

Pavel Schauer
Automatizace měření

2015

Obsah

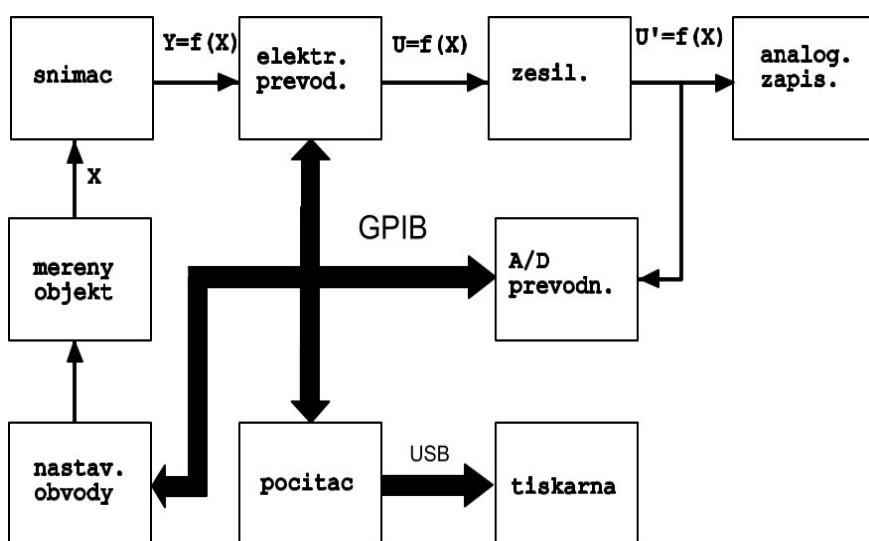
Úvod.....	4
Snímače měřených veličin	5
Odporové snímače.....	5
Kontaktní snímače.....	5
Potenciometrické snímače.....	5
Tenzometrické snímače.....	6
Teplotní odporové snímače	6
Odporové snímače záření.....	7
Indukčnostní snímače.....	7
Kapacitní snímače.....	8
Propojovací soustava (sběrnice) GPIB	9
Využití počítače PC k automatizaci měření	9
Komunikace s přístroji přes sběrnici GPIB.....	10
Příkazy (kódová slova) GPIB, sjednocená pro všechny GPIB od různých výrobců	11
Propojovací kabely a konektory GPIB.....	12
Řídící slovo, které určuje výrobce přístroje	12
pristroj=instrument("GPIB0::XX::INSTR")	12
pristroj.write(retezec).....	13
pristroj.read(retezec),	13
Adresa přístroje.....	13
Popis přístrojů se sběrnici GPIB od firmy Agilent	14
Multimetr Agilent 34401A.....	14
Popis.....	14
Funkce.....	15
Technické údaje.....	15
Řídící slova multimetru Agilent 34401A.....	15
Formát zobrazení na displeji přístroje.....	16
Formát zobrazení na displeji pro různá nastavení.....	17
Formát výstupu dat do Vašeho programu	17
Číslicově řízený zdroj Agilent E3631A	18
Popis.....	18
Funkce.....	19
Technické údaje.....	20
Řídící slova zdroje Agilent E3631A	20
Programovatelný funkční generátor Agilent 33120A	22
Popis.....	22
Funkce.....	22
Technické údaje.....	23
Řídící slova generátoru Agilent 33120A.....	23
Seznam vybraných parametrů:.....	24
Příklady	24
Osciloskop Agilent 54600A	25
Popis.....	25
Technické údaje.....	26
Program (skript) pro ovládání jednoduchého měřícího celku s možností simulace	28
Simulace měření.....	28
Vytvoření „exe“ souboru z „pythonovského“ programu.....	30
Ukládání dat do souboru a cyklus	32
Ukázky	35

Poznámka k jednotce visa_simul.py	35
Ukázka programu, jednoduché měření s použitím jednoho přístroje a simulační jednotky	35
Výstup programu jednoduché měření s použitím jednoho přístroje do souboru	36
Úprava výstupu do souboru s využitím formátovacího řetězce	37
Formátovací řetězec	37
Ukázka části programu, který zajistí požadovaný výstup	38
Ukázka měření elektrického odporu virtuálně	39

Úvod

V současné době, kdy je dostupná kvalitní výpočetní technika, je vhodné měření fyzikálních veličin, které se bude častěji opakovat, automatizovat. Podstatně tím snížíme pracnost měření a zajistíme si značné množství velmi kvalitních informací o sledovaném jevu. Automatizace měření však skrývá jedno nebezpečí, může dojít k zastínění podstaty sledovaného jevu. Experimentátor by neměl při automatizaci experimentu přestat vnímat podstatu prováděného experimentu, a to zejména tehdy, když měření sestaví a naprogramuje jiná osoba. V dalších odstavcích si podrobněji všimneme automatizované měřicí aparatury, snímačů měřených veličin, propojovací soustavy, využití počítače PC a seznámíme se s některými přístroji vhodnými k automatizaci měření.

