

Data do internetu

- základy tvorby webových stránek – HTML
- moduly pro WiFi komunikaci
- jak dostat data do internetu
- oživení obsahu s PHP – bezdrátová meteostanice
- moderní domácnost – RGB LED a ovládání osvětlení z webu

Webové stránky

- každodenní využití internetu je primárně založeno na otevírání zajímavých webových stránek ve vhodném prohlížeči (internet **browser**)
 - v mnoha případech zejména mobilních sociálních sítí či firemních řešení je to často nahrazeno vhodnou **aplikací** (Facebook, Youtube, Twitter, Instagram, ...) pro mobilní zařízení či instalací programu (client) v klasickém PC
- v nejjednodušším případě se ale jedná o stažení textového souboru s **HTML** obsahem (hypertext markup language) z webového serveru do lokálního počítače a jeho zobrazení
 - webový server může klidně běžet i na tom lokálním počítači, což je velmi výhodné při tvorbě webových stránek či vývoji webových aplikací
 - soubor webové stránky – třeba info.html – se dá zobrazit pomocí prohlížeče (**prezentace obsahu**), a nebo se dá otevřít pomocí nějakého textového editoru – **zobrazení kódu** generujícího obsah

Tvorba webové stránky

- **html** je prostý text – jde tedy vytvořit nebo upravit v nejběžnějším textovém editoru – Poznámkový blok (Win - Notepad)
- pohodlnější je nějaký programátorsky orientovaný editor, který dokáže barevně zdůraznit syntaktické prvky HTML kódu (Notepad++, tuzemský **PSPad**)

```
biochem_info.htm - Notepad
File Edit Format View Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>Biochemické informace
    </TITLE>
    <META content="text/html; charset=utf-8" http-equiv=Content-Type>
    <STYLE> BODY {font-family=arial; font-size=20; line-height=1.5}
H2 {font-weight=bold; color=darkred; font-size=30} H3 {font-
style=italics; font-weight=bold; font-size=26; color=orangered}
    </STYLE>
    <META name=GENERATOR content="MSHTML 8.00.6001.18876">
  </HEAD>
  <BODY style="tab-interval: 35.4pt" lang=CS link=blue vLink=purple>
    <IMG border=2 width="100%" src="OPVK_MU_rgb.tif"> <br> <br> <br>
    <DIV style="BORDER-BOTTOM: blue 4px solid; BORDER-LEFT: blue 4px
solid; PADDING-BOTTOM: 1px; BACKGROUND-COLOR: #ffCC00; PADDING-LEFT: 20px;
PADDING-RIGHT: 4px; BORDER-TOP: blue 4px solid; BORDER-RIGHT: blue 4px
solid; PADDING-TOP: 1px"><H1>BIOCHEMICKÉ INFORMACE
  <FONT size=2>(2013/III)
  </FONT></H1>
  <P><B>
  <FONT size=4>Petr Skládal (Ústav biochemie PŘF MU)
  </FONT></B>
  </P>
</DIV>&nbsp; <H2>Výukový materiál - informační zdroje pro
biochemii</H2>
  <P>
  <LI>primární zdroje - publikované články, patenty
  <BR>
  <LI>sekundární zdroje - databáze obsahů časopisů nebo vydaných
patentů (včetně klíčových slov a abstraktu)
  <BR>
  <LI>biochemická / biotechnologická databáze na internetu
```

```
PSPad - [H:\edu\sem_bakpr_1\bioinfo\biochem_info.htm]
File Projects Edit Search View Format Tools Scripts HTML Settings Window Help
1.. New1.txt 2.. biochem_info.htm
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>Biochemické informace
    </TITLE>
    <META content="text/html; charset=utf-8" http-equiv=Content-Type>
    <STYLE> BODY {font-family=arial; font-size=20; line-height=1.5} H
color=darkred; font-size=30} H3 {font-style=italics; font-weight=bold
orangered}
    </STYLE>
    <META name=GENERATOR content="MSHTML 8.00.6001.18876">
  </HEAD>
  <BODY style="tab-interval: 35.4pt" lang=CS link=blue vLink=purple>
    <IMG border=2 width="100%" src="OPVK_MU_rgb.tif"> <br> <br> <br>
    <DIV style="BORDER-BOTTOM: blue 4px solid; BORDER-LEFT: blue 4px soli
BACKGROUND-COLOR: #ffCC00; PADDING-LEFT: 20px; PADDING-RIGHT: 4px; BORDER-
BORDER-RIGHT: blue 4px solid; PADDING-TOP: 1px"><H1>BIOCHEMICKÉ INFORMACE
  <FONT size=2>(2013/III)
  </FONT></H1>
  <P><B>
  <FONT size=4>Petr Skládal (Ústav biochemie PŘF MU)
  </FONT></B>
  </P>
</DIV>&nbsp; <H2>Výukový materiál - informační zdroje pro biochemii</H
  </D>
```

HTML

<tagy> text uzavřený mezi </tagy>

- tagy určují, jak se text zobrazí v prohlížeči – forma prezentace, tagy jsou uzavřeny v <ostrých závorkách>

– co není v ostrých závorkách, je text, který se bude zobrazovat

- jednoduchá stránka:

- **co je co:**

<html> začíná a

</html> končí dokument

<head> a **</head>** začíná a končí hlavičku, která se sice nezobrazuje, ale obsahuje některé důležité údaje, například

<title> a **</title>** vymezují název dokumentu (klidně jiné než název souboru)

<body> a **</body>** co je mezi nimi, to se bude zobrazovat – jedná se o vlastní tělo (body) dokumentu

- výše uvedené tagy by měl obsahovat každý html soubor
- tento **1) způsob** je jistě nejnáročnější, ale máme vše pod kontrolou

```
<html>
  <head>
    <title>První webová stránka</title>
  </head>

  <body>
    Zde začíná text stránky...
  </body>
</html>
```

jednodušeji?

- pro tvorbu webu jsou samozřejmě dostupné
- 2) dedikované grafické **editační programy**, které dle požadovaného formátu samy vkládají potřebné tagy – uživatel je tedy v podstatě vůbec nemusí znát (**WYSIWYG**)
 - předpřipravené šablony, vše je velmi pohodlné, než se vyskytnou nějaké problémy, nebo je třeba vytvořit něco nestandardního
- 3) speciální **redakční systémy** (CMS, content management system) – WordPress, Drupal, Joomla – pouze se vkládá text a upravuje formát
 - stáhne se, nainstaluje na vlastní server, a může se začít plnit...
- 4) **webové služby** - wix, webnode, webgarden, estranky a Proweb
 - po registraci se pracuje přímo na webu poskytovatele, výsledkem je web přístupný přes adresu typu `www.vlastni_nazev.poskytovatel_sluzby.cz`
- ... u variant 3,4 je velmi složité přejít s hotovým webem na jinou platformu
- pro jednoduché záležitosti (tato přednáška, domácí projekty) lze doporučit variantu 1) případně v kombinaci se začleněním výstupů z varianty 2)

další „vymoženosti“

- reálné webové soubory (neco.html nebo neco.htm) neobsahují pouze html kod, ale i další elementy
- **definující styl** obsahu - `<style>` a `</style>` umístěné v rámci hlavičkové části, to ale může být i ve speciálním samostatném souboru `style.css` (kaskádové styly)
 - výhodné pro unifikaci stylu pro různé stránky v rámci webu a pro snadné změny proveditelné na jediném místě
- **javascript** – pro „oživení“ webového obsahu, resp. možnost reagovat na uživatelské akce (myš, klávesnice, ...)
- zapisuje se do html mezi tagy `<script>` a `</script>` v rámci těla dokumentu
 - je to vlastně interpretovaný programovací jazyk, běží v rámci prohlížeče na lokálním počítači
 - na prohlížeči to ale může uživatel zakázat (bezpečnostní riziko)
- populární je také tvorba webového obsahu pomocí **PHP**
 - skript („program“) běží na webovém serveru a z tam přítomných `skript.php` souborů se vytváří webový obsah, který pak posílá do prohlížeče uživateli

zde to ale nebudeme řešit...

