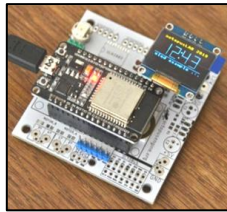


### 4. díl OctopusLAB

#### Nastavení ESP32 a první pokusy s Micropythonem

Modul ESP32 (verze 2 x 15 pinů) zapojený v ROBOT-board – vývojová deska umožňující velmi jednoduchým způsobem připojit celou řadu periférií – od senzorů, přes displeje (na obrázku barevný 128x64 OLED) až po akční prvky (motory, serva a podobně) ▶



Původní logo Micropythonu naznačovalo, že se jedná

◀ o systém použitelný v mikrokontrolérech.

Podle návodu, který jsme uvedli v minulém díle a terý je podrobně popsán na stránkách:

<https://www.octopuslab.cz/micropython-octopus/>

připojíme modul pomocí kabelu k počítači a přes terminál můžeme začít s prvním experimentováním. Přes terminál **putty.exe** (připojený na COM port jako „serial“ přenosovou rychlostí 115200) by s námi modul ESP už měl komunikovat.

Ještě si zmíníme dvě základní klávesové zkratky:

CTRL+C vede k přerušení běhu programu

CTRL+D provede "soft" restart zařízení

(pro hard restart použijte tlačítko „EN“ na modulu)

Modul se vám přihlásí přednastaveným výpisem procesů, které se provádějí po restartu, pak systém vypíše, která verze Micropythonu je aktuální (měla by to být Micropython v1.11-495 a vyšší) - a pak se objeví takzvaný „prompt“, tři šipky „>>>“, což je terminálová **výzva**, abychom tam něco napsali: **příkaz** nebo „*posloupnost příkazů*“ - zkuste napsat například: „a=123“ a konec řádku pokaždé potvrdit ENTER:

```
>>> a = 123 # do proměnné se uložila hodnota (číslo 123)
>>> a # vytiskne / zobrazí hodnotu proměnné nebo napište
print(a) # pokud to chcete použít v programu.
```

```
123
```

\*) pouze v této úvodní části jsme naznačili, co píše systém (tučně) a co píšete vy (normální) - poznámky pak píšeme *šikmo* (občas za znak #, který se používá v Pythonu)

```
>>> a + 10
```

```
133
```

*zobrazí vypočtenou hodnotu (jako kalkulačka)*

Někdy chceme složitější matematické výrazy, než je + sčítání, - odčítání, \* násobení, / dělení

```
>>> import math
```

importujeme knihovnu, až když jí potřebujeme, jinak nám zbytečně blokuje operační paměť

```
>>> math.log10(1000)
```

```
3.0
```

```
>>> math.pi
```

3.141593 *počet desetinných míst je v Micropythonu oproti Pythonu více omezený*

```
>>> šipky nahoru nebo dolů -> historie příkazů
```

```
>>> metoda + tečka > TAB doplnění / nápověda
```

*Micropython má obrovskou výhodu v tom, že běží jako interpret (není kompilovaný - přeložený do strojového kódu)*

A tak když uživatel napíše název proměnné, třídy nebo instance objektu, skoro všechno se o nich můžeme okamžitě a průběžně dozvědět. Právě proto máme k dispozici všechny metody, po „tečka TAB“.

Klávesa TAB (tabelátor) tedy slouží i jako „nápověda“ nebo pro efektivní našeptávač pro dokončování příkazů, což s jistotou praxí může znatelně urychlit práci s příkazovými řádky (podobně jako v Linuxu)

První verze knihovny octopus() spatřila světlo světa již začátkem roku 2018. Od té doby prošla několika zásadními proměnami. Nyní se soustředíme na externí knihovny a spolupracující moduly.

S rozšířením o octopus() nám po správném zapojení a inicializaci na velkou část práce s perifériemi postačí „jednořádkové“ příkazy:

```
led3.value(1) # rozsvítí LED diodu 3
ws.color(BLUE) # barevná RGB svítí modře
piezzo.beep() # bzučák jednou pípne
```

```
servo.set_degree(90) # natočí servo na 90 stupňů
display.show(123) # na displeji zobrazí číslo „123“
val = photo.get_adc_aver(20) # A/D fotoodpor průměr 20ti
načtených hodnot
i2c = i2c_init() # inicializace sběrnice i2c
i2c.scan() # najde všechna připojená zařízení na
sběrnici
wifi.connect() # připojí se k WiFi
web_server() # spustí webový server
```

#### Práce s připojenou LEDkou

from util.led import Led # importování spec. knihovny pro LED  
# Micropython natahuje knihovnu do paměti a takto máme paměť pod naší kontrolou, přesune tam pouze to, co potřebujeme.

```
led = Led(2) # Pin 2 určuje vestavěnou LED na modulu ESP
led.value(1) # metoda value() mění hodnotu na 1 > ledka
svítí
led.value(0) # ledka zhasne
led.blink() # metoda blink() - ledka jednou blikne
led.toggle() # změna hodnoty..
```

```
led.TAB # zkuste si po tečce zmáčknout TAB (klávesu
tabelátor)
```

# Led s velkým L je objekt a příkazem led3 = Led(pin3) vytváříme jeho „instanci“. O objektovém programování si povíme více v dalších dílech.

