

# LABORATÓRNY ZDROJ LK-20

Ing. Ľubomír Karlík

Medzi základné prístroje v dielni každého rádioamatéra, alebo kutila zaoberajúceho sa konštrukciou a opravami elektronických prístrojov, nepochybne patrí napájací zdroj. Na začiatku to najčastejšie býva zdroj s pevným napätím +5 V pre napájanie číslicových obvodov,  $\pm 15$  V, alebo  $\pm 12$  V pre napájanie obvodov s operačnými zosilňovačmi, s celkovým výkonom do 50 W, neskôr regulovateľný zdroj do 30 V s výkonom okolo 100 W. Takýto zdroj je samozrejme možné kúpiť, ale cena tých najlacnejších sa šplhá k sume 1000 €. Ekonomické dôvody boli iniciátorom toho, že som si takýto zdroj zhotovil sám.



Potreba regulácie a viacerých napätí zdroja, či väčšieho výkonu, prichádza časom s pribúdajúcimi skúsenosťami, či prechodom od jednoduchších konštrukcií k zložitejším. Čoraz častejšie som narážal na potrebu použitia viacerých napätí, alebo napätia nad 30 V, či prúdu väčšieho ako 5 A.

## Požiadavky

Praktické skúsenosti a doterajšia potreba definovali parametre budúceho zdroja. Hlavnými požiadavkami boli:

- Dve samostatné izolované vetvy výstupného napätia
- Samostatná regulácia napätia a prúdu
- Ochrany proti prúdovému preťaženiu, prehriatiu ...
- Rozsah výstupného napätia 0–50 V
- Rozsah výstupného prúdu 0–20 A
- Nízka hladina rušenia na obidve strany napájacieho zdroja (do siete, aj do napájaného obvodu)
- Čo najväčší výstupný výkon
- Jednoduché ovládanie a prijateľná cena

## Návrh riešenia

Z uvedeného zadania vyplynulo obmedzenie príkonu zdroja do 3,5 kW, aby mohol byť napájaný z jednej fázy elektrickej siete.

Nad zdrojom, dnes už klasickej konštrukcie (Obr. 1) trafo – usmerňovač – filter – regulácia – meranie, som ani neuvažoval. Dôvodom je niekoľko prozaických príčin: transformátor by bol obrovský a ťažký, chladenie usmerňovača a regulátora objemné, vzhľadom na požadovaný výstupný prúd by bloky kondenzátorov filtra museli byť tiež veľké, lebo pracujeme s filtermi pre transformátor a usmerňovač 50 Hz, ... Rozhodnutie padlo na spínaný zdroj.

Takéto konštrukcie spínaných zdrojov boli na stránkach Amatérského rádia publikované opakovane z teoretického aj praktického hľadiska. Mojim cieľom nebolo zhotoviť

spínaný zdroj z diskretných súčiastok, ale zostaviť zdroj primeraných parametrov z dostupných modulov. Podobne ako v počítačovej technike už opúšťame návrh CPU jednotiek, ale vyberieme pre naše konštrukcie niektorý z dostupných modulov CPU. Takto nejako som pristúpil k riešeniu zadanej úlohy.

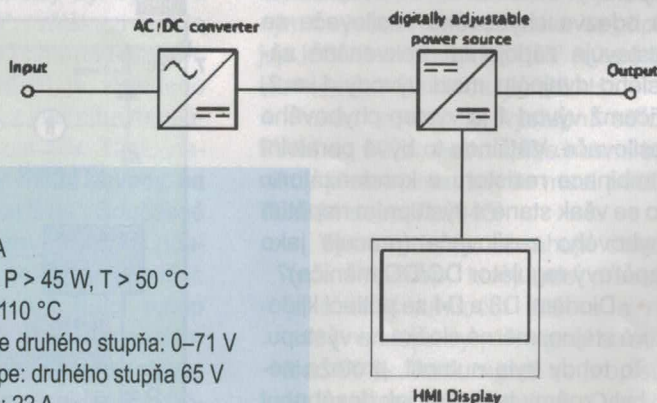
Aby sa znížil pomer medzi rozsahom vstupného a výstupného napätia, zvolil som dvojstupňové riešenie (Obr. 2).

Prvý stupeň, AC/DC prevodník je napájací spínaný zdroj, zabezpečí stabilizáciu pevného napätia na jeho výstupe, ktoré musí byť vyššie najmenej o hodnotu úbytku napätia na druhom stupni zabezpečujúcom reguláciu.

Druhý stupeň je DC/DC prevodník zabezpečujúci reguláciu výstupného na-

## → Obr. 2.

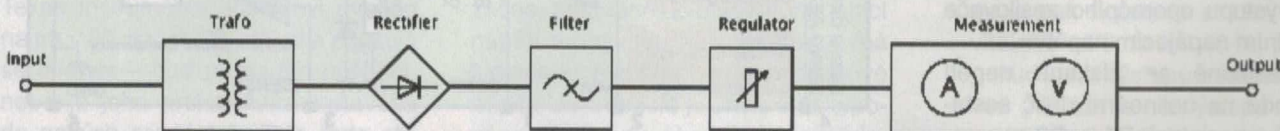
Bloková schéma – vybrané riešenie



## Technické parametre

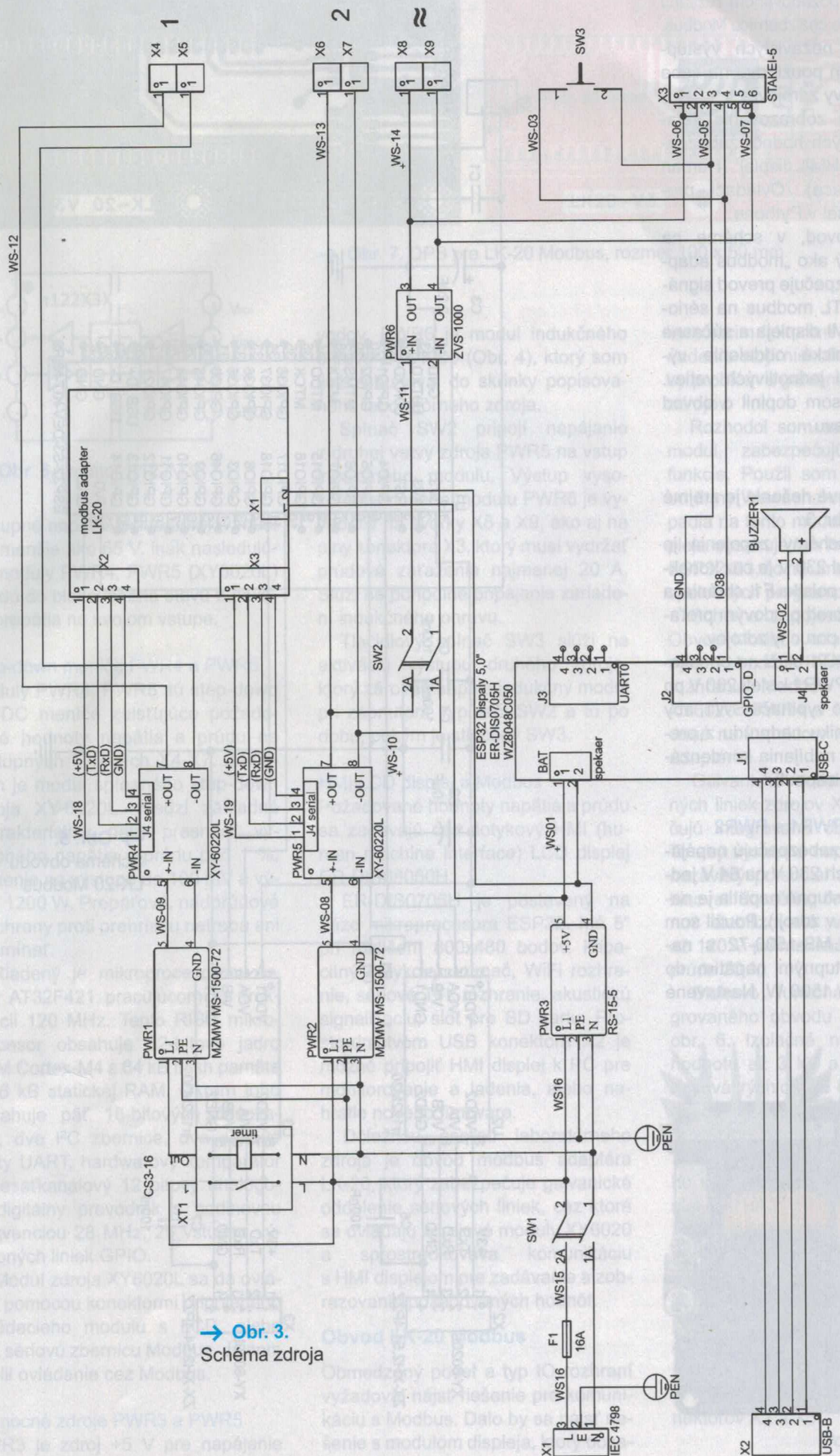
Režim meniča: step-down/buck  
Vstupné napätie: AC 195 V–245 V  
Výstupné napätie: 0–60 V  
Výstupný prúd: 0–20 A  
Kolísanie výstupu: 100 mV  $V_{pp}$   
Výstupný výkon: 1200 W