Automatizovaná měřicí aparatura



obr. 1 Blokové schéma měřicí aparatury

K automatizaci jsou vhodná zejména měření elektrických veličin, která mají následující přednosti: lze u nich dosáhnout velké přesnosti, citlivosti, rychlosti měření, možnost dálkového přenosu veličin, možnost dalšího zpracování a analýzy, možnost záznamu, možnost paralelního měření na větším počtu měřících míst a jiné. Proto, pokud měříme neelektrické veličiny, snažíme se je převádět na veličiny elektrické pomocí snímačů. Typické blokové schéma automatizované měřicí aparatury je na .

Snímač sejme měřenou fyzikální veličinu X a převede ji na elektrickou veličinu Y . Na výstupu snímače je tedy elektrická veličina $Y = f(X)$. Přenosovou charakteristiku $Y = f(X)$ musíme dobře znát a ta musí mít vyhovující tvar (nejlépe lineární funkce). Dbáme a to, aby snímač co nejméně zatěžoval měřený objekt a aby okolní vlivy (kolísání teploty, vlhkost, vnější elektromagnetické pole, ...) co nejméně narušovaly činnost snímače. Podle okolností vkládáme do měřicí aparatury elektrický převodník, na jehož výstupu je vybraná elektrická veličina, a to elektrické napětí. Elektrický převodník tedy převede obecnou elektrickou veličinu na elektrické napětí a můžeme ho vynechat, když na výstupu snímače už elektrické napětí existuje. Dalším významným blokem měřicí aparatury je zesilovač. Má za úkol výrazně zesílit napěťový signál, který je na výstupu snímače nebo elektrického převodníku většinou velmi slabý. Výstupní napětí na zesilovači by mělo dosahovat řádové hodnoty 1 V. Zesilovač plní ještě jednu významnou funkci. Zajišťuje převod velké výstupní impedance snímače nebo elektrického převodníku na nízkou impedanci na výstupu zesilovače. Je to analogie s vlastnostmi elektrického zdroje, u něhož požadujeme malý vnitřní odpor. Nízkou impedancí na výstupu zesilovače vyžadují další připojované bloky, analogový zapisovač (nebo jen ručkový voltmetr), který umožňuje kontrolu a slouží ke zkušebním měřením a analogově-digitální (A/D) převodník, který má za úkol převést elektrické napětí na jeho číselné hodnoty, které můžeme zapisovat do paměti počítače. Kromě těchto jednotek může měřicí aparatura ještě obsahovat

nastavovací obvody řízené počítačem, které zajišťují nastavení správných podmínek měření a periferie počítače, jako je např. tiskárna, zapisovač a pod. Počítač je s jednotkami, se kterými spolupracuje, propojen speciální sběrnicí GPIB, o které podrobněji pojednáme v odstavci a s tiskárnou sběrnicí LPT.

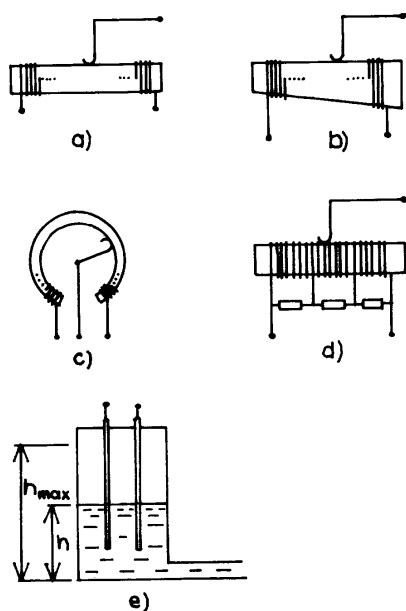
Je vhodné si uvědomit, že velmi často jsou jednotky elektrický převodník-zesilovač-A/D převodník-voltmetr konstruovány jako jeden celek a tvoří komerčně vyráběný elektrický přístroj, např. impedanční most, umožňující měřit odpory, indukčnosti a kapacity.

Z předchozího výkladu vyplývá, že při automatizaci experimentu hrají významnou roli zejména snímač a řídicí počítač se svými sběrnicemi. Proto se dále zaměříme především na tyto dvě jednotky.