- není naším cílem naučit se vytvářet webové stránky – to je nad rámec našich možností
- základní dovednosti lze získat **samostudiem**, velmi doporučuji

www.jakpsatweb.cz

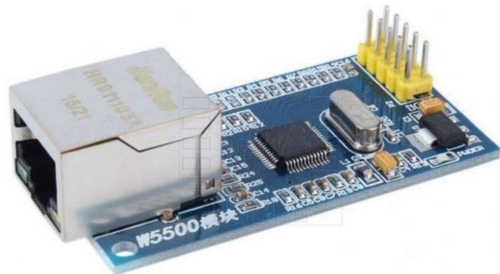
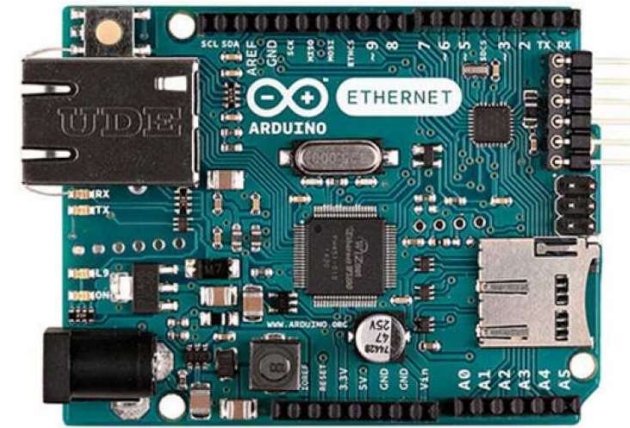
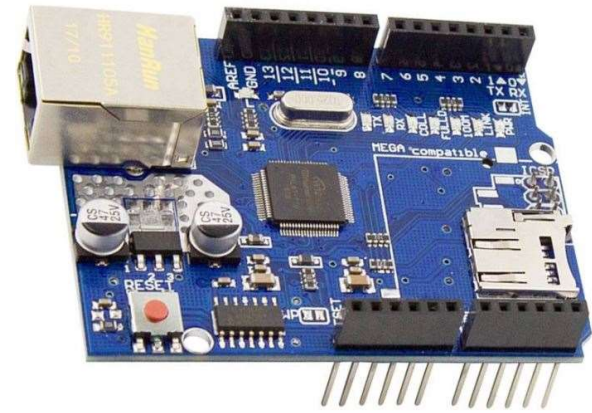
- naučí srozumitelnou formou základy tvorby webu pomocí html, css stylů a javascriptu ... existuje velmi dlouho, je průběžně aktualizováno
- další velmi názorný výukový web je

www.w3schools.com

- velmi příjemné jsou zde „try it yourself“ části, kdy se v levém okně zadává kód, a vpravo se hned ukáže, jak to bude vypadat v prohlížeči
- nadále pouze použijeme drobné fragmenty, které nám poslouží pro realizaci IOT potřeb a požadavků

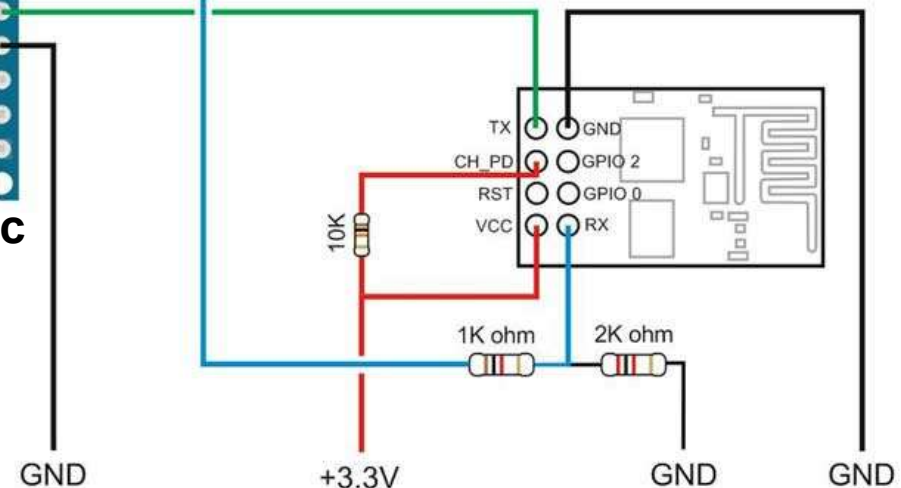
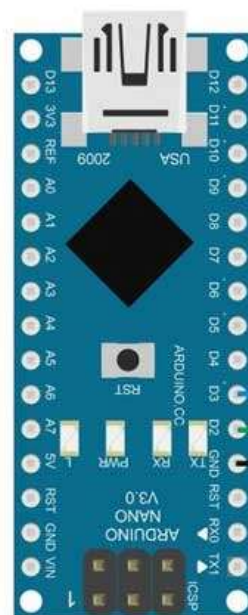
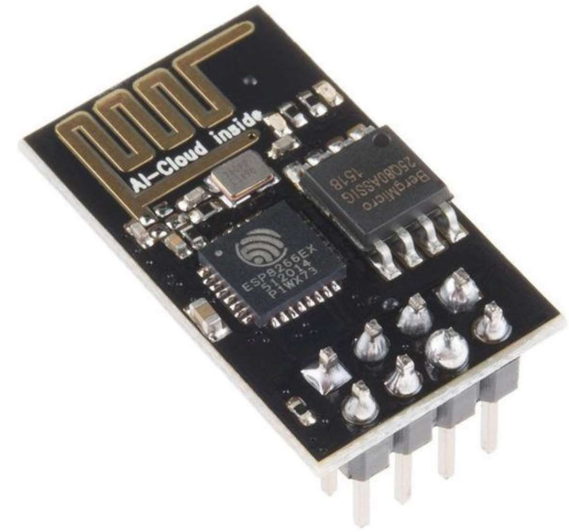
Z arduina do internetu?