Zakladem je, že objekty mají nějaké definované **metody** - podprogramy, funkce - co se s daným objektem dá provádět: value(), blink(), toggle()...

#### Od pípání k jednoduché melodii

```
from util.buzzer import Buzzer
piezzo = Buzzer(33) # Pin 33 pro bzučák, může být i jiný
piezzo.beep() # jednoduché krátké pípnutí
piezzo.TAB # po tečce TAB opět vypíše dostupné
metody
```

# chceme-li zahrát jednoduché melodie, potřebujeme noty:  
from util.buzzer.notes import \*  
# převede názvy not na frekvenci: např. A = 440Hz („komorní a“)

```
# celá sekvence pro hraní:
from util.octopus import piezzo
from util.buzzer.melody import jingle1 # importuje jednu
melodii
piezzo.play_melody(jingle1) # a tu pak zahraje
```

# délka not je definována dělitelem ve zlomku:  
2 > 1/2, 4 > 1/4, 8 > 1/8  
# 8 je tedy pro notu osminovou

```
# vlastní melodie, jako pole dvojic [nota a její délka]
jingle2 = [[C6,4], [0,4], [C5,8], [0,8], [C6,8]]
piezzo.play_melody(jingle1)
```

# pro hraní melodie se dá nastavit globální rychlost  
# 60 BMP (beats pro minute) > celá nota trvá 1 sec (60 sec / min)

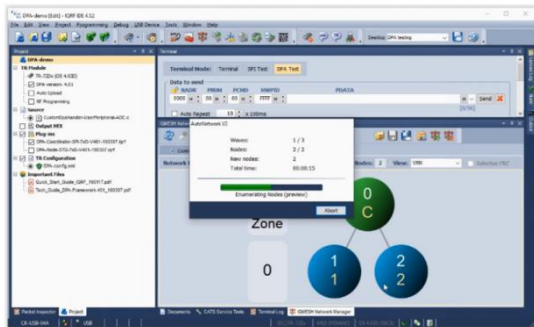
```
piezzo.speed = 120 # vede ke dvojnásobnému zrychlení
piezzo.play_melody(alert1)
piezzo.speed = 30 # vede ke zpomalení na polovinu
```

# oproti metodám, které jsou vždy se závorkami, mají objekty také vlastnosti (properties):

```
piezzo.speed = 60 # nastavení vlastnosti
piezo.piezzo.play_melody(song) # spuštění metody
```

Máme připraveno mnoho dalších podobných příkazů, ze kterých se dá pochopit funkce HW a následně jejich propojováním se realizují první velmi jednoduché programy. **Například program digitálních hodin nebo teploměru má pak pouze několik řádků.** S dalšími konkrétními ukázkami budeme pokračovat příště.

Milí čtenáři, těším se s vámi opět nashledanou v HK 142.  
Jan Čopák, [www.octopuslab.cz](http://www.octopuslab.cz)



### AutoNetwork – třetí ze tří způsobů, jak vytvoříte síť IQRF

Touto metodou můžete automaticky připojit do sítě všechny připravené nody, které mají stejné Access Password jako koordinátor a jsou (nebo budou) v dosahu koordinátoru nebo nově vytvářené sítě. Proces přidávání nodů do sítě a zjištění jejich umístění v síti (Discovery) probíhá automaticky. Postupně, v jednotlivých vlnách, jsou nalézány nové nody, těm je přidělena adresa a provedeno zjištění topologie – Discovery. V dalších vlnách jsou tyto nody využity pro další hledání nových nodů. Proces skončí ve chvíli, kdy nedojde k nalezení žádného dalšího nodu v několika vlnách po sobě (počet je možno definovat před zahájením AutoNetwork), případně jsou splněny další podmínky ukončení procesu. Podívejte se na video o vytváření IoT sítě s IQRF:

<https://youtu.be/qRwnxiovps?t=794>

Ivona Spurná, IQRF Smart School Manager, [ivona.spurna@iqrf.org](mailto:ivona.spurna@iqrf.org), [www.iqrfalliance.org](http://www.iqrfalliance.org)



**Japonští radioamatéři** popisují svoji návštěvu u Oldřicha Burgera, OK2ER v jeho firmě **B plus TV**, kde obdivovali vývoj a výrobu jeho skvělých smyčkových antén. Japonský radioamatérský časopis CQ návštěvě v Klimkovicích věnoval celé čtyři stránky! ▶

**Historika s Audionem** Je to ještě běh na delší dobu - audion mám také na stole, chybí mi jen pár spojů, ale když já mám teď období rádiové chcíplé myši:) Budou svátky a tak snad zase pokročím. Když jsem si vymyslel, že jej překopu na moderní ladičák a elku se zapojenou mřížkou G2 s +napětím (a G3 v soklu spojenou s katodou), na malém prkénku, tak mě nenapadlo, že poslední dva spoje budu dělat každý cca hodinu:)