Snímače měřených veličin

Jak jsem se již zmínili v odstavci , snímač zabezpečuje sejmutí měřené veličiny a převedení na elektrickou veličinu. V následujících odstavcích se zaměříme na **pasivní snímače**, které sami nejsou zdrojem elektrické energie, nýbrž vyžadují vnější zdroj elektrického proudu. U pasivních snímačů se působením neelektrické veličiny mění některý z parametrů snímače, např. elektrický odpor, indukčnost, kapacita a pod. Tyto snímače neodebírají z měřených objektů energii. Jinak je tomu u **aktivních snímačů**, které sice nevyžadují napájení vnějším zdrojem elektrického proudu, protože odebírají energii ve formě tepla, světla, pohybu a pod. z měřeného objektu, tím však měřený objekt zatěžují. Pasivní snímače bývají citlivější a stabilnější. Z těchto důvodů bývají pasivní snímače výhodnější.

Dále popíšeme činnost těch snímačů neelektrických veličin, které přicházejí v úvahu ve stavebnictví.



obr. 2 Potenciometrické snímače

Odporové snímače

Odporový snímač je v podstatě rezistor s proměnným odporem zabezpečující změnu jeho velikosti v závislosti na měřené veličině. Z hlediska konstrukce si všimneme následujících odporových snímačů:

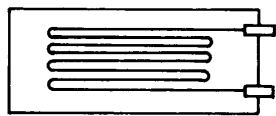
Kontaktní snímače

Působením neelektrické veličiny mění tento snímač hodnotu odporu skokem, a to přepínáním kontaktů. Jde o indikaci zvoleného počtu hodnot. Podle uspořádání kontaktního systému rozlišujeme kontakty **mechanické** a **kapalinové**. Kontaktní snímače se používají k indikaci geometrických rozměrů, teploty, tlaku, zrychlení, otáček, výšky hladiny a pod.

Potenciometrické snímače

Tyto snímače mají pohyblivý kontakt, jehož poloha je vázána na měřenou veličinu. Takto měníme odpor mezi začátkem (případně koncem) rezistoru a pohyblivým kontaktem (jezdcem). Různá závislost odporu na měřené veličině se dosáhne buď změnou hustoty vinutí nebo tvarem tělíska, na

kteří se navijí odporový drát. Příklady konstrukce potenciometrických snímačů jsou na . Potenciometrické snímače umožňují získat libovolný tvar převodní charakteristiky. Nedostatkem je poměrně velké tření pohyblivého kontaktu o povrch odporového vodiče. Tak může dojít k jeho opotřebení a změně přechodové charakteristiky snímače. Nejčastěji se potenciometrické snímače používají při měření posuvů a stočení v oblasti statických měření. Tím jsou rovněž vhodné k měření výšky hladiny, tlaku a sil. Někdy se také používají k měření vibrací, zrychlení a geometrických rozměrů.



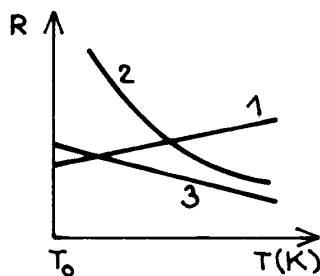
obr. 3 Drátový tenzometrický snímač

Tenzometrické snímače

Tenzometrické snímače jsou odporové snímače, využívající změnu odporu vlivem změny geometrických rozměrů nebo změnu odporu vlivem působením mechanických sil. Nejčastěji se používá **drátový tenzometrický snímač**. Je to tenký odporový drátek nalepený na podkladové papírové fólii (). Jako odporového materiálu se používá kovu (konstantan, chrom, nikl, platina-iridium a pod.) nebo polovodiče (typ P nebo typ N). Polovodičové tenzometry jsou 100 i vícekrát citlivější než kovové.

Na zkoumanou konstrukci se tenzometrické snímače lepí. Vrstva lepidla musí být tenká a dostatečně tuhá, aby se deformace přenášela co nejpřesněji na snímač. Vhodná jsou organická lepidla.

U tenzometrických měření je nezbytné dobře stabilizovat teplotu, protože ta silně ovlivňuje odpor tenzometru nebo použít kompenzačního zapojení tenzometru. Nepřímé důsledky může mít i vlhkost prostředí v okolí tenzometru. Tenzometrické snímače se používají pro statická a dynamická měření tlakové (tahové) síly, tlaku (i diferenciálního), kroutícího momentu, zrychlení a dalších veličin, které se dají převést na rozměrovou deformaci.



obr. 4 Závislost odporu na teplotě pro kov (1), polovodič (2) a přechod PN (3)

Teplotní odporové snímače

Tyto snímače využívají závislost odporu vodičů a polovodičů na teplotě. Z kovů jsou nejvhodnější platina, nikl, měď a slitina mědi a stříbra. Polovodičová čidla, kterým říkáme termistory, jsou nejčastěji na bázi kyslíčků různých kovů (Fe_2O_3 , TiO_2 , CuO , NiO a pod.). Závislost odporu R kovového teploměru na teplotě T lze vyjádřit vztahem

$$R = R_0[1 + \alpha(T - T_0) + \beta(t - t_0)^2 + \dots] \quad (1)$$

kde R_0 je odpor při základní teplotě T_0 , α a β jsou teplotní součinitelé odporu. Pro malý rozsah teplot se uvažují jen první dva členy vztahu .