- historicky šlo připojení do ethernetu provést přes „**shield**“ modul, který se k základní Uno desce shora připojil a poskytl klasický RJ-45 konektor a SD kartu (webové soubory)
 - základem je čip Wiznet 5100 - použití protokolů TCP i UDP, při programování je možné využít knihovnu Ethernet library
- následně se to spojilo do jediného modulu **Arduino Ethernet**, scházel mu USB konektor
 - programovalo se přes externí USB-seriový převodník
 - aktuálně sice převládají bezdrátové varianty WiFi komunikace, ale pevné připojení může být stále výhodné, třeba díky spolehlivosti a robustnosti, nemožnosti rušení
- lze pořídit ethernetové moduly na bázi čipu **Wiznet 5500** či ENC28J60, připojují se přes SPI rozhraní, nebo lze přes seriový port připojit XPort modul **Lantronix**



WiFi komunikace

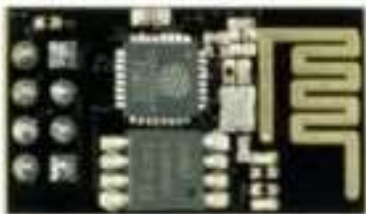
- revolučním krokem pro bezdrátové připojení do internetu (a nebo pouze do domácí WiFi sítě, dle potřeby...) se stal primárně komunikační modul **ESP8266** od Espressif varianta **ESP-01**:
- primárně se dá připojit k jinému mikrokontroleru (Arduino, Xiao, ...) pomocí sériové linky (i SoftwareSerial)
 - ESP jede na 3.3 V, ale mělo by „přežít“ i 5 V TTL úroveň (?)
 - ukázka **bezpečného** propojení:
 - pak se nastavení modulu, připojení do WiFi a posílání dat do internetu realizuje **prostřednictvím AT příkazů**
 - mimo Rx a Tx je ještě nutné propojit enable pin (aktivační) CH_PD s Vcc a samozřejmě také Vcc a GND
 - napětí 3.3 V vyvedené z Arduino modulu může být příliš „měkké“ pro nároky WiFi komunikace – stabilnější je získat „tvrdé“ 3.3 V z 5 V výstupu přes AMS1117 obvod



ESP-xx

- k dispozici je celá řada modulů lišících se typem antény, počtem a uspořádáním připojovacích pinů
- s prototypovací deskou se nejsnadněji připojí třeba ESP-05

ESP-01



ESP-02



ESP-03



ESP-04



ESP-05



ESP-06



ESP-07



ESP-08



ESP-09



ESP-10



ESP-11



ESP-12



ESP-13

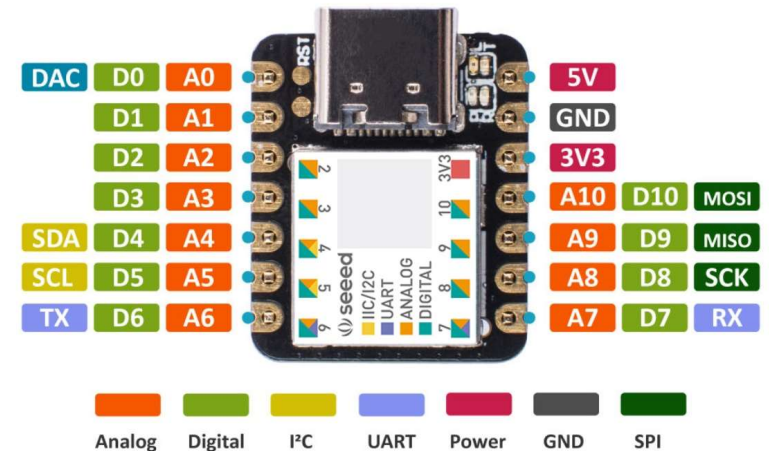
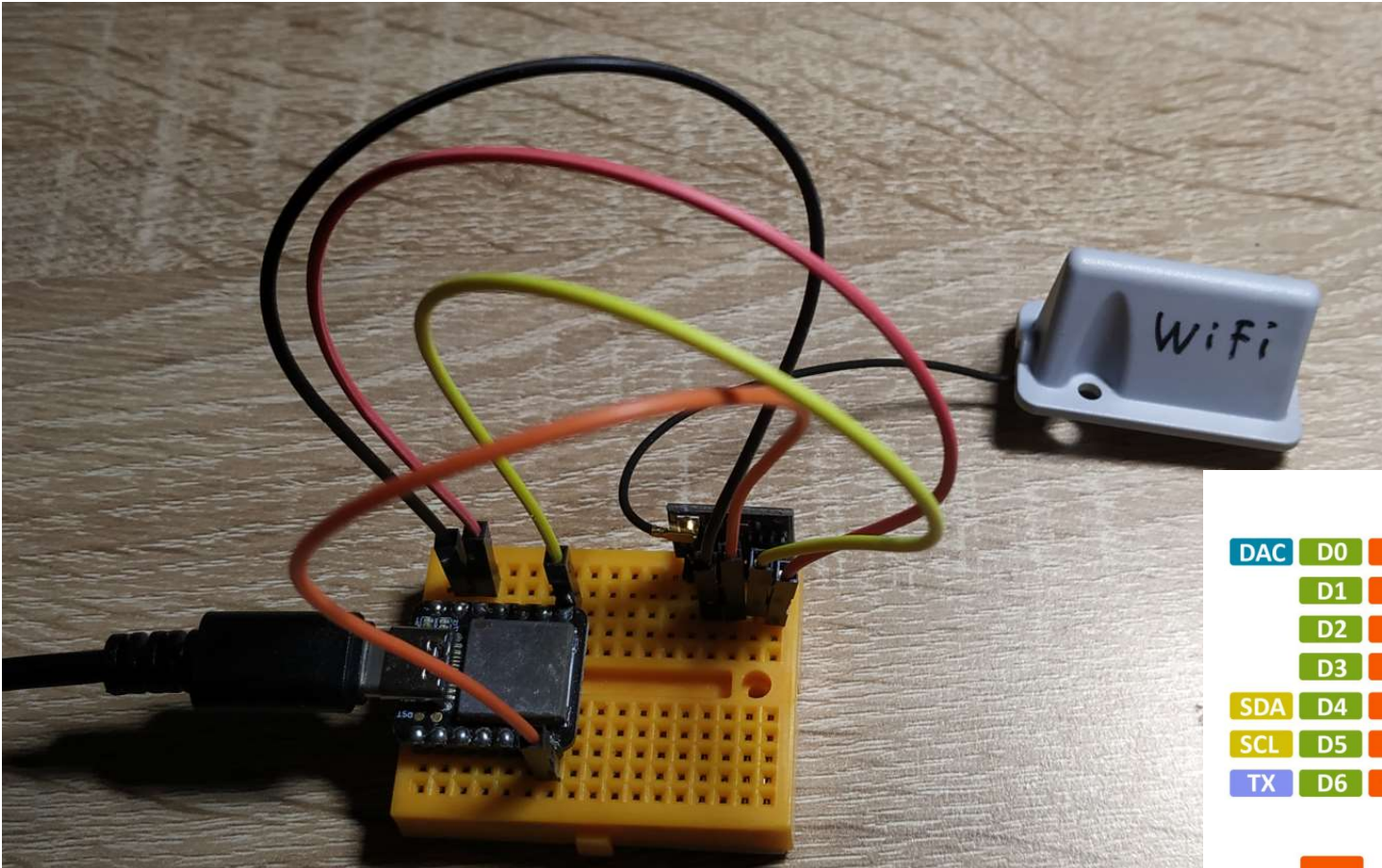
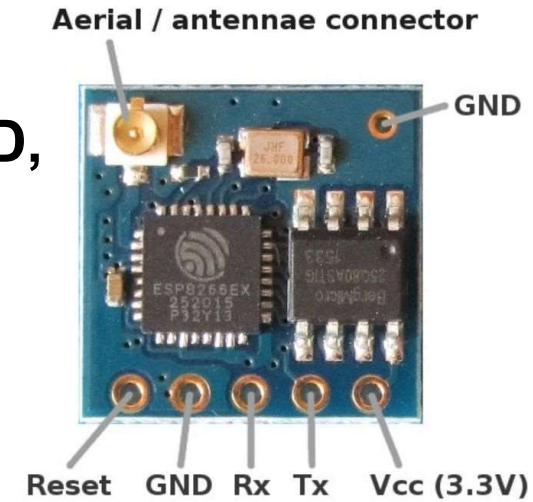