Rozlíšenie merania: 0,01 V/A  
Aktivácia ventilátora:  $I > 2$  A,  $P > 45$  W,  $T > 50$  °C  
Externá tepelná ochrana: 0–110 °C  
Prepätiová ochrana na vstupe druhého stupňa: 0–71 V  
Prepätiová ochrana na výstupe: druhého stupňa 65 V  
Prúdová ochrana na výstupe: 22 A



→ Obr. 1. Blokova schéma klasickej konštrukcie





→ Obr. 3.  
Schéma zdroja



pätia a prúdu v požadovanom rozsahu ovládaný číslicovo cez zbernicu Modbus.

Pre dvojicu nezávislých výstupných napätí som použil dve na sebe nezávislé zostavy zdrojov.

Zadávanie a zobrazovanie želaných a skutočných hodnôt zabezpečuje dotykový HMI displej (Human machine interface). Ovládací program som napísal v Pythone.

Pomocný obvod, v schéme na obr. 3 označený ako „modbus adaptér LK-20“ zabezpečuje prevod signálov rozhrania TTL modbus na sériové rozhranie HMI displeja a súčasne zaisťuje galvanické oddelenie výstupných napätí jednotlivých vetiev. Skrinku zdroja som doplnil o obvod indukčného ohrevu.

## Zapojenie

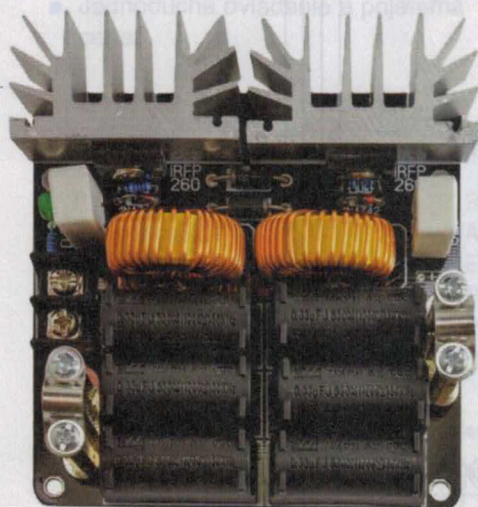
Celkové obvodové riešenie je zrejmé zo schémy na obr. 3.

Z obrázku schémy zapojenia je zrejmé, že prívod 230 V je cez konektor X1. Za ním je poisťka F1, chrániaca celé zariadenie pred prúdovým preťažením v prípade poruchy zdrojov.

Časové relé XT1 slúži na oneskorené pripojenie PWR1 k sieti 230 V po zapnutí hlavného vypínača SW1, aby sa zabránilo vzniku nadprúdu z prechodového javu nabíjania kondenzátorov zdrojov pri zapnutí.

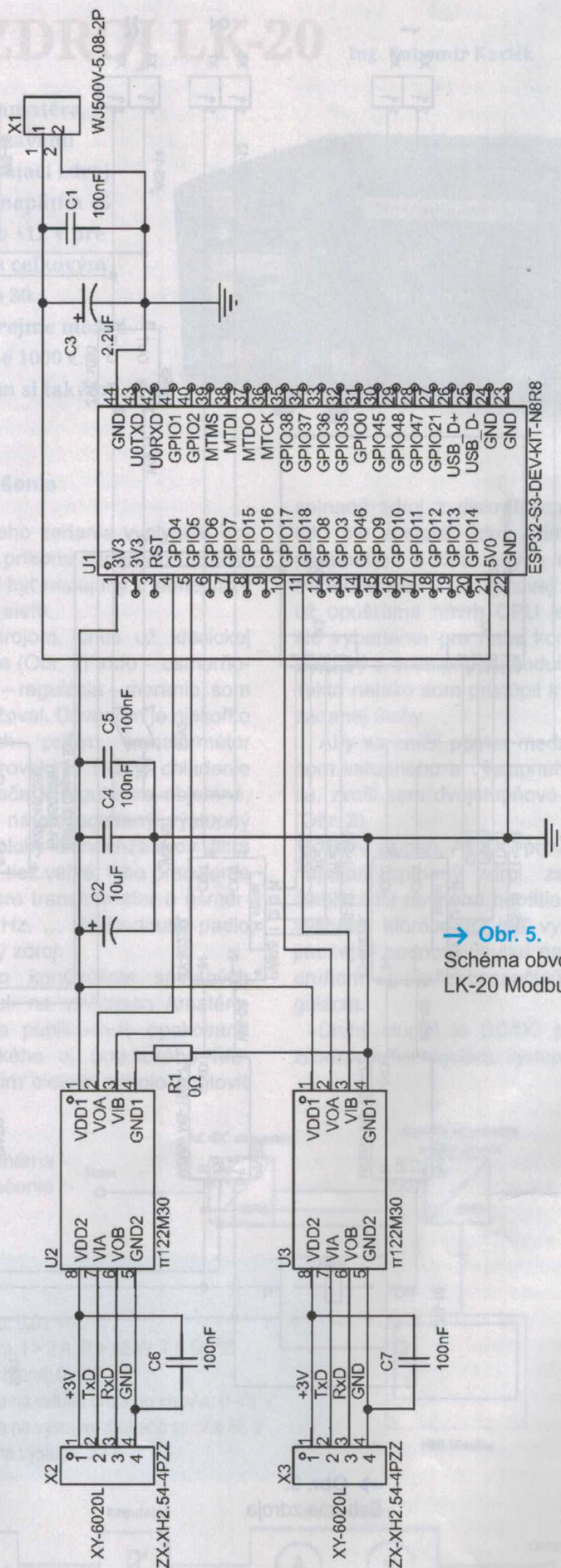
## Spínané zdroje PWR1 a PWR2

PWR1 a PWR2 zabezpečujú napájanie zo striedavých 230 V na 64 V jednosmerných (výstupné napätie je nastavené trimrom v zdroji). Použil som spínané zdroje MS-1500-72 s nastaviteľným výstupným napätím do 72 V a výkonom 1500 W. Nastavené



→ Obr. 4.

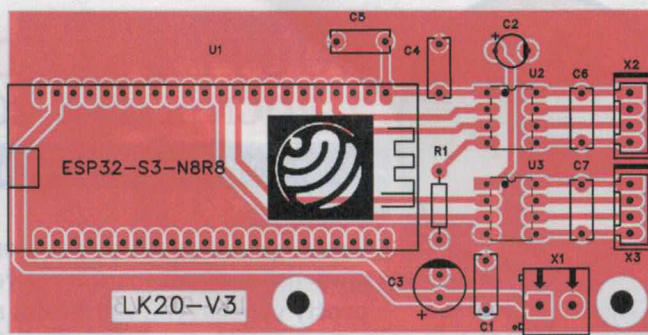
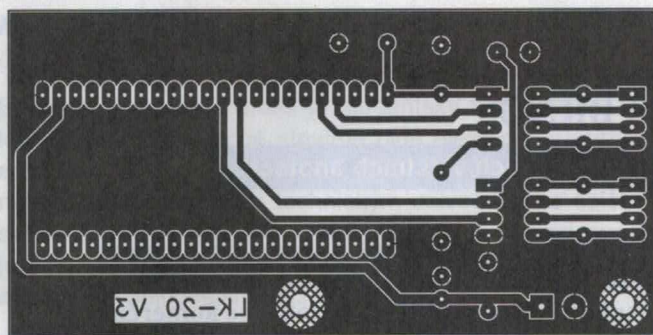
Indukčný modul ZVS 1000W



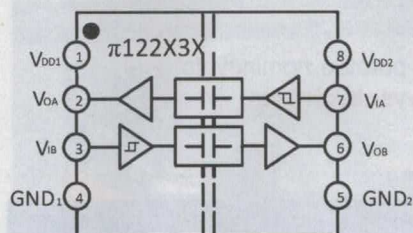
→ Obr. 5.