Minulý víkend jsem milý Audion držel v ruce, koukám na něj a cítím, jak se páli bakelit (fakt typický pach pálicího se bakelitu). Tak si říkám, zapojený není, nic se na něm netaví a nemůže - tak už definitivně blbnu, jsem zralý do blázince. No, žena připálila na plynovém sporáku bakelitová ouška od kastrólu:) Robert Basl

**V dobách minulých** se většina oslav neobešla bez lampionového průvodu dětí. Mládež neznalá vložila do lampionu svíčku, což považovala za dostačující. Trochu technicky založení mládežníci zakoupili plochou baterii, žárovku a sklízeli obdiv za svoji šikovnost. Mně se to zdálo málo - přece moje dcera se nespokojí s takovýmto řešením. Nainstaloval jsem do lampionu blikáč a vyšli jsme mezi dav. Blikáč blikal a obdiv se stupňoval, Jen do té doby, kdy mě zastavila paní od vedle: „sousedo, vaše dcera má hezký lampion, ale nějak jí to bliká. To jí to neumíte spravit?“. Jaroslav Winkler, OK1AOU, [intep@centrum.cz](mailto:intep@centrum.cz)



**... nabízet dětem tvůrčí činnost** v jakémkoliv oboru, v našem případě v elektronice, která je nám nejbližší. Probouzet u dětí zvědavost, vést je ke zkoumání jak věci fungují, vést je k silnému pocitu, že svými silami mohou měnit svět směrem k lepšímu...

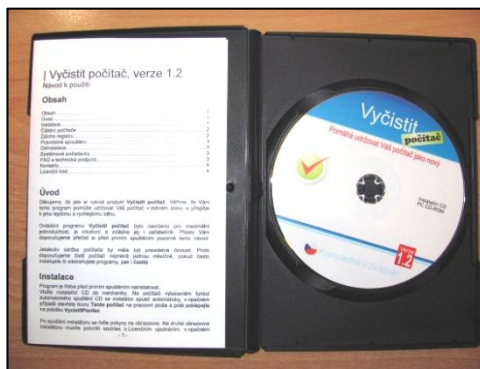
Petr Fišer, OK1XGL, [ok1xgl@seznam.cz](mailto:ok1xgl@seznam.cz)

**Výsledky Minitestů z HK 139** Označme  $n$  počet viditelných schodů, když schodiště stojí. Za jednotku času sejdu dolů u jeden schod. Jdu-li dolů po jedoucím schodišti, na 50 kroků ujde schodiště  $n-50$  schodů v 50 jednotkách času. Nahoru ujele 125- $n$  schodů za 125:5=25 jednotek času. Z rovnice, kterou tím dostáváme,  $(n-60):50=(125-n):25$  je hledaný počet schodů  $n=100$ .

Z juniorů jako první a jediný správně odpověděl Pavel Horský, získal 16 bodů a vyhrál **stavebnici Elektronické uspávkno**. Pavel píše o sobě: Chodím na Gymnázium Brno-Řečkovice. O Hamíkově Koutku jsem se dozvěděl na letošní akci Cihelna v Králíkách. Letos jsem se zúčastnil celostátního kola v radioelektronice a získal jsem 4. místo v kategorii do 16 let. Ze stavebnic jsem poskládal osciloskop, funkční generátor a navrhl jsem teplotní čidlo se záznamem a gsm modulem. **Pavel dostane ke stavebnici EI. uspávkno ještě balíček elektronických součástek.**

Z dospěláků mají po 16 bodech Zdeněk Hladík OK7DR, Tomáš Petřík OK2VWE, Ladislav Valenta OK1DIX, Vladimír Štemberg, Jiří Němejc OK1CJN. Bylo to tentokrát trochu obtížnější.

**Náš Minitestík** Co to v radiotechnice byl či je éter? Obtížnost: 8 bodů. Naši junioři soutěží o **CD Vyčistit počítač a balíček součástek** ▶



**Zdibec moudra na závěr**

**Věnuj osmdesát procent svého času tomu, že se soustředíš na příležitosti zítřka místo problémů včerejška.**

Brian Tracy

**HAM** je mezinárodně používaný pojem pro radioamatéra  
**HAMÍK** je tedy mladý, začínající, budoucí radioamatér

Toto číslo vyšlo 30. listopadu 2019  
Vychází každou sobotu v 08:00 h

**HAMÍKOV KOUTEK** je přílohou Bulletinu Českého radioklubu,

je určen pro vedoucí a členy elektro - radio - robo kroužků, jejich učitele, rodinné kluby, rodiče, prarodiče a všechny příznivce práce s mládeží; vzniká ve spolupráci s ČRK, ČAV a OK QRP klubem

Všechna předchozí čísla HK, adresy kroužků, stavební návody a mnoho dalšího najdete na <http://www.hamik.cz/>  
© Petr Prause, OK1DPX, redakce HAMÍK, Čechovská 59, 261 01 Příbram, tel. 728 861 496, [dpx@seznam.cz](mailto:dpx@seznam.cz)