ESP-14



arduino - WiFi - internet

- kombinace Xiao – ESP požaduje propojení GND-GND, 3.3V-Vcc, Rx-Tx, Tx-Rx
- potřebujeme znát název WiFi sítě a heslo
- a webový server pro spuštění PHP skriptu, co bude přijímat data – stačí na počítači spustit balík WAMP či LAMP (Windows/Linux Apache MySQL PHP)



kod pro Xiao

```
String ssid = "aaaa"; String password = "xxxxxx";  
String data;  
String server = "147.251.99.99"; // your server ip  
String uri = "/_sys/esplog.php"; // location of the script  
String sadc;
```

```
void setup() {  
  Serial1.begin(115200); Serial.begin(115200);  
  reset(); connectWifi(); analogReadResolution(12);  
  Serial.println("Ready");  
}
```

```
void reset() {  
  Serial1.println("AT+RST"); delay(1000);  
  if(Serial1.find("OK") ) Serial.println("Module Reset");  
}
```

```
void connectWifi() {  
  String cmd = "AT+CWJAP=\"" + ssid + "\",\"" + password + "\"";  
  Serial1.println(cmd); delay(4000);  
}
```

```
void loop () {  
  sadc = String(analogRead(A1), DEC); Serial.println(sadc);  
  data = "value=" + sadc; httpPost(); delay(2000);  
}
```

```
void httpPost () {  
  Serial1.println("AT+CIPSTART=\"TCP\",\"" + server + "\",80"); //start a TCP connection.  
  if( Serial1.find("OK")) { Serial.println("TCP connection ready"); } delay(100);
```

```
String postRequest = "POST " + uri + " HTTP/1.0\r\n" + "Host: " + server + "\r\n" + "Accept: *" + "/" + "*\r\n" +  
  "Content-Length: " + data.length() + "\r\n" + "Content-Type: application/x-www-form-urlencoded\r\n" + "\r\n" + data;
```

```
String sendCmd = "AT+CIPSEND="; //determine the number of characters to be sent.
```

```
Serial1.print(sendCmd); Serial1.println(postRequest.length() ); delay(500);
```

```
if(Serial1.find(">")) {
```

```
  Serial.println("Sending.."); Serial1.print(postRequest);
```

```
  if( Serial1.find("SEND OK")) {
```

```
    Serial.println("Packet sent");
```

```
    while (Serial1.available()) { String tmpRSerial1 = Serial1.readString(); Serial.println(tmpRSerial1); }
```

```
    Serial1.println("AT+CIPCLOSE"); // close the connection
```

```
  }
```

```
}
```

```
} // thanks to arduino.stackexchange.com/questions/51806/sending-data-from-esp8266-to-php
```

- ssid a password vlastní
- na web serveru je v podadresáři `_sys` umístěn skript `esplog.php`
- průběžně se posílá hodnota z analogového vstupu A1
- ... http protokol není zrovna jednoduchý

přijímací PHP skript

- hodnota poslaná přes **POST** se doplní o časový údaj a uloží jako nový řádek v textovém souboru esplogs.txt:

```
<?php

$value = date("H:i:s")."\t".$_POST['value'];

$textfile = "esplogs.txt";
$fileLocation = "$textfile";
$fh = fopen($fileLocation, 'a') or die("Something went wrong!");
fwrite($fh, $value); fwrite($fh, "\r\n"); fclose($fh);

//echo "received: ".$value."<br>";

?>
```

- při nekvalitním WiFi připojení – větší vzdálenost – se data někdy nemusí podařit doručit
- PHP je velmi výkonný programovací jazyk pro tvorbu dynamických webů, běží na serveru a do prohlížeče přichází až hotová webová stránka (bezpečné...)

18:06:58	492
18:07:01	502
18:07:03	483
18:07:07	512
18:07:10	517
18:07:12	489
18:07:32	511
18:07:51	484
18:07:53	491
18:08:12	500
18:08:31	516
18:08:51	521

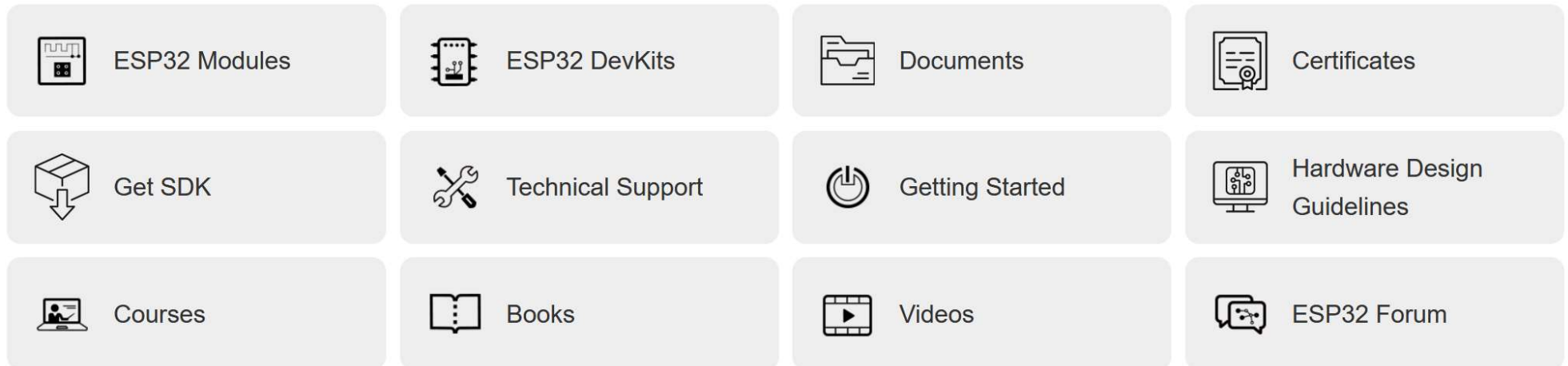
ESP jinak?

- nabízí se otázka - nešlo by program z arduina umístit rovnou do ESP?
- to je další ve skutečnosti velmi oblíbená a nyní již převládající alternativa, **ESP2866** a zejména výkonnější dvoujádrový **ESP32** mají dostatek prostoru i výkonu pro běh programu a komunikaci
- to byl ten správný základ, který vlastně „nastartoval“ nástup IOT
- ESP se dá programovat přímo v prostředí Arduino IDE po doplnění odpovídajícím modulem
 - jinou variantou je jazyk MicroPython – interpretované příkazy
 - (Arduino IDE samozřejmě generuje při kompilaci přímo mnohem rychlejší spustitelný kód ...)
- k dispozici je celá řada knihoven pro využití všemožných síťových protokolů, webová stránka jde vytvořit a spustit přímo z ESP modulu a zobrazit data z připojených sensorů