Schéma obvodu LK-20 Modbus





→ Obr. 7. DPS pre LK-20 Modbus, rozmer 100 x 50 mm



→ Obr. 6. Izolátor π122M30

výstupné napätie týchto zdrojov musí byť menšie ako 65 V, inak nasledujúce moduly PWR4, PWR5 (XY6020L) prejdú do blokovanej stavu z dôvodu prepätia na svojom vstupe.

#### Step-down meniče PWR4 a PWR5

Moduly PWR4, PWR5 sú step-down DC/DC meniče zaisťujúce požadované hodnoty napätia a prúdu na výstupných svorkách X4-X7. Základom je modul spínaného step-down zdroja XY-6020L. Medzi základné charakteristiky patrí presnosť výstupného napätia a prúdu pod 1 %, zvlnenie na výstupe do 100 mV a výkon 1200 W. Prepäťové, nadprúdové a ochrany proti prehriatiu netreba ani spomínať.

Riadený je mikroprocesorom Artery AT32F421 pracujúcim na frekvencii 120 MHz. Tento RISC mikroprocesor obsahuje 32-bitové jadro ARM Cortex-M4 s 64 kB flash pamäte a 16 kB statickej RAM. Okrem toho obsahuje päť 16-bitových časovačov, dve I<sup>2</sup>C zbernice, dva sériové porty UART, hardwarový komparátor a desaťkanálový 12 bitový analógovo-digitálny prevodník s hodinovou frekvenciou 28 MHz, 25 vstupno výstupných liniek GPIO.

Modul zdroja XY6020L sa dá ovládať pomocou konektormi pripojeného ovládacieho modulu s LCD, alebo cez sériovú zbernicu Modbus. Ja som zvolil ovládanie cez Modbus.

#### Pomocné zdroje PWR3 a PWR5

PWR3 je zdroj +5 V pre napájanie displeja a pomocných riadiacich ob-

vodov. PWR6 je modul indukčného ohrevu ZVS1000 (Obr. 4), ktorý som tiež zabudoval do skrinky popisovaneho laboratorneho zdroja.

Spínač SW2 pripojí napájanie z druhej vetvy zdroja PWR5 na vstup indukčného modulu. Výstup vysokofrekvenčného modulu PWR6 je vyvedený na svorky X8 a X9, ako aj na piny konektora X3, ktorý musí vydržať prúdové zaťaženie najmenej 20 A. Služí na pohodlné pripájanie zariadení indukčného ohrevu.

Tlačidlový spínač SW3 slúži na aktiváciu výstupu druhého zdroja, ktorý zároveň napája indukčný modul pri zapnutom vypínači SW2 a to po dobu, pokým je stlačený SW3.

#### HMI LCD displej a Modbus

Požadované hodnoty napätia a prúdu sa zadávajú cez dotykový HMI (human machine interface) LCD displej ER-DIS08050H.

ER-DIS0705H je postavený na báze mikroprocesora ESP32. Má 5" pri rozlíšení 800x480 bodov, kapacitný dotykový snímač, WiFi rozhranie, sériové TTL rozhranie, akustickú signalizáciu, slot pre SD kartu. Prostredníctvom USB konektora X2 je možné pripojiť HMI displej k PC pre monitorovanie a ladenia, alebo nahrať nové firmwae.

Dôležitou časťou laboratorneho zdroja je obvod modbus adaptéra LK-20, ktorý zabezpečuje galvanické oddelenie sériových liniek, cez ktoré sa ovládajú zdrojové moduly XY6020 a sprostredkováva komunikáciu s HMI displejom pre zadávanie a zobrazovanie požadovaných hodnôt.

#### Obvod LK-20 Modbus

Obmedzený počet a typ IO rozhraní vyžadoval najsť riešenie pre komunikáciu s Modbus. Dalo by sa najsť riešenie s modulom displeja, ktorý obsahuje viacero sériových liniek, ale tu je

limitujúcim faktorom cena. Takéto zariadenie stojí niekoľko sto € a zostal by problém s galvanickým oddelením zdrojov.

Rozhodol som sa vytvoriť vlastný modul, zabezpečujúci požadované funkcie. Použil som vývojový modul od firmy Waweshare, za pár €. Voľba padla na tento modul, lebo doska displeja obsahuje ten istý mikroprocesor ESP32-S3, čo mi umožnilo použiť časť spoločných programových knižníc.

Schéma zapojenia je na obr. 5. Obvod Modbus adaptéra je napájaný z 5 V zdroja PWR5. Sériové linky rozhrania Modbus modulov zdroja XY6020 (PWR4 a PWR5) sú privedené na konektory X2 a X3 Modbus adaptéra.

Galvanické oddelenie komunikačných liniek zdrojov XY6020 zabezpečujú integrované obvody π122M30. To aby výstupné napätia zdrojov mali nezávislý potenciál a mohli sa kombinovať do série, čo umožní dosiahnuť stabilizované výstupné napätie až 120 V pri maximálnom výstupnom prúde 20 A.

Blokovo funkčná schéma integrovaného obvodu π122M30 je na obr. 6. Izolačné napätie dosahuje hodnotu až 3 kV a maximálna prenosová rýchlosť je 10 Mbit. Použitý obvod je dvojkanálový a umožňuje aj prevod napäťových úrovní medzi 3 V a 5 V, pričom spotreba samotného integrovaného obvodu je menšia ako 100 μA.

Napájanie integrovaných obvodov na strane mikropočítača U1 je zabezpečené stabilizátorom napätia 3,3 V umiestneného na doske vývojového kitu U1 a vyvedeného na pin č.1. Napájanie izolačných obvodov na stranu ovládaných zdrojov XY6020 je 3 V zo zdrojov cez vývody 1 a 4 konektorov X2 a X3.

Pokračování příště.



# LABORATÓRNY ZDROJ LK-20

Ing. Lubomír Karlík

## Pokračování

Použité kondenzátory C1 až C7 majú filtračnú funkciu. Rezistor R1 má hodnotu 0 Ohm (prepojka), ktorý som použil iba pre zjednodušenie dosky plošných spojov (DPS), aby mohla byť realizovaná ako jednostranná.

Mikropočítač U1 ESP32-S3-DEV-KIT-N8R8 je napájaný zo zdroja 5 V cez svorky X1 a zabezpečuje komunikáciu protokolom Modbus so zdrojmi XY6020 cez porty GPIO4, GPIO5 a GPIO17, GPIO18. Cez WiFi, protokolom TCP, komunikuje s HMI displejom, ktorý je tiež osadený mikroprocesorom ESP32.

Použitie vývojového kitu ESP32-S3-DEV-KIT-N8R8 má svoje výhody. Predovšetkým som sa vyhol potrebe realizácie DPS metódou SMD a nemusel som riešiť pomocné obvody, napríklad pre pripojenie CPU k USB portu.

### DPS pre LK-20 modbus

Motív plošného spoja a osadenie súčiastok ukazuje obr. 7. DPS má rozmery 100 mm x 50 mm, je jednostranná a tak jednoduchá, že nepotrebuje ďalší popis a jej realizácia je zrejmá z obrázku.

### Oživenie LK-20 Modbus

Oživenie spočíva v kontrole plošného spoja na skraty a prípadné prerušenia spojov pri vŕtaní, či výrobe DPS

a naprogramovaniu CPU U1 spôsobom popísaným v ďalšej časti príspevku.