Závislost odporu termistoru na teplotě má tvar

$$R = R_0 \exp[\beta(\frac{1}{T_0} - \frac{1}{T})] \quad (2)$$

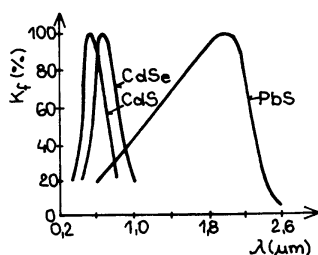
kde R_0 , T_0 mají význam jako v rovnici , β je součinitel závislý na materiálu termistoru.

Přibližně stejné citlivosti jako u kovů lze dosáhnout u teplotních čidel na bázi přechodu PN (polovodičová diody). Jejich závislost odporu na teplotě se řídí rovněž rovnicí , jako u kovů, součinitel α je však záporný.

Grafické závislosti, odpovídající rovnici pro kov (1) a polovodičovou diodu (3) a odpovídající rovnici pro termistor (2), jsou na . Předností termistorů a zejména polovodičových diod jsou jejich velmi malé rozměry (objem $< 1 \text{ mm}^3$), takže mají malou tepelnou setrvačnost. Kovová teplotní čidla musí být značně rozměrnější (délky řádově cm) a i tak nedosahují tak vysokých rezistancí jako polovodičová čidla a diody. Nevýhodou termistorů je jejich malá časová stálost, která je podstatně horší než u kovů a horší než u polovodičových diod. Cenově jsou nejlevnější polovodičové diody. Ty se celkově ukazují jako nejvýhodnější. Pro správnou funkci odporového čidla musí být proud jím procházející co nejmenší, aby se tímto proudem čidlo neohřívalo vzhledem k jeho rozměrům.

Snímače teploty se používají rovněž k dalším aplikacím, jako k měření tepelného toku, tepla, ale také rychlosti proudění kapalin a plynů (na principu ochlazení čidla vyhřívaného elektrickým proudem). Na posledně jmenovaném principu pracují rovněž odporové vakuometry a analyzátoři plynů.

Odporové snímače záření



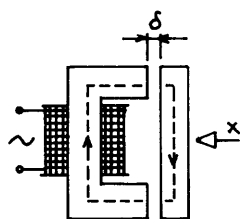
obr. 5 Spektrální charakteristiky pro některé fotosnímače

Odporové snímače záření jsou fotoelektrické snímače. Využívají závislosti různých, zejména polovodičových součástek (fotodiod, fotoodporů, fototranzistorů) na osvětlení. Činnost fotoelektrických snímačů je založena na změně elektrického odporu polovodiče působením elektromagnetického záření. U většiny fotocitlivých polovodičů odpor při ozáření klesá.

Rozhodující pro aplikace fotoelektrických čidel je jejich spektrální charakteristika. Udává závislost měrné citlivosti fotosnímače ($v \%$) na vlnové délce dopadajícího světla. Spektrální charakteristiky pro některé fotocitlivé polovodiče jsou znázorněny na . U fotosnímačů tedy musíme rozlišovat, v které oblasti vlnových délek má čidlo pracovat. Zda v oblasti viditelného světla, infračervení, ultrafialového záření nebo roentgenova či dokonce jaderného záření.

Velmi podstatná je rovněž lux-ampérová charakteristika udávající závislost fotoproudu snímačem na jeho osvětlení. Kromě parametrů záření je možno fotosnímačem nepřímo měřit znečištění kapalin a plynů, tloušťky poloprůsvitných látek, otáčky, jakost povrchu, rosný bod, mikrometrický posuv, ale také počítat výrobky na pásu, spínat vodovodní ventily. Nepřímo je rovněž možné pomocí fotosnímačů měřit teplotu, protože všechna tělesa vyzařují elektromagnetické tepelné záření.

Specifické postavení mezi odporovými snímači záření mají **bolometry**. Bolometr je odporový snímač infračerveného záření, pracující na principu pohlcování infračerveného záření, přičemž se mění jeho teplota a tím i jeho odpor. Pracuje v oblasti vlnových délek $0,8 \mu\text{m}$ až $50 \mu\text{m}$. Velkou výhodou bolometrů je relativně malá setrvačnost (časová konstanta).

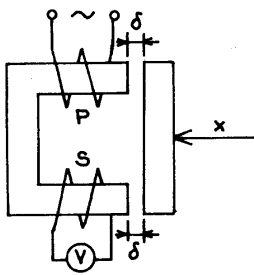


obr. 6 Indukční snímač pracující s proměnnou vzduchovou mezerou

Indukční snímače

Cívka, nebo dvě cívky, jejichž indukčnost se mění působením měřené veličiny, mohou být použity jako **indukční snímač**. Indukčnost cívky je závislá na počtu závitů, na tvaru feromagnetického jádra, na tvaru cívky a na permeabilitě obvodu. Nejčastěji se ve snímačích využívá změny geometrických rozměrů magnetického obvodu, hlavně

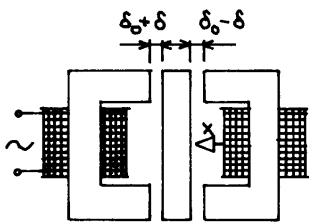
vzduchové mezery.



obr. 7 Transformátorový snímač s proměnnou vzduchovou mezerou

Nejjednodušší a nejrozšířenější indukčnostní snímač je na , pracující s proměnnou šířkou vzduchové mezery. Potřebujeme-li realizovat indukčnostní diferenciální snímač, je možné jej realizovat podle . Přesnější snímač s proměnnou vzduchovou mezerou je transformátorový snímač, znázorněný na . Indukčnostní snímače nejsou tak přesné jako odporové, protože kromě teplotních vlivů má na měření vliv i frekvence napájecího napětí. Vzhledem k tomu, že se většinou používají diferenciální snímače v můstkovém zapojení, uvedené vlivy jsou minimální.

Kapacitní snímače

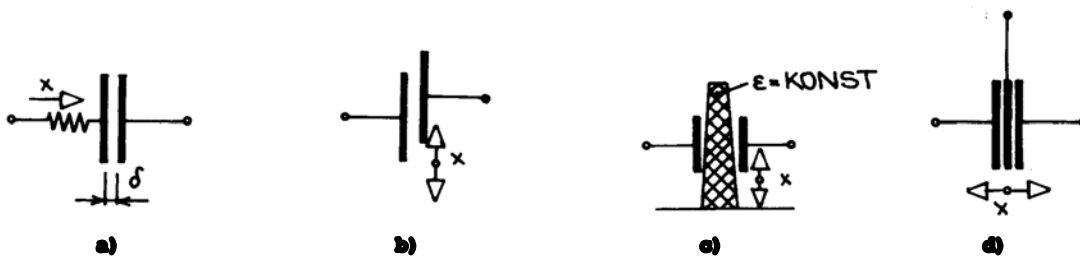


Prvky, měnící svoji kapacitu nebo dielektrické ztráty v závislosti na měřené veličině, jsou vhodné jako **kapacitní snímače**. Principiálně vychází z rovnice pro kapacitu deskového kondenzátoru

obr. 8 Indukčnostní diferenciální snímač

$$C = \frac{\epsilon S}{d}, \quad (3)$$

kde ϵ je permitivita dielektrika mezi deskami, S je plocha desek a d je jejich vzdálenost.



obr. 9 Kapacitní snímač pracující s proměnnou šířkou vzduchové mezery (a), s proměnnou plochou desek (b), s proměnnou permitivitou (c) a diferenciální kapacitní snímač (d)

Nejjednodušší kapacitní snímače pracují s proměnnou šířkou vzduchové mezery (a) nebo s proměnnou plochou desek (b), případně s proměnnou permitivitou (c). Kapacitní snímače se dají rovněž realizovat jako diferenciální, jak ukazuje d.

Určitá nepřesnost kapacitních snímačů vzniká parazitními kapacitami přívodů spojujících snímač s obvody připojeného přístroje. Parazitní kapacity snižují citlivost snímače. Další nevýhodou kapacitních snímačů je potřeba pomocného zdroje napětí vyšších frekvencí.

Kapacitní snímače s proměnnou šířkou vzduchové mezery se používají na měření malých posuvů (0,1 μm až 0,1 mm), ale rovněž k měření rychlosti posuvů, zapojí-li se do obvodu se stejnosměrným proudem a sleduje-li se vzniklý přechodový jev během posuvu, tj. změny kapacity. Snímače s proměnnou plochou se používají k měření větších posuvů (nad 1 cm) nebo úhlových posuvů (do 270 $^\circ$). Snímače s proměnnou permitivitou se k měření posuvů, jak ukazuje b, používají zřídka. Jsou však velmi vhodné k měření vlhkosti tuhých a sypkých materiálů, k měření tloušťky izolačních

materiálů a tlaků (permitivita některých materiálů se mění v závislosti na tlaku).

Propojovací soustava (sběrnice) GPIB

Podmínkou toho, aby bylo možné počítačem řídit měření, je vybavení počítače a přístrojů vhodnou **sběrnici**, tj. **propojovací soustavou**. Vhodnější propojovací soustava je paralelní, ke které připojujeme svazky vodičů vedoucí k jednotlivým přístrojům (obecně periferiím) paralelně (vedle sebe) podobně, jako připojujeme elektrické spotřebiče k jedné síťové zásuvce. Paralelní propojovací soustava umožňuje rychlejší přenos informací a připojení většího počtu periferií. Každá sběrnice umožňuje **adresování** jednotlivých periferních zařízení, přenos **řídících signálů** pro tyto jednotky a **přenos dat**, tj. naměřených hodnot. Propojovací soustava je tvořena mnohožilovým vedením a bývá standardizována. V našich podmínkách využijeme nejčastěji sběrnici **GPIB** (General Purpose Interface Bus = Sběrnice pro obecné použití).

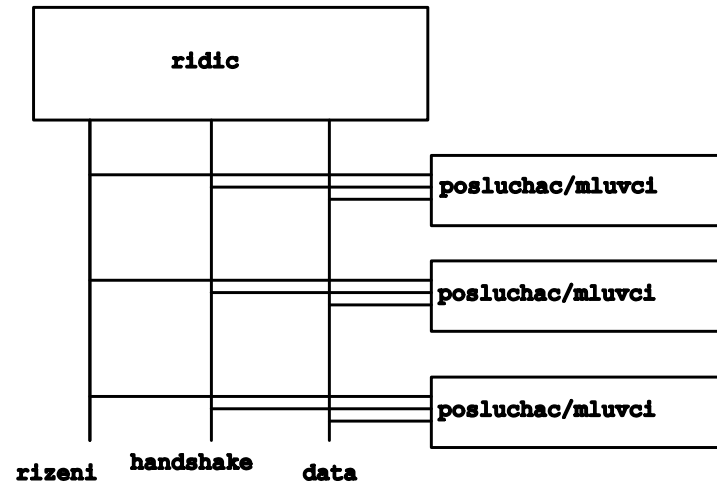
Sběrnice (anglicky interface bus) GPIB byla navržena v roce 1972 firmou Hewlett - Packard (USA). Zkratka jejího názvu pochází z anglického názvu Hewlett-Packard Interface Bus a v USA bývá označována **IEEE-488**. Ve východním bloku (pro státy bývalé RVHP) byl v roce 1986 zaveden obdobný systém s označením **IMS-2** (ČSN 356522), který se lišil jen konektory.

Soustava GPIB umožňuje sestavení automatického systému měření vytvořeného ze samostatných přístrojů (vybavených touto sběrnici), které jsou navzájem propojeny "kabely". Sběrnice (komunikační kabel) GPIB má celkem **16 vodičů** umožňujících **přenos oběma směry**, jejich rozpis ukazuje obr. 11. **Maximální počet** mezi sebou komunikujících **zařízení je 15**, při větším počtu jsou vyžadovány speciální úpravy. Délka propojení **mezi dvěma zařízeními** smí být **maximálně 2 m**. **Celková délka** propojení je maximálně **20 m**. **Rychlost přenosu** údajů přes kterýkoliv vodič je maximálně **1 Mbit/s**. Centrální jednotkou bývá obvykle **počítač**, který řídí spolupráci ostatních přístrojů. Je zajištěna vysoká rychlost a spolehlivost přenosu, při každém vyslaném **řídícím slově** (viz odst.) potvrzuje přijímači funkční jednotka její příjem. Tím se zároveň umožňuje spolupráce nesterjně rychlých jednotek.

Využití počítače PC k automatizaci měření

Jak už jsme poznamenali, centrální jednotkou měřicí soustavy je **počítač**. Norma IEEE-488 ho nazývá **řidič (controller)**. Nejvhodnějším počítačem k automatizaci měření je v současné době **počítač řady PC**, který je cenově nejvýhodnější. Při automatizaci měření není kladen velký význam na rychlost procesoru, grafiku a kvalitu ostatních sběrnic, proto pokud víme, že počítač bude sloužit jenom k měření, zcela postačí nejlevnější z této řady. K tomu, aby bylo možné pomocí počítače měřit, musíme ho vybavit vhodnou sběrnici, nejlépe GPIB. V praxi to znamená koupit "kartu" do počítače a nasadit ji do volné pozice (slotu) uvnitř počítače. Přestože pro hardwarovou (technickou) jednotnost sběrnice GPIB je přísně dodržována norma, aby bylo možno ke sběrnici různých výrobců připojit libovolný přístroj, který tuto normu splňuje, v programové podpoře těchto sběrnic neexistuje příliš velká jednotnost, i když určité zásady jsou respektovány. Abychom dosáhli programové uniformity alespoň v rámci určitého pracoviště (laboratoře), je vhodné ke každé zakoupené kartě GPIB od různých výrobců vytvořit **programovou jednotku** (viz příručky Python, Turbo Pascal, C++ a dalších programovacích jazyků), dejme jí např. název **driver488**, jejíž procedury se budou u všech zakoupených sběrnic volat stejným jménem a se stejnými parametry. Pak při dalším programování našeho měření zohledníme typ karty GPIB na začátku programu při deklaraci použitých programových jednotek a dále již nemusíme registrovat, se kterou konkrétní kartou budeme měření provádět.

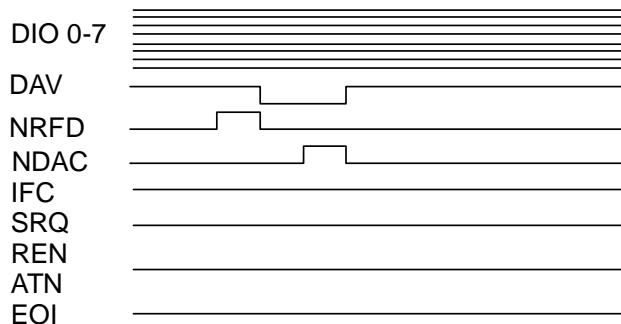
Komunikace s přístroji přes sběrnici GPIB



obr. 10 Propojení počítače (řidiče) s přístroji (posluchači/mluvčími)

nebo i řidič). Některé méně vybavené přístroje mohou být jen mluvčí (voltmetr předá změřený údaj do paměti počítače), nebo jen posluchač (zdroj je po přijetí instrukce z počítače nastaven na požadované napětí). Přístroje jsou rozlišovány pomocí **adres**, tj. různých čísel, přístroj se voláním příslušné adresy aktivuje.

Na obr. 10 je znázorněno propojení počítače (řidiče) s přístroji (posluchači/mluvčími) po třech **linkách: řízení, handshake a data**.



obr. 11 Šestnáct vodičů sběrnice GPIB umožňující přenos oběma směry

zajišťuje obousměrný přenos dat (tj. číselných hodnot nebo příkazů).

- **Linka řízení** (GIMB) je **pětivodičová** a má za úkol předávat speciální jednobitové příkazy, kterými se sběrnice GPIB řídí.

- **Linka handshake** (DBTCB), která je určena k řízení toku dat je **tvořena třemi vodiči**.

- **Linka data** je **osmivodičová** a

Datové vodiče – 8 vodičů (**DI0 1 až DI0 8**) slouží pro přenos dat a příkazů mezi přístroji po bytech.

Vodiče pro handshake (řízení přenosu/potvrzení přenosu) – 3 vodiče zajišťující asynchronní přenos na datové sběrnici:

DAV – Data Valid = potvrzení dat na sběrnici

NRFD – Not Ready for data = zařízení není připraveno přijmout data

NDAC – Not data accepted = data nepřijata

Vodiče pro řízení (řízení sběrnice) – 5 vodičů pro všeobecné řízení a koordinaci procesů na sběrnici:

ATN – Attention

IFC – Interface clear

REN – Remote enable

SRQ – Service request

EOI – End or identify

Příkazy (kódová slova) GPIB, sjednocená pro všechny GPIB od různých výrobců

Celkem existuje 40 příkazů GPIB, zde jen 7 základních:

příkaz	DIO							D	N	N	A	E	S	I	R	
	7	6	5	4	3	2	1	0								
UCG	X	0	0	1	X	X	X	X	X	X	1	X	X	X	X	
MLA	X	0	1	L	L	L	L	L	X	X	1	X	X	X	X	
UNL	X	0	1	1	1	1	1	1	X	X	1	X	X	X	X	
MTA	X	1	0	T	T	T	T	T	X	X	1	X	X	X	X	
UNT	X	1	0	1	1	1	1	1	X	X	1	X	X	X	X	
SCG	X	1	1	X	X	X	X	X	X	X	1	X	X	X	X	
END	X	X	X	X	X	X	X	X	X	X	0	1	X	X	X	

obr. 12 Stav vodičů GPIB pro kódová slova 7 základních instrukcí GPIB

L adresa přijímače (dvojkové číslo pro aktivované zařízení)

T adresa vysílače (dvojkové číslo pro aktivované zařízení)

X kde je označené, nesmí se během vysílání příkazu měnit log. úroveň signálu, jinak není změna zohledněna

0 logická nula

1 logická jednička

UCG (Universal Command) universální příkaz na nastavení spec. funkcí měř. přístrojů; na vodiči ATN povinně log. jednička (L)

MLA (My Listen Adress) aktivace přijímače, nastavuje funkci přijímače na vybraném zařízení; na vodiči ATN povinně (L)

MTA (My Talk Adress) aktivace vysílače, nastavuje funkci vysílače na vybr. zařízení; na vodiči ATN povinně (L)

UNL (Unlisten) neposlouchej, ruší funkci přijímače na všech zařízeních; na vodiči ATN povinně (L)

UNT (Untalk) nevysílá, ruší funkci vysílače na všech zařízeních,; na vodiči ATN povinně (L)

SCG (Secondary Command Group)

CANON		AMPHENOL
1	— dio1	— 1
2	— dio2	— 2
3	— dio3	— 3
4	— dio4	— 4
5	— ren	— 17
6	— eio	— 5
7	— dav	— 6
8	— nrfd	— 7
9	— ndac	— 8
10	— ifc	— 9
11	— srq	— 10
12	— atn	— 11
13	— stin	— [12]
14	— dio5	— 13
15	— dio6	— 14
16	— dio7	— 15
17	— dio8	— 16
18	— log.gnd.	— 12
19	— zem(6)	— 24
20	— zem(7)	— 18
21	— zem(8)	— 19
22	— zem(9)	— 20
23	— log.gnd.	— 21
24	— zem(11)	— 22
25	— zem(12)	— 23

obr. 13 Schéma propojení GPIB mezi konektory CANON a AMPHENOL

Propojovací kabely a konektory GPIB

Nejednotností jsou dva typy propojovacích konektorů k propojení GPIB karty s měřicími přístroji, případně přístrojů mezi sebou navzájem. Je to buď

1. v USA používaný konektor **AMPHENOL** s 24 vývody s vysunutými jazýčkovými spoji (zástrčka) a zapuštěnými jazýčkovými spoji (protikus-zásuvka), nebo
2. v Evropě rozšířený **CANON** s 25 vývody, kolíkový (zástrčka) a dutinkový (protikus-zásuvka).

Oba typy konektorů existují i průchozí, t.j. takové, kdy na jednom konektorovém tělese jsou na protilehlých stranách osazeny zásuvka i zástrčka. Protože se většinou nevyhneme tomu, aby alespoň jeden přístroj nebo karta počítače nebyly s odlišným konektorem, je vhodné mít k dispozici propojovací kabel nebo redukci, u nichž je na jednom konci konektor AMPHENOL a na opačném konci konektor CANON. Schéma propojení ukazuje obr. 13.

Řídící slovo, které určuje výrobce přístroje

Z praktického hlediska nás bude nejvíce zajímat linka **data**. Pomocí ní budeme na přístroj přenášet řídicí instrukci, které budeme říkat **řídící slovo** a z přístroje budeme pomocí této linky přenášet změřený údaj do paměti počítače. Řídící slovo je skupina (řetězec) ASCII znaků, která po přenesení přes styk GPIB zajistí vykonání požadované operace, t.j. nastavení přístroje nebo přijetí informace z přístroje. Tento řetězec se začleňuje do instrukce daného programovacího jazyka (např. C++, Python), která kromě jmenovaného řetězce musí obsahovat i adresu periferie. Skladba řídicího slova je určena výrobcem přístroje a není zavedena žádná jednotná tvorba řídicích slov.

V dalším výkladu předpokládáme využití programovacího jazyka Python. Na začátku se definuje přístroj zapojeny do naší aparatury

```
pristroj=instrument("GPIB0::XX::INSTR")
```

kde proměnná *XX* je maximálně dvojciferná adresa přístroje, zbývající klíčová slova jsou předepsána výrobcem přístroje, zde jsme použili definici přístroje **pro přístroj z produkce Agilent**.

Pokud si správně připravíme procedury a programové jednotky (u nás se např. bude jmenovat **visa**), bude řídicí slovo odesláno do přístroje pomocí instrukce

pristroj.write(retezec)

proměnná *retezec* obsahuje řídicí slovo. Řídicí slova mohou být odesílána jednotlivě, v tom případě se provedou po přijetí instrukce *pristroj.write*. Řídicí slova však můžeme odesílat i