ESP32

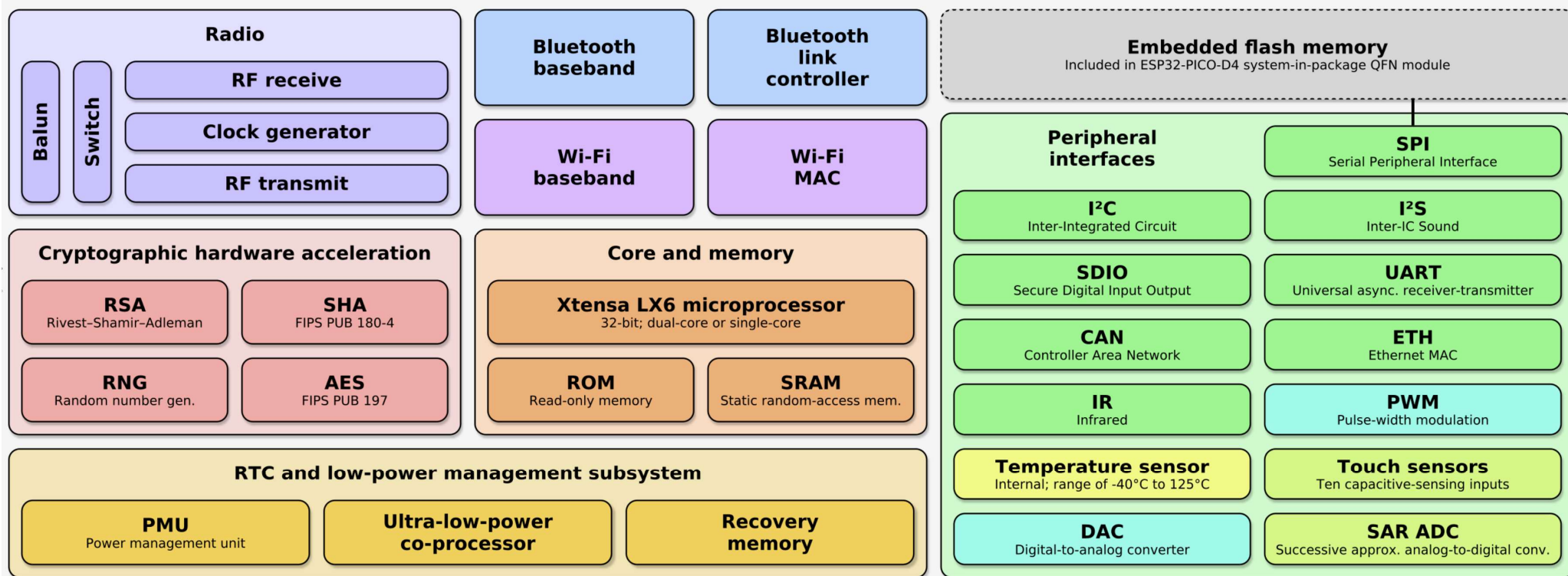
<https://www.espressif.com/en/products/socs/esp32>

- MCU bohatý na funkce s integrovaným Wi-Fi a Bluetooth
- konektivita pro široký rozsah aplikací



- **oficiální moduly:** ESP32-DevKitC ESP-WROVER-KIT ESP32-PICO-KIT ESP32-Ethernet-Kit ESP32-DevKit-S(-R) ESP32-PICO-KIT-1 ESP32-PICO-DevKitM-2 ESP32-DevKitM-1
- **mnoho** dalších **klonů**, často přidávajících další periferie (display, kamera, RF LoRa, ...)
- programování: espressif IDE, MicroPython, **Arduino IDE** (plugin)
- ideální základ pro IOT <http://esp32.net/>

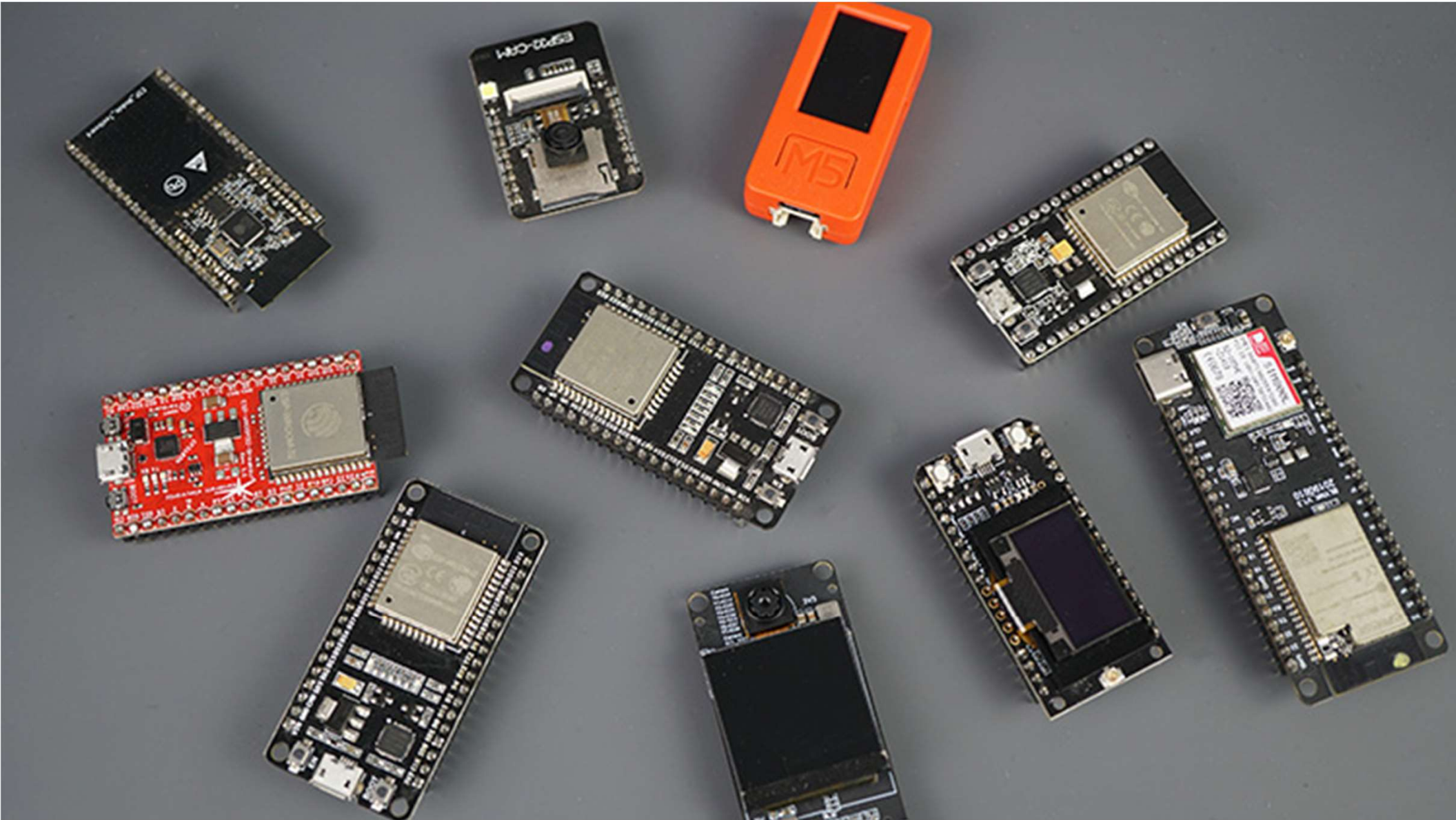
ESP32 funkční uspořádání – blokové schéma



- využívá buď mikroprocesor Tensilica Xtensa LX6 ve dvou- i jednojádrových variantách
- alternativně i 2-dvoujádrový Xtensa LX7 nebo 1-jádrový RISC-V
- zahrnuje vestavěné anténní spínače, výkonový zesilovač, nízkošumový přijímací zesilovač, filtry a správu napájení
- vyvinut Espressif Systems (Šanghaj), vyráběn TSMC 40 nm procesem