Drobný problém prináša použitie integrovaného obvodu  $\pi 122M30$ , ktorý sa vyrába iba v puzdre SOIC-8 pre SMD montáž. Problém vyriešil malý adaptér SOIC-8 na DIP8 (obr. 8) na ktorý sa dá čip  $\pi 122M30$ . Pri troche opatrnosti a s použitím tenšieho hrotu sa dá naspájať aj obyčajnou trafo spájkovačkou. Ako vývody DIP8 puzdra som použil 4 piny jednoradovej DIP lišty.

Osadená doska LK-20, Modbus adaptéra, pripojená a namontovaná na hliníkovom paneli je na obr. 9.

### Popis software

Vzhľadom na to, že prevažná časť hardware je zostavená z hotových modulov, veľký podiel úsilia muselo byť venované programovému vybaveniu. Obslužné programy sú napísané v jazyku MicroPython, čo je programovací jazyk Python optimalizovaný pre použitie s limitovanými zdrojmi.

Komunikácia medzi displejom a doskou s Modbus interface prebieha cez WiFi pomocou TCP. Do-

ska displeja je v režime klienta a doska Modbus interface je v režime AP. Všetky potrebné programy je možné stiahnuť na adrese [1].

Na strane AP-čka je hlavným programom `main_AP.py`. Do `boot.py` je potrebné doplniť riadok:

```
import main_AP
```

Týmto sa zabezpečí automatické spustenie hlavného programu pri zapnutí. Na strane klienta (HMI displej) podobne, v `main_client.py` doplniť riadok:

```
import main_client
```

Program zabezpečuje čítanie a zápis Modbus registrov z regulovaných zdrojov XY6020L.

Komunikačné rozhranie je TTL sériový port a využíva protokol MODBUS-RTU. Z továrne je ID zariadenia nastavené na 0x01 (01 hexadecimálne). Pracuje s úrovňami 3,3 V aj 5 V. Formát sériových dát je: Baudrate = 115200, 8 dátových bitov, bez parity, jeden stop bit (115200,8,N,1).

Pred oživením celého zdroja je možné zmeniť adresu ID druhého zdroja XY6020 na hodnotu 0x02 (02 hexadecimálne), alebo inú hodnotu pre diagnostické účely, pomocou vhodného nástroja. Na začiatku som použil voľne dostupný nástroj OpenModScan, ale pre opakovanie tejto procedúry som vytvoril krátky Python program, ktorý sa dá spustiť na ESP32 module po pripojení sériovej linky modulu XY6020. Modifikáciou premennej `MODBUS_ADDR` je možné zapísať ľubovoľnú *slave ID* hodnotu. Celý program je dostupný k stiahnutiu na [1].

```
from machine import UART
import time

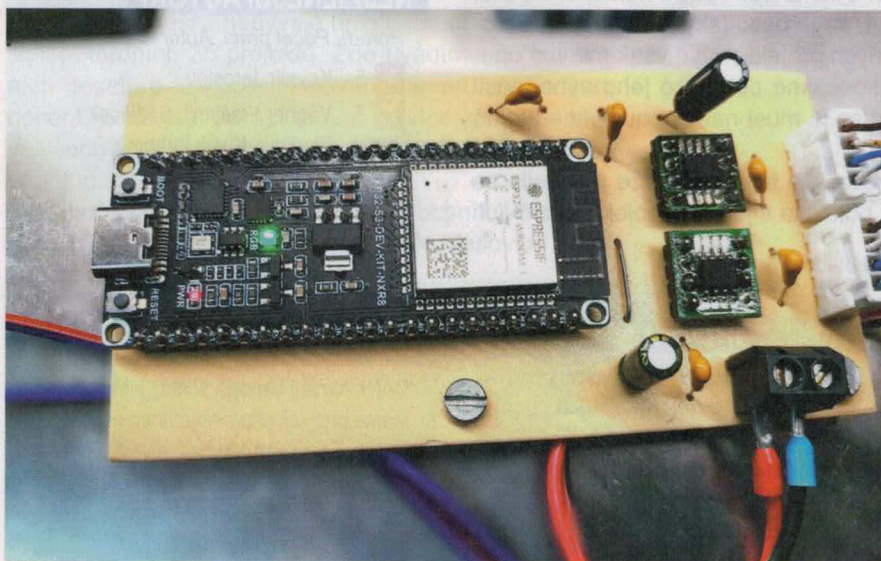
# Nastavenie UART
uart1 = UART(1, baudrate=115200,
tx=4, rx=5, timeout=500)

MODBUS_ADDR = 1 # Adresa Modbus
zariadenia
...
...
```

Ak nebola hodnota slave ID menená a je nastavená od výrobcu na hodnotu 1, nie je potrebné ju meniť, lebo modul Modbus adaptéra LK-20 komunikuje na samostatných sériových kanáloch a obidva zdroje XY6020 môžu mať rovnaké ID.



→ Obr. 8.  
Adaptér SOIC-8 na DIP8



→ Obr. 9. Osadená doska LK-20, Modbus adaptéra



```

TOTAL_REGISTERS = 32
MAX_RETRIES = 3
AP_STATUS_REG_INDEX = 32
STA_STATUS_REG_INDEX = 33
# COMMAND_REG_INDEX = 33
STATUS_ERROR_BIT = 0
STATUS_NEW_DATA_BIT = 1
STATUS_MODBUSR_BIT = 2
STATUS_SEND_DATA_BIT = 3
STATUS_WIFI_BIT = 4
STATUS_APC_BIT = 5
STATUS_STA_BIT = 6
STATUS_WRITE_BIT = 7
STATUS_TCP_BIT = 8

regw_uart1 = 0b00000000000001000000000000000011
regw_uart2 = 0b00000000000001000000000000000011
regr_uart1 = 0b0000000000000001000000000000001111
regr_uart2 = 0b0000000000000001000000000000001111

xy6020_data1 = [0] * (TOTAL_REGISTERS+ 2)
xy6020_data2 = [0] * (TOTAL_REGISTERS+ 2)

xy6020_data1w = [0] * (TOTAL_REGISTERS+ 2)
xy6020_data2w = [0] * (TOTAL_REGISTERS+ 2)

# Počet registrov XY6020
# Maximálny počet opakovaní
# Index apečka pre status, word v datovom poli
# Index pre status klienta, word v datovom poli
# Index pre priznak požiadavky zápisu v datovom poli
# nastala nejaká chyba
# bit indikuje prijatie nových dát na zápis do XY registrov
# chyby UART na modbus
# sú nové dáta k odviselaniu na server
# chyba spojenia na server, AP, alebo STA
# spracováva sa wifi požiadavka odoslania, či príjmu na strane AP
# spracováva sa wifi požiadavka odoslania, či príjmu na strane klienta
# požiadavka na zápis do registrov XY
# 1 znamená chybu v TCP komunikácii
# 32-bitové masky pre určenie ktoré registry sa majú zapísať
# Každý bit zodpovedá jednému registru (bit 0 = Reg0, bit 31 = Reg31)
# Příklad: zapíše sa len register 0,1,18
# Příklad: zapíše sa len register 0,1,18
# Příklad: číta len register 0,1,2,..
# Příklad: číta len register 0,1,2,

# Dátové polia pre registre - čítanie z XY60020
# 32 registrov + 1 status + ....
# 32 registrov + 1 status + ....
# Dátové polia pre registre- zápis do XY60020
# 32 registrov + 1 status + ....
# 32 registrov + 1 status + ....

```

Potrebné programové funkcie na komunikáciu so zdrojmi XY6020 sú obsiahnuté v knižnici *modbus.py*.

Ako už bolo povedané, WiFi HMI displeja je v režime klienta a WiFi modul jednotky ESP32 na doske modbus adaptéra je v režime AP. Inicializácia WiFi rozhraní je v knižnici *wifi\_comm.py*, ktorá je spoločná pre klienta aj AP.

Všetky knižnice, a teda aj *my\_lib.py* sú spoločné pre obidva moduly (Modbus rozhranie aj displej). Rozdielne sú iba *main\_AP.py* a *main\_client.py* pre modul displeja.

