od výroby
- Pokud máte svůj MCU

```
< AT+CWMODE=1                          -- Nastaví Clinet mode
> OK

< AT+CWLAP                             -- Seznam AP
> +CWLAP:(0,"WiFi",-91,"ac:a3:1e:d1:c7:e0",1)

< AT+CWJAP="WiFi","heslo"              -- Připojit k AP
> OK

…
```

Lua

- FW s interpretrem vyššího jazyka
- V současnosti nejpropracovanější
- Dva projekty:
  - Čistý interpret

    https://github.com/nodemcu/nodemcu-firmware
  - Podpora cloudu

    http://nodelua.org/
- IDE ESPlorer

  http://esp8266.ru/esplorer/

µPython

- **Silně** ve vývoji!
- Aktuálně v3.4.0
- Velký potenciál…

https://github.com/micropython/micropython/tree/master/esp8266

Frankenstein

- FW pro „*nix guys“
- Klasický terminál (ala busybox, cisco, XPort)
- Musí se kompilovat (`make menuconfig && make`)

https://github.com/nekromant/esp8266-frankenstein

Arduino IDE

- Kdo by neznal Arduino
- Port Arduino knihoven pod ESP8266 SDK
- Včetně podpory z Arduino IDE

http://www.arduinesp.com/

Demo

- Co chcete vidět?

VOJTA

Pro opravdové ~~muže~~ vývojáře

ESP8266 SDK

ESP8266 SDK - součásti

- Xtensa lx106 toolchain
- ESP8266 IoT SDK
- Open source componenty
  - lwIP
  - Contiki
  - axTLS
  - wpa_supplicant
  - net80211/ieee80211

https://github.com/pfalcon/esp-open-sdk

Co nabízí

- Vlákna
- Časovače
- Práci s WiFi
- SPI read, write
- TCP/UDP stack
- JSON
- GPIO, UART, PWM, I2C

Sekce paměti

- .text – Program v IRAM (max 32Kb)
- .irom0.text – Program ve SPI FLASH (ICACHE)
- .data – Data v DRAM (inicializované)
- .bss – Data v DRAM (neinicializované)
- .rodata – Read only Data v DRAM

ESP8266 SDK - sestavení

- Vyžaduje GNU/POSIX
- Pro ubuntu je nutné instalovat:

  make unrar autoconf automake libtool gcc g++ gperf flex bison texinfo gawk ncurses-dev libexpat-dev python sed

- Poté jen spustit kompilaci:

  make STANDALONE=y

Vytvoření binárek

- Program je potřeba rozdělit
- .text zapisujeme na adresu 0x00000
- .irom0.text podle Linker scriptu (většinou 0x40000)
- Můžeme použít nástroj **esptool**:

```
./esptool.py elf2image app.elf
```

Upload firmware

- Za pomoci UARTu
- Nutný bootloader mód
- Lze využít naříklad **esptool**

```
./esptool.py -p /dev/ttyS0 write_flash 0x00000 app-0x00000.bin 0x40000 app-0x40000.bin
```

```
/esptool.py --port /dev/ttzS0 write_flash 0x00..
Connecting...
Erasing flash...
Writing at 0x00009400... (100 %)
Erasing flash...
Writing at 0x00070800... (100 %)

Leaving...
```

Příklady kódu

- Hello world
- Použití časovače
- Použití vláken
- GPIO
- TCP server

Ukázka kódu

```
void ICACHE_FLASH_ATTR user_init() {

    uart_div_modify(0, UART_CLK_FREQ / 115200);
    os_printf("Hello world\r\n");

}
```

Časovač

```
LOCAL volatile os_timer_t some_timer;

LOCAL void ICACHE_FLASH_ATTR timerCallback(void *arg){
    // periodicky spouštěný kód
}

LOCAL void ICACHE_FLASH_ATTR init(){
    os_timer_disarm(&some_timer);
    os_timer_setfn(&some_timer, (os_timer_func_t *) timerCallback, NULL);
    os_timer_arm(&some_timer, 2500, 1);
}
```

Vlákna

```
#define MQTT_TASK_PRIO      0
#define MQTT_TASK_QUEUE_SIZE    1
LOCAL  os_event_t procTaskQueue[TASK_QUEUE_SIZE];

LOCAL void ICACHE_FLASH_ATTR ThreadTask(os_event_t *e){

    os_delay_us(1000);  // prostor pro váš kód

    system_os_post(MQTT_TASK_PRIO, 0, 0);
}
LOCAL void ICACHE_FLASH_ATTR user_init(){
    system_os_task(ThreadTask, TASK_PRIO, procTaskQueue, TASK_QUEUE_SIZE);
    system_os_post(MQTT_TASK_PRIO, 0, 0);
}
```

GPIO

```
LOCAL void ICACHE_FLASH_ATTR user_init(){
    PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U, FUNC_GPIO2); // set mapping to GPIO2

    GPIO_OUTPUT_SET(2, 1);  // set GPIO2 to OUTPUT and Log1
    os_delay_us(250000);
    GPIO_OUTPUT_SET(2, 0); // set GPIO2 to OUTPUT and Log0
    os_delay_us(250000);

    GPIO_DIS_OUTPUT(2); // GPIO2 to INPUT
    PIN_PULLUP_EN(PERIPHS_IO_MUX_GPIO2_U); // enable Pullup on GPIO2

    if(GPIO_INPUT_GET(2) == 1){ // is GPIO2 set?
        os_printf("up\r\n");
    }

}
```

TCP server

```
LOCAL void ICACHE_FLASH_ATTR webserver_listen(void *arg) {
        struct espconn *pesp_conn = arg;

        espconn_regist_recvcb(pesp_conn, webserver_recv); // callback to recv function
        espconn_regist_reconcb(pesp_conn, webserver_recon); // callback on reconnection start function
        espconn_regist_disconcb(pesp_conn, webserver_discon); // callback on disconnected function

}

LOCAL void ICACHE_FLASH_ATTR user_init(){
        LOCAL struct espconn esp_conn;
        LOCAL esp_tcp esptcp;

        esp_conn.type = ESPCONN_TCP;
        esp_conn.state = ESPCONN_NONE;
        esp_conn.proto.tcp = &esptcp;
        esp_conn.proto.tcp->local_port = 80;
        espconn_regist_connectcb(&esp_conn, webserver_listen);

        //#define SERVER_SSL_ENABLE
        #ifdef SERVER_SSL_ENABLE
                espconn_secure_accept(&esp_conn);
        #else
                espconn_accept(&esp_conn);
        #endif
}
```

Demo

- Ukázka SDK + MQTT klient

```
> mosquitto_sub -t /questions
```

Vice na:
http://macgyver.siliconhill.cz/wiki/projekty/esp8266

Adam Hořčica (@horcicaa)
Vojtěch Suk (@VojtechSuk)

http://macgyver.siliconhill.cz/kontakt.html