ESP32

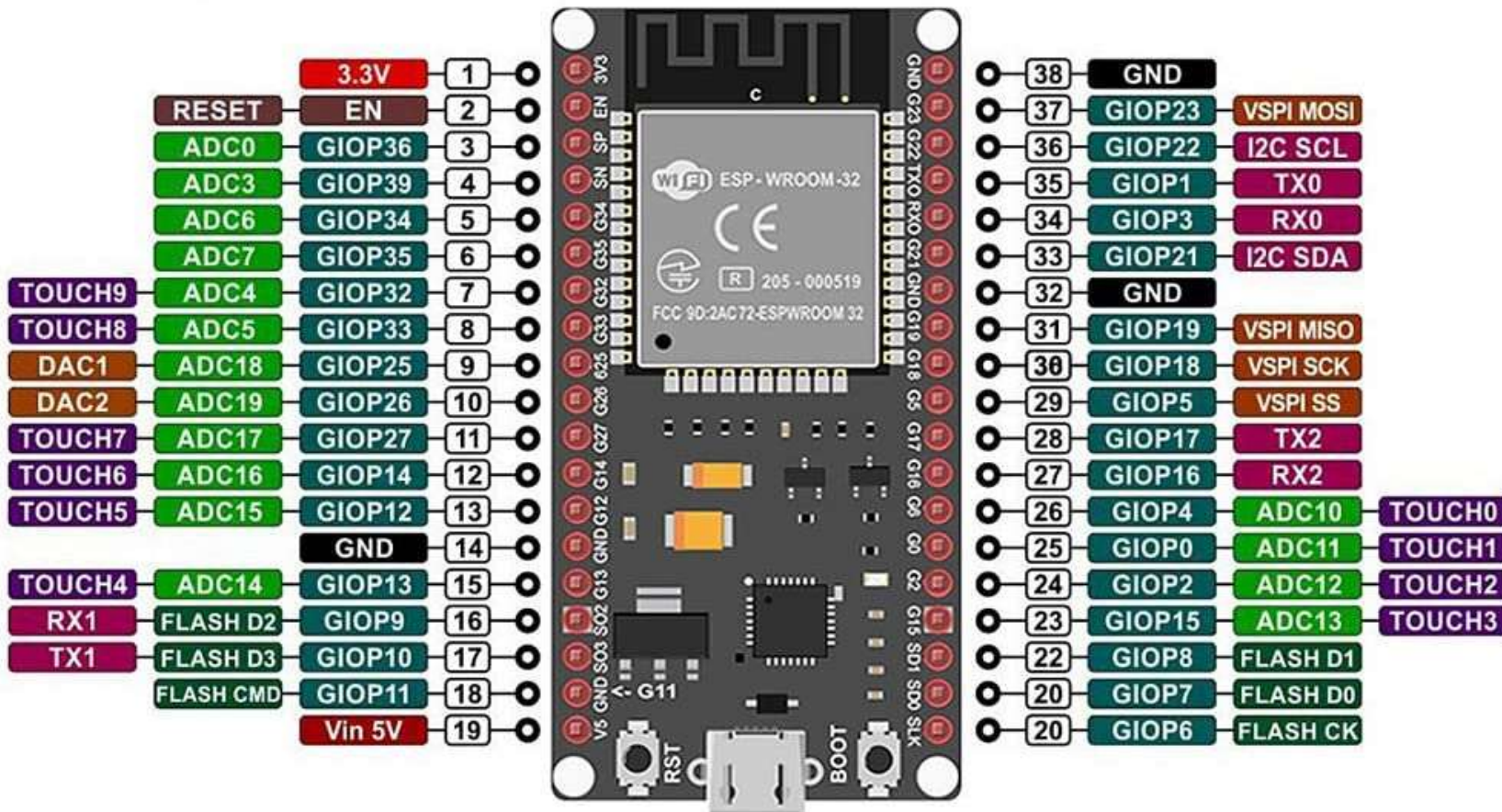
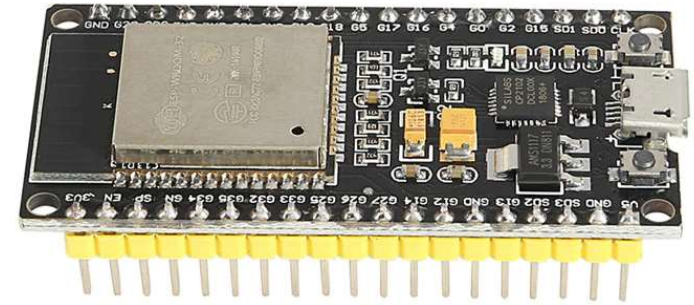
- mnoho variant <https://dratek.cz/205-espressif/>



Devkit C

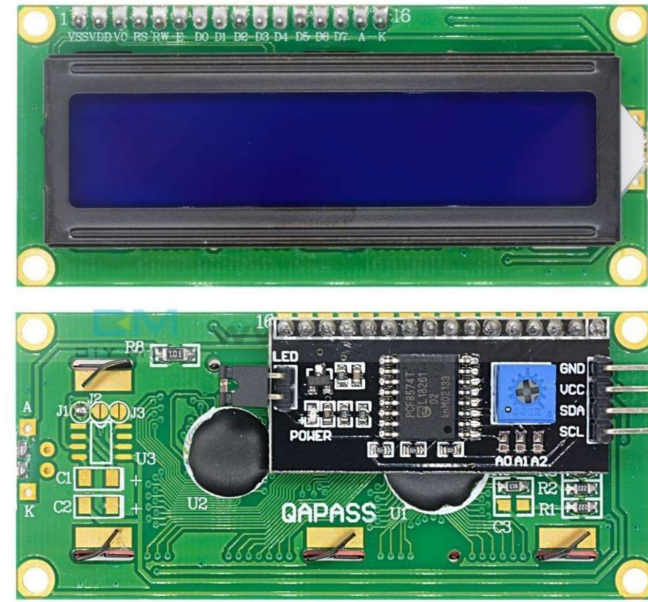
levné 227 Kč / 1,39 USD

- dvě CPU jádra se dají jednotlivě ovládat, frekvence nastavitelná od 80 po 240 MHz
- lze programově odpojit hlavní CPU a využít kopi spotřebou pro monitorování změn nebo překročení prahových hodnot u periférií – a pak aktivovat hlavní procesor
 - dlouhodobá funkce na baterii
- 38 pinů – řada periférií od kapacitních dotykových snímačů, Hallův snímač, rozhraní SD karet, Ethernet, SPI, UART, I2S (dig. audio) a I2C



LCD 16X2 znaků

- připojme si display - pohodlně s I2C modulem
- ať vidíme, co se bude dít ...
- GND-GND, Vcc-5 V, SDA-SDA, SCL-SCL



```
#include <LiquidCrystal_I2C.h>
```

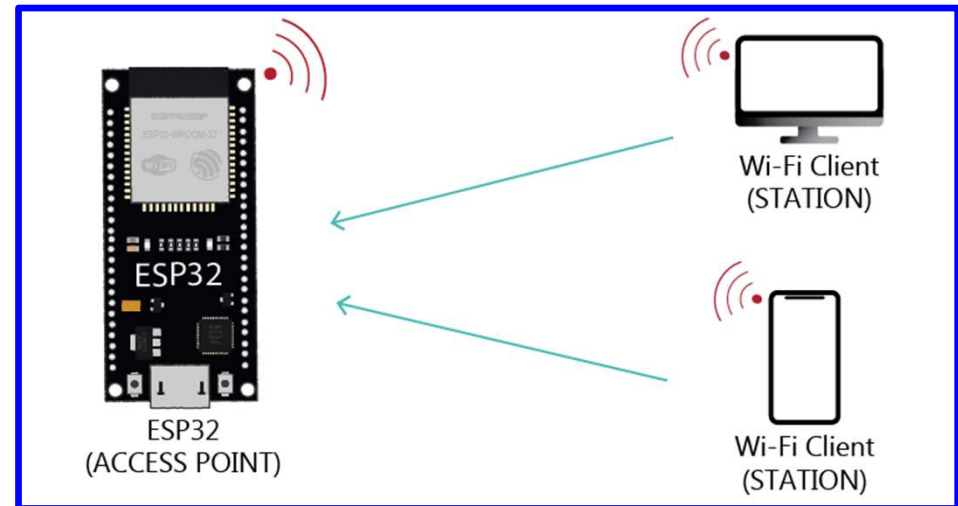
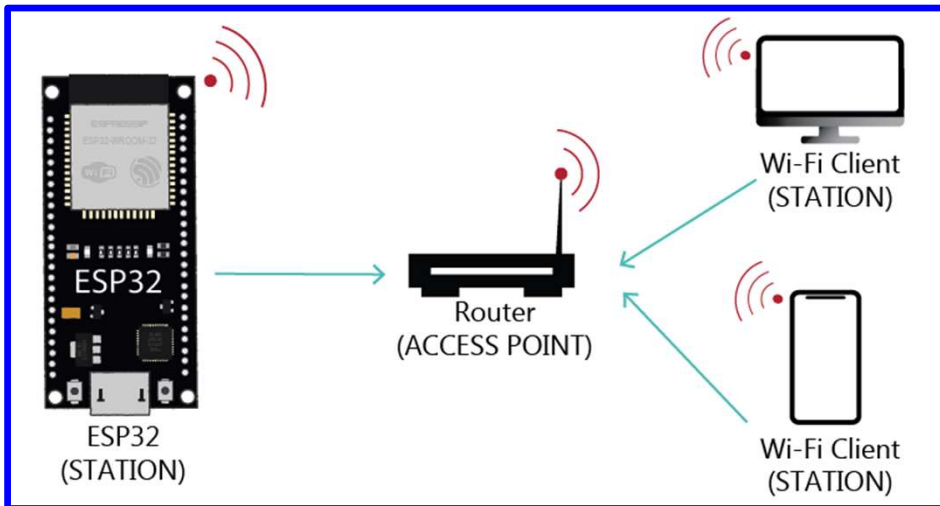
```
LiquidCrystal_I2C lcd (0x27,16,2); // create object, address and 16 cols x 2 rows
```

```
void setup () {  
  lcd.init (); // initialize the LCD  
  lcd.backlight (); // backlight on  
  lcd.print ( "Hello, world" ); // display text  
}
```

```
void loop () {  
  lcd.setCursor (0, 1); // cursor on first pos col=0 of the second line row=1  
  lcd.print ( millis () / 1000); lcd.print ( " s" ); // print something  
  delay (200);  
}
```

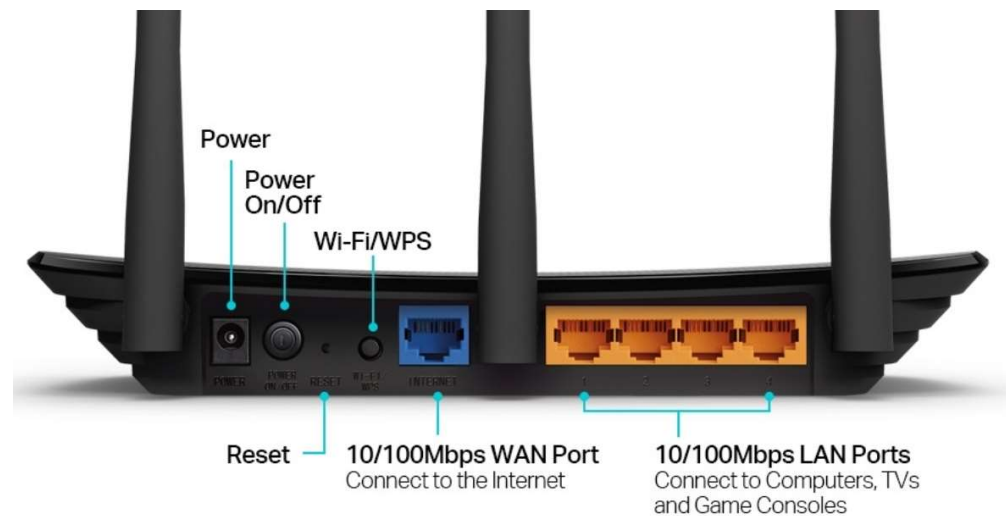
ESP32 – WiFi mody

- **station** - při práci v tomto režimu se ESP32 chová jako zařízení s podporou WiFi připojené ke stávající existující aktivní síti WiFi
- **access point** - při práci v režimu Soft AP se ESP32 chová jako hostitel / poskytovatel WiFi sítě, k ní se mohou připojit další zařízení
 - takto lze např. sdílet mobilní připojení přes vytvoření malé wifi sítě
- třetí možností je použití obou módů současně, resp. smíšeně
 - třeba <https://techtutorialsx.com/2021/01/04/esp32-soft-ap-and-station-modes/>



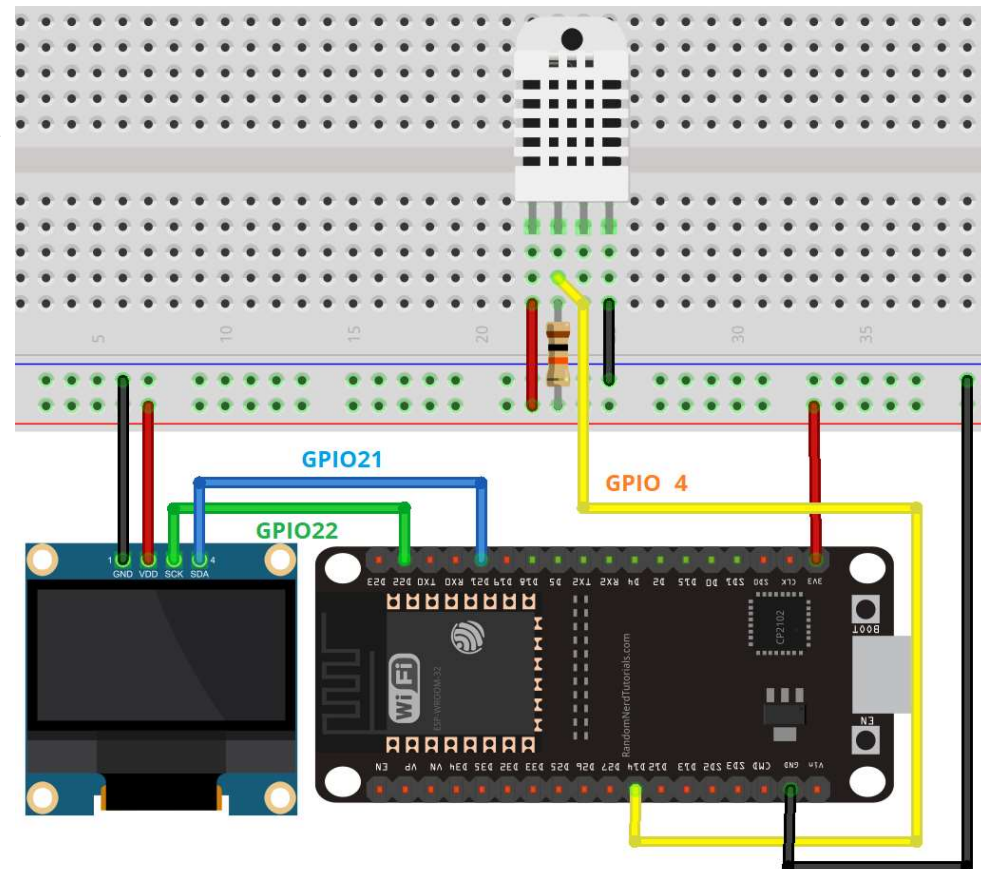
Router

- **WiFi síť** typicky v domácím prostředí realizuje **WiFi router**
- může být součástí nějakého přístupového zařízení, které nám nainstaluje poskytovatel internetového připojení (optický kabel, po drátě, bezdrátově – externí WiFi nebo mobilní, skylink, ...)
 - k připojení potřebujeme znát název sítě a heslo (ssid, password)
- konektivita pro široký rozsah aplikací, LAN porty pro vnitřní síťový segment
- široké možnosti nastavování, někdy i VPN (virtual private network, bezpečné připojení do domácích / pracovní sítě)



ESP32 jako AP

- takto vznikne „smart“ zařízení, které nabídne své služby do okolí prostřednictvím vytvořené WiFi sítě, nejde tedy o přenos do internetu jako takový
- typicky přístup na vnitřní webovou stránku, která zobrazuje signály z připojených sensorů (moderní instrumentace), nebo nabídne ovládání připojených periférií (barevné žárovky, inteligentní zásuvky, ...)
- **příklad** – zobrazení údajů ze sensoru DHT22 – teplota a vlhkost
- navíc přidán OLED display – potřebujeme zjistit IP adresu modulu, když nemáme připojení přes USB do počítače



1 nastavování

```
#include <WiFi.h>
#include <WebServer.h>
#include <U8x8lib.h>
#include "DHT.h"

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

const char* ssid = "ESP32AP"; const char* password = "1234";

WebServer server(80);

uint8_t DHTPin = 4;
DHT dht(DHTPin, DHTTYPE);
U8X8_SH1106_128X32_VISIONOX_HW_I2C u8x8(/* reset=*/ U8X8_PIN_NONE); // ok for 0.91" OLED 128x32

float t,rh;
char buf[20];

void setup() {
  Serial.begin(115200); delay(100);
  pinMode(DHTPin, INPUT); dht.begin();
  u8x8.begin(); u8x8.setPowerSave(0); u8x8.setFont(u8x8_font_amstrad_cpc_extended_r);
  WiFi.mode(WIFI_AP); WiFi.softAP(ssid, password);
  sprintf(buf,"%s",WiFi.softAPIP().toString().c_str());
  Serial.print("WiFi AP IP: "); Serial.println(buf); u8x8.drawString(1,2,buf);

  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);

  server.begin();
  Serial.println("HTTP server started");
}
```

2 reakce na přístupy

```
void loop() { server.handleClient(); }
```

```
void handle_OnConnect() {  
  t = dht.readTemperature(); rh = dht.readHumidity();  
  server.send(200, "text/html", SendHTML(t,rh));  
}
```

```
void handle_NotFound(){ server.send(404, "text/plain", "Not found"); }
```

```
String SendHTML(float Temperaturestat,float Humiditystat){  
  String ptr = "<!DOCTYPE html> <html>\n";  
  ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-  
scalable=no\">\n";  
  ptr += "<title>ESP32 Weather</title>\n";  
  ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align:  
center;}\n";  
  ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;}\n";  
  ptr += "p {font-size: 24px;color: #444444;margin-bottom: 10px;}\n";  
  ptr += "</style>\n";  
  ptr += "</head>\n <body>\n";  
  ptr += "<div id=\"webpage\">\n";  
  ptr += "<h1>ESP32 Weather</h1>\n";  
  ptr += "<p>Temperature: ";  
  ptr += (float)Temperaturestat;  
  ptr += " oC</p>";  
  ptr += "<p>Humidity: ";  
  ptr += (float)Humiditystat;  
  ptr += "%</p>";  
  ptr += "</div>\n </body>\n </html>\n";  
  return ptr;  
}
```


Příklad – Nanophotometer

- The NanoPhotometer is equipped with WiFi, which can be used as a WiFi network or as a WiFi hotspot
 - WiFi network allows data transfer and direct printing via AirPrint® / IPP compatible printers supporting PDF format
 - Note: IPP version 2.2 is required and some printer configuration settings might need to be changed in order to allow communication with the NanoPhotometer
- The WiFi Hotspot provides the option to control the NanoPhotometer by other WiFi devices as Windows computer, smartphones or tablets
- WiFi Hotspot connection details:
 - SSID: NanoPhotometer serial number
 - password: Implenuser



ESP32 smíšený mod

```
#include <WiFi.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd (0x27,16,2);
const char* wifi_network_ssid = „wrouter“;
const char* wifi_network_password = „hesloxxx“;
const char* soft_ap_ssid = "ESP32_AP";
const char* soft_ap_password = "hesloxxx";

void OnWiFiEvent(WiFiEvent_t event) {
  switch (event) {
    case SYSTEM_EVENT_STA_CONNECTED: Serial.println("ESP32 Connected to WiFi Network"); break;
    case SYSTEM_EVENT_AP_START: Serial.println("ESP32 soft AP started"); break;
    case SYSTEM_EVENT_AP_STACONNECTED: Serial.println("Station connected to ESP32 soft AP"); break;
    case SYSTEM_EVENT_AP_STADISCONNECTED:
      Serial.println("Station disconnected from ESP32 soft AP"); break;
    default: break;
  }
}

void setup() {
  Serial.begin(115200); lcd.init (); lcd.backlight ();
  WiFi.onEvent(OnWiFiEvent);
  WiFi.mode(WIFI_MODE_APSTA); WiFi.softAP(soft_ap_ssid, soft_ap_password);
  WiFi.begin(wifi_network_ssid, wifi_network_password);
  while (WiFi.status() != WL_CONNECTED) { delay(500); Serial.println("Connecting to WiFi.."); }
  Serial.print("ESP32 IP as soft AP: ");
  Serial.println(WiFi.softAPIP()); lcd.setCursor(0,0); lcd.print(WiFi.softAPIP());
  Serial.print("ESP32 IP on the WiFi network: ");
  Serial.println(WiFi.localIP()); lcd.setCursor(0,1); lcd.print(WiFi.localIP());
}

void loop() {}
```

ESP32 LED ovládání

- <https://lastminuteengineers.com/creating-esp32-web-server-arduino-ide/>
- přes webovou stránku

Další informace

- **Santos R. a Santos S. Learn ESP32 with Arduino IDE. 2020, 640 stran.**
 - **Stručný úvod do elektroniky, základních součástek a prvních triviálních experimentů, včetně různých mikrokontrolerů.**