Na strane AP je inicializovaný DHCP server, ktorý prideli IP adresu klientovi pri pripojení. BSSID a heslo sú definované v knižnici *wifi\_comm.py* a môžu byť ľubovoľne zmenené. Tiež je tam vytvorený TCP server, na ktorý sa pripojí TCP klient HMI displeja s LVGL knižnicou (Light Versatile Graphics Library), umožňujúcou pohodlnú tvorbu grafických užívateľsky priateľských rozhraní.

Komunikácia je riadená prerušením na pozadí. TCP klient a server si pravidelne vzájomne aktualizujú dátové štruktúry *xy6020data1* a *xy6020data2* obsahujúce prečítané hodnoty registrov XY6020, ako aj požadované hodnoty registrov pre zápis a status registre. Pomocou nich sa riadi komunikácia a zápis do registrov a teda aj režimy štart, alebo stop zdrojov.

Dátové polia sú definované v knižnici funkcií *modbus.py* a majú 2 x 34 bytov pre každý zdroj. 34 bytov pre zápis *xy6020\_data1w[34]* a 34 *xy6020\_data1r[34]* bytov pre čítanie registrov

→ Tab. 1. Povolené Modbus funkcie

| Kód  | Funkcia                  | Popis                                           |
|------|--------------------------|-------------------------------------------------|
| 0x03 | Read register data       | Čítanie dát z jedného alebo viacerých registrov |
| 0x06 | Write single register    | Zápis do jedného registra                       |
| 0x10 | Write multiple registers | Zápis do viacerých registrov                    |

a stavových hodnôt. Obsahujú komplet-ný obraz registrov pamäťovej sady M0.

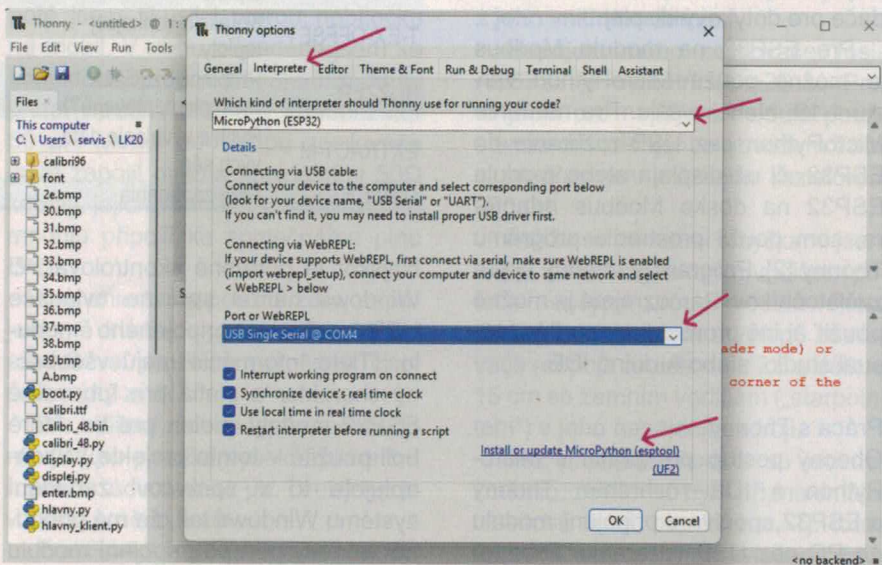
Definícia niektorých dôležitých pamäťových premenných dátových polí, či už bitových, alebo bytových hodnôt *xy6020\_data1w[]* a *xy6020\_data1r[]* vyzerá takto:

Aby sa vždy nezapisovali alebo čítali registre, ktoré nepodliehajú zmene, alebo keď obsah nie je práve zaujímavý, je zavedená bitová maska *regw\_uart1* a *xy6020\_data1w* (pre druhý zdroj *regw\_uart2* a *xy6020\_data2w*), ktorá definuje byty registrov určené pre

operáciu zápisu, alebo čítania. Jednotlivé bity určujú register v XY6020, ktorý sa bude čítať alebo zapisovať, tak ako je to v horeuvedenom príklade.

Mikropočítač na doske Modbus adaptéra nepretržite periodicky komunikuje cez opticky izolované rozhranie so zdrojmi a číta aktuálne hodnoty napätia, prúdu a stavu z ich registrov pomocou sady príkazov. Pre toto zariadenie (XY6020) sú povolené iba nasledovné Modbus funkcie v Tab. 1.

XY6020 obsahuje nasledovnú sadu registrov (Holding Registers). Viď Tab.



→ Obr. 10. Nastavenia komunikácie ESP32 a Thonny - 1



2. Každý register pozostáva z 16 bitov, prenášaných ako dvaja byte.

Okrem týchto registrov má XY6020 celkovo desať pamäťových dátových sád M0-M9. Každá sada obsahuje celkom 14 dátových registrov. Počiatočná adresa dátovej sady M0 je 0050H pre uloženie nastavení a ich rýchle vyvolanie. Dátová skupina M0 sa štandardne vyvoláva pri zapnutí výrobku, dátové skupiny M1 a M2 sú rýchlo vyvolané z panelu výrobku a M3-M9 sú bežné pamäťové dátové skupiny.

Výpočtová metóda počiatočnej adresy dátovej skupiny je: 0050H + číslo dátovej skupiny \* 0010H. Napríklad počiatočná adresa dátovej skupiny M3 je: 0050H + 3 \* 0010H = 0080H.

Vyvolanie pamäte M1-M9 sa vykoná zápisom požadovaného čísla 1..9 do registra EXTRACT-M na adrese 001DH. V dôsledku toho sa vyvolané pamäťové hodnoty skopírujú do oblasti M0 na adrese 0050H a ďalej.

M0 je aktuálna účinná operačná dátová sada. Aktualizácie tejto oblasti nadobúdajú okamžitý účinok. (Vyvolanie pamäte M0 nemá žiadny účinok.) Túto vlastnosť využívam zápisom do registrov k okamžitej zmene požadovaných hodnôt a laboratórny zdroj nevyužíva pamäťové sady M1-M9.

### Nahrание programov

Na nahrание potrebného programového vybavenia je potrebné do modulov nahráť firmware obsahujúci potrebnú verziu MicroPython a ovládacie programy.

Zvláštnu pozornosť vyžaduje HMI displej, kde odporúčam ušetriť si ťažkosti a použiť firmware výrobcu displeja, ktorý obsahuje potrebné ovládacie pre dotykový displej.

Pre ESP32 na module Modbus je možné použiť MicroPython verziu 1.19, alebo vyššiu. Pre nahrание MicroPython cez USB rozhranie do ESP32, či už displeja alebo modulu ESP32 na doske Modbus adaptéra, som použil prostredie programu Thonny [2]. Program je vhodný aj pre začiatočníkov. Samozrejme je možné použiť aj iné prostredie, napríklad Visual studio, alebo Arduino IDE.

### Práca s Thonny

Obecný postup pre prácu s MicroPython a IDE rozhraním Thonny a ESP32 spočíva v pripojení modulu na PC cez USB rozhranie. Predpokladám použitie OS Windows.

→ Tab. 2. XY6020 – Sada registrov

| Názov       | Popis                                  | Bajtov | Dec. | Jednotka | Čítanie a zápis | Adresa registra |
|-------------|----------------------------------------|--------|------|----------|-----------------|-----------------|
| V-SET       | Nastavenie napätia                     | 2      | 2    | V        | R/W             | 0000H           |
| I-SET       | Nastavenie prúdu                       | 2      | 2    | A        | R/W             | 0001H           |
| VOUT        | Zobrazovaná hodnota výstupného napätia | 2      | 2    | V        | R               | 0002H           |
| IOUT        | Zobrazovaná hodnota výstupného prúdu   | 2      | 2    | A        | R               | 0003H           |
| POWER       | Zobrazovaná hodnota výstupného výkonu  | 2      | 1    | W        | R               | 0004H           |
| UIN         | Zobrazovaná hodnota vstupného napätia  | 2      | 2    | V        | R               | 0005H           |
| AH-LOW      | Výstupné AH nízkych 16 bitov           | 2      | 0    | mAh      | R               | 0006H           |
| AH-HIGH     | Výstupné AH vysokých 16 bitov          | 2      | 0    | mAh      | R               | 0007H           |
| WH-LOW      | Výstupné WH nízkych 16 bitov           | 2      | 0    | mWh      | R               | 0008H           |
| WH-HIGH     | Výstupné WH vysokých 16 bitov          | 2      | 0    | mWh      | R               | 0009H           |
| OUT_H       | Čas zapnutia - hodiny                  | 2      | 0    | H        | R               | 000AH           |
| OUT_M       | Čas zapnutia - minúty                  | 2      | 0    | M        | R               | 000BH           |
| OUT_S       | Čas zapnutia - sekundy                 | 2      | 0    | S        | R               | 000CH           |
| T_IN        | Hodnota vnútornej teploty              | 2      | 1    | °F/°C    | R               | 000DH           |
| T_EX        | Hodnota vonkajšej teploty              | 2      | 1    | °F/°C    | R               | 000EH           |
| LOCK        | Zámok kľaves                           | 2      | 0    | -        | R/W             | 000FH           |
| PROTECT     | Stav ochrany                           | 2      | 0    | -        | R/W             | 0010H           |
| CVCC        | Stav konštantného napätia/prúdu        | 2      | 0    | -        | R               | 0011H           |
| ONOFF       | Prepínač výstupu                       | 2      | 0    | -        | R/W             | 0012H           |
| F-C         | Symbol teploty                         | 2      | 0    | -        | R/W             | 0013H           |
| B-LED       | Úroveň jasu podsvietenia               | 2      | 0    | -        | R/W             | 0014H           |
| SLEEP       | Čas do zhasnutia obrazovky             | 2      | 0    | M        | R/W             | 0015H           |
| MODEL       | Číslo produktu (bolo 6100h)            | 2      | 0    | -        | R               | 0016H           |
| VERSION     | Číslo verzie firmvéru (bolo 0071h)     | 2      | 0    | -        | R               | 0017H           |
| SLAVE-ADD   | Adresa slave (podriadeného zariadenia) | 2      | 0    | -        | RAW             | 0018H           |
| BAUDRATE_L  | Prenosová rýchlosť (6 = 115200)        | 2      | 0    | -        | RAW             | 0019H           |
| T-IN-OFFSET | Korekcia vnútornej teploty             | 2      | 1    | °F/°C    | RAW             | 001AH           |
| T-EX-OFFSET | Korekcia vonkajšej teploty             | 2      | 1    | °F/°C    | RAW             | 001BH           |
| BUZZER      | Prepínač bzučiaka (neimplementované?)  | 2      | 0    | -        | RAW             | 001CH           |
| EXTRACT-M   | Rýchle vyvolanie dátových sád          | 2      | 0    | -        | RAW             | 001DH           |
| DEVICE      | Stav zariadenia                        | 2      | 0    | -        | RAW             | 091EH           |

Najskôr je potrebné skontrolovať, či Windows nahral správne ovládacie USB rozhrania pripojeného modulu. (Tieto informácie majú všeobecný charakter a platia pre ľubovoľné ESP32 moduly, nielen pre tie, ktoré boli použité v tomto projekte.) Skontrolujete to v správcovi zariadení systému Windows tak, že má pribudnúť sériový port po pripojení modulu k PC.

Teraz môžete spustiť na svojom PC program Thonny a nahráť firmware MicroPython. Úvodná obrazovka signalizuje chýbajúci „backend“, čo znamená, že Thonny nemá aktívnu komunikáciu s pripojeným ESP32 modulom. Kliknite na položku „Options ...“ v menu „Tools“. Potom, v novootvorenom okne, viď Obr. 10, kliknite na záložku „Interpreter“. Pokračovanie v príštím čísle.



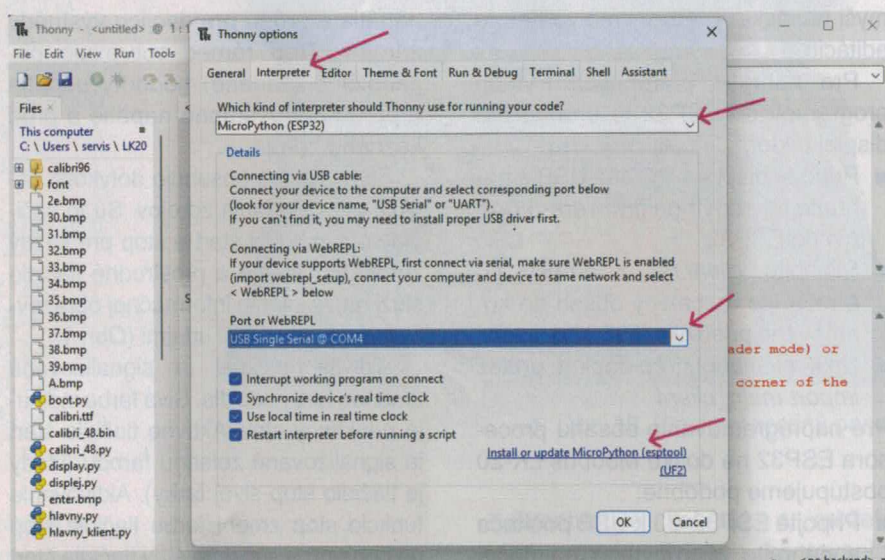
## Dokončení

Vyberte typ interpretéra „MicroPython (ESP32)“ a zvolte sériový port, ktorý už vášmu modulu pridelil Windows.

Potom kliknite na odkaz „Install or update MicroPython (esptool)“ a v nasledujúcom okne (Obr. 11) v riadku „Target port“ vyberiete sériový port priradený vášmu ESP32 modulu. V spodnej časti obrazovky rozkliknete menu a vyberiete „Select local MicroPython image...“, kde budete v nasledujúcom okne vyzvaný na zadanie názvu súboru s obrazom MicroPython. Nalistujete ho vo vašom počítači a zvolíte ho na nahratie.

Následné okno (Obr. 12) informuje, kde je cieľový port. Vyberiete typ procesora, v našom prípade ESP32S3. Riadok version ukazuje verziu zvoleného image MicroPython určeného na nahratie. Kliknutím na tlačidlo „Install“ naštartujete zápis do flash pamäte modulu ESP32.

Po úspešnom nahraťí bude obrazovka programu Thonny v záložke „shell“ obsahovať výpis konzoly vášho MicroPython systému a môžete zadávať príkazy. Viď Obr. 13.



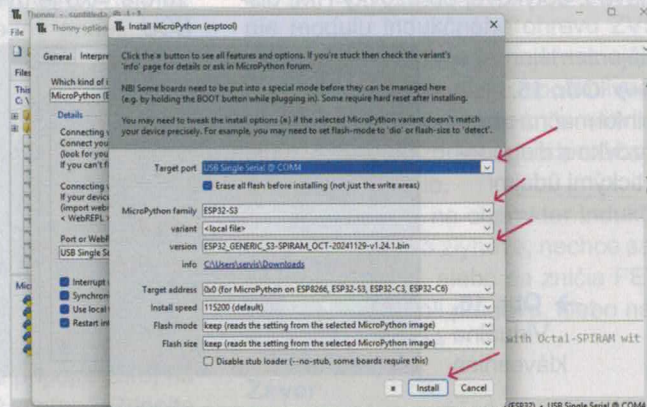
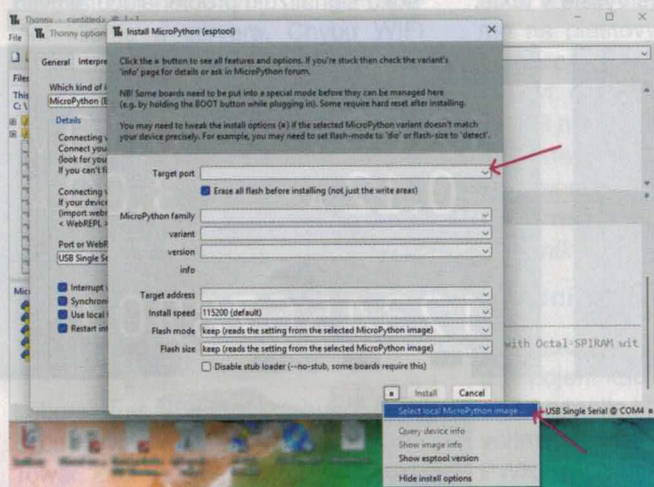
→ Obr. 10. Nastavenia komunikácie ESP32 a Thonny - 1

Stavový riadok vpravo dole vás informuje, že ste pripojený cez USB sériový port (v mojom prípade COM4) k MicroPython systému s ESP32. V ľavej časti okna vidíte záložku „Files“, ktorá v hornej časti zobrazuje obsah lokálneho priečinku vo vašom

PC a v spodnej časti súbory nahraté vo vašom module.

**Nahratie potrebných Python programov do ESP32**

Nastavte kurzor na lokálny súbor v okne Thonny, kliknite pravým tlačid-

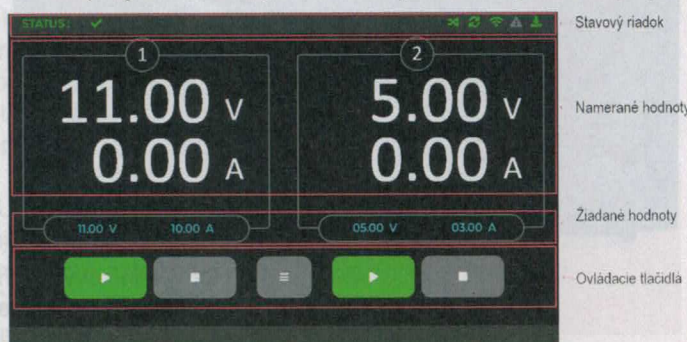
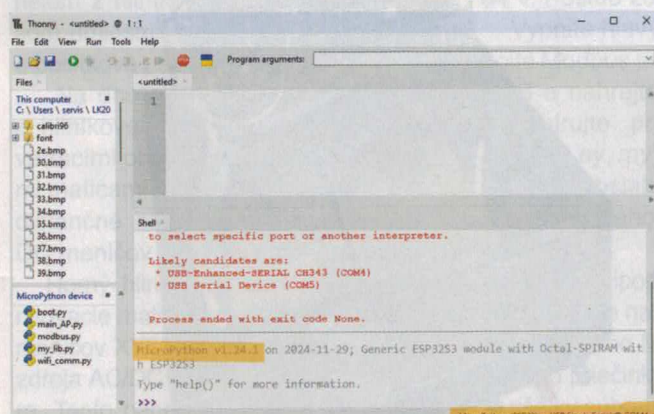


← Obr. 11. O Nastavenia komunikácie ESP32 a Thonny - 2

↑ Obr. 12. Nastavenia komunikácie ESP32 a Thonny - 3

← Obr. 13. Shell programu Thonny

↓ Obr. 14. Ovládanie z dotykovej obrazovky HMI displeja





lom myši a zo zobrazeného kontextového menu vyberte „upload ...“. Tým prekopírujete zvolený súbor do modulu ESP32. Dvojklikom ľavého tlačidla myši na niektorý súbor ho otvoríte pre editáciu.

Pre nahranie potrebných Python programov do ESP32 to urobíte pre displej takto:

- Pripojte displej k PC cez USB a nahrajte Micropython *firmware-5.0-A.bin* do ESP32.
- Stiahnite *client.rar* z adresy [1] a nahrajte rozbalený obsah do koreňového priečinka ESP32 displeja.
- Na koniec bootstrap doplňte príkaz `import main_client`.

Pre naprogramovanie obsahu procesora ESP32 na doske Modbus LK-20 postupujeme podobne:

- Pripojte ESP32-S3 k USB počítača a nahrajte Micropython najmenej verziu v1.19.1 (2022-06-18).
  - Stiahnite *ap.rar* z adresy [1] a nahrajte rozbalený obsah do koreňového priečinka ESP32 displeja.
- Na koniec *boot.py* doplňte príkaz `import main_AP`. Nahrajte všetky súbory z archívu, nielen tie s príponou „py“.

## Ovládanie zdroja

Ovládanie dvojice zdrojov sa vykonáva z dotykovej obrazovky HMI dis-

pleja, viď Obr. 14, rozdelenej do štyroch rámcov.

Prvý obsahuje stavový riadok. Druhý obsahuje skutočné hodnoty napätia a prúdu pre dvojicu výstupov zdrojov. Tretí rámec zobrazuje požadované (zadané) hodnoty napätia a prúdu pre výstupné napätie a prúd každého zdroja.

Štvrtý rámec obsahuje dotykové tlačidlá pre ovládanie zdrojov. Sú to tlačidlá so symbolmi štart a stop pre každý zdroj samostatne a prostredné tlačidlo slúži na vyvolanie informačnej obrazovky s diagnostickými údajmi (Obr. 15).

Aktivita tlačidiel je signalizovaná zmenou farby tlačidla. Sivá farba indikuje neaktívny stav. Aktívne tlačidlo štart je signalizované zelenou farbou (vtedy je tlačidlo stop sivej farby). Aktivovanie funkcie stop zmení farbu tlačidla stop na červenú a súčasne stav tlačidla štart zmení na sivú. Pre prostredné diagnostické tlačidlo je aktívny stav signalizovaný modrou farbou tlačidla.

Netreba hádam pripomínať, že funkcia štart znamená aktiváciu napätia na výstupe zdroja na zadané hodnoty a funkcia stop vypne výstupné napätie príslušnej vetvy zdroja.

Použitie hardwarového tlačidla SW3 aktivuje výstupné napätie druhého zdroja. Ako je uvedené vyššie v texte,

čo je signalizované aj zmenou farieb tlačidiel štart a stop pre druhý zdroj.

Priloženie prsta na zobrazovanú hodnotu napätia alebo prúdu v druhom rámci obrazovky, kde sa zobrazujú skutočné hodnoty napätia a prúdu, vyvolá virtuálnu klávesnicu na zadanie požadovaných výstupných hodnôt napätia alebo prúdu (Obr. 16).

Klávesnica obsahuje čísllice a potvrdzovaciu klávesu „Enter“. Zadávanie vstupných hodnôt sa ukončí stlačením klávesy „Enter“ na klávesnici, alebo zadáním štyroch znakov, podľa toho, čo nastane skôr.

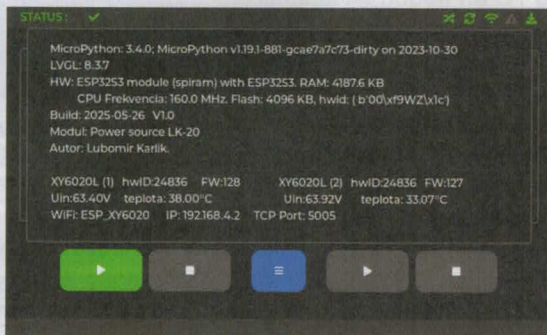
Zadávať sa hodnoty zľava doprava, ak je zadaných menej ako 4 znakov, doplnia sa nuly. Napríklad 0+5+„enter“ znamená 05.00 Voltov, alebo Ampér. Počet desatinných miest je pevný, nastavený na 2. Pri zobrazovaní výstupných hodnôt zdroja sa úvodná nula potláča.

## Stavový riadok

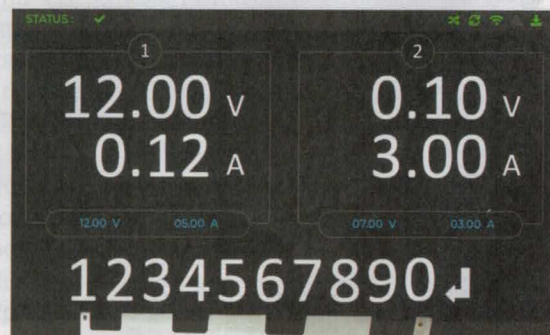
Prvý rámec na obrazovke je vyhradený indikácii stavu (Obr. 14). V ľavej časti rámca za názvom status je zobrazený zelený znak ✓, ak je všetko v poriadku alebo červené ✗, ak nastane nejaká chyba.

Ikony v pravej časti stavového riadku signalizujú prebiehajúcu aktivitu,

→ Obr. 15. informačná obrazovka s diagnostickými údajmi



→ Obr. 16. Virtuálna klávesnica



→ Obr. 17. Zadný panel



→ Obr. 18. Horný hliníkový plech pre uchytenie DC/DC meničov a kabeláže



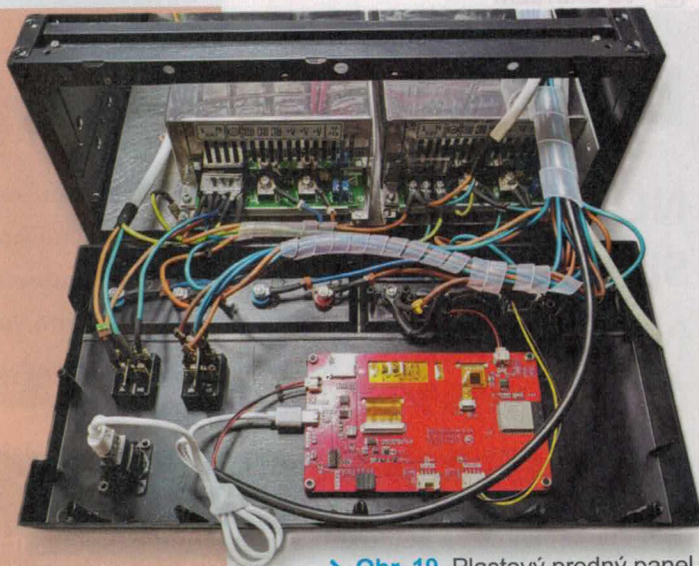


## → ZOZNAM SÚČIASTOK

|            |                                          |
|------------|------------------------------------------|
| BUZZER1    | Buzzer – reproduktor z PC                |
| F1         | ZHL70, poistkové puzdro Stelvio Kontek   |
| PWR1, PWR2 | Zdroj AC/DC ZJIVNV MS-1500-72, 2 ks      |
| PWR3       | Zdroj AC/DC MEANWELL RS-15-5             |
| PWR4, PWR5 | Zdroj DC/DC Recom Power XY-60220L, 2 ks  |
| PWR6       | Indukčný ohrev Recom Power ZVS 1000W     |
| SW1        | Vypínač Marquardt, typ 1835.3114         |
| SW2        | Vypínač Marquardt, typ 1835.3128         |
| SW3        | Spínač PC XKB, typ MINIGI                |
| X1         | Konektor IEC 4798 Shurter, typ 4798.9000 |
| X2         | USB-B panelový konektor CLIFF            |
| X2, X3     | Konektor JST ZX-XH2.54-4PZZ, 2 ks        |
| X3         | Konektor HIRSCHMANN STAKEI-5             |
| X4, X6     | Zdierka Hirschmann PKI 10 A RD, 2 ks     |
| X5, X7     | Zdierka Hirschmann PKI 10 A BU, 2 ks     |
| X8, X9     | Zdierka Hirschmann PKI 10 A BK, 2 ks     |
| XT1        | Časový spínač Zamel CS3-16               |

### Použité súčiastky pre dosku Modbus LK-20

|                    |                                        |
|--------------------|----------------------------------------|
| C1, C4, C5, C6, C7 | 100 nF keramický kondenzátor, 5 ks     |
| C2                 | 10 $\mu$ F elektrolytický kondenzátor  |
| C3                 | 2,2 $\mu$ F elektrolytický kondenzátor |
| R1                 | 0 $\Omega$ rezistor                    |
| U1                 | Vývojový modul, ESP32-S3-DEV-KIT-N8R8  |
| U2, U3             | Integrovaný obvod $\pi$ 122M30, 2 ks   |
| X1                 | Svorkovnica WJ500V-5.08-2P             |
| X2, X3             | Konektor ZX-XH2.54-4PZZ, 2 ks          |



→ Obr. 19. Plastový predný panel

alebo zdroj chyby. Ikony zľava doprava signalizujú zmenou farby a akustickým signálom: Chybu na Modbus zbernici zdrojov, Chybu na TCP spojení klienta a servera, Chybu WIFI spojenia. Ikona trojuholníka s výkričníkom signalizuje oranžovou farbou prehriatie modulu XY6020. Posledná ikona blikaním signalizuje načítavanie veľkých fontov počas bootovania.

Pre každú ikonu platí nasledovné pravidlo: neznámy stav je signalizovaný sivou farbou, bezchybný stav zelenou farbou a chybový stav červenou farbou.

### Mechanická konštrukcia

Laboratórny zdroj som zabudoval do počítačovej skrine MC MF-320B Micro Tower. Zadný kovový panel skrine som sčasti odstránil a nahradil vlastným panelom z hliníkového plechu, na ktorý som umiestnil zásuvku IEC 230V/16A a poistkové puzdro. Viď Obr. 17.

Na dno skrine som uchytil panel z hliníkového plechu, s vyrezanými vetracími otvormi a osadený nitovacími maticami, ktoré zároveň slúžia ako distančné stĺpiky pre uchytenie AC/DC meničov v prvom stupni zdroja.

Horný hliníkový plech má rovnako nitovacie matice, pre uchytenie DC/DC meničov XY6020, indukčného meniča, zdroja AC/DC +5 V a Modbus adaptéra. Tento panel je uchytený na zadnú

časť skrine klavírovým pántom, aby bol dobrý prístup ku kabeláži. Viď Obr. 18.

Výstupné svorky, konektor, vypínače a HMI displej sú uchytené priamo na plastový predný panel. USB porty som zo skrine odstránil a otvory prekryl malým panelom, ktorý som nalepil na čelo skrine. Viď Obr. 19.

Výkresy vŕtania panelov a ďalšia dokumentácia je dostupná z [1] alebo webu redakcie časopisu *praktická elektronika.cz/files2articles/*.

### Oživenie a nastavenie

Po mechanickom zostavení zdroja a pripojení kabeláže pripojte zdroj na sieť a zapnite hlavný vypínač. Pripojte voltmeter na výstupné svorky zdroja PWR1 a pomocou potenciometra na zdroji nastavte výstupné napätie na 64 V. Postup zopakujte aj pre PWR2.

Vypnite hlavný vypínač zdroja a pripojte Modbus adaptér LK-20 k PC cez USB a nahrajte Micropython do U1. Nakopírujte programy *main\_AP.py*, *modbus.py*, *my\_lib.py*, *wifi\_comm.py*, ktoré sú obsiahnuté v archíve *ap.rar*, do koreňového priečinka procesora na U1.

Potom pripojte USB kábel z PC na HMI displej a nahrajte firmware displeja obsahujúceho Micropython, a do koreňového priečinka CPU displeja nahrajte všetky súbory z archívu *client.rar*.

Skontrolujte kabeláž a môžete zdroj zapnúť. Priebeh bootovania, nadväzovanie Modbus komunikácie a TCP cez WIFI bude signalizované v hornom stavovom riadku a malo by skončiť stavom STATUS: ✓. Ak je to tak, váš zdroj je pripravený k činnosti. Ak nie, s najväčšou pravdepodobnosťou ste urobili chybu v kabeláži.

### Poznámky

Praktickým denným používaním zdroja sa ukázalo, že zabudovanie modulu indukčného ohrevu ZVS 1000 W (PWR6) nie je najšťastnejšie riešenie pre jeho nízku spoľahlivosť a prípadným záujmom o stavbu laboratórneho zdroja neodporúčam jeho zabudovanie.

Blok je citlivý na charakter indukčnej záťaže, často zlyháva, nechce samovoľne kmitať, alebo sa zničia FET bez toho, aby boli prúdovo, alebo napäťovo preťažené.

### Záver

Ukazuje sa, že bez využitia mikro-počítačov a elektronických riadiacich jednotiek (ECU) je dnes obtiažne a neefektívne zostavovať samostatné funkčné celky. Tento projekt ukazuje možnosti využitia moderných profesionálnych dielov v amatérskej praxi s použitím vysokoúrovňového jazyka Python, ktorý je ľahké si osvojiť. ■

### → ZDROJE

- [1] <https://dedeideas.eu/index.php/sk/laboratory-zdroj-s-xy6020/>
- [2] <https://thonny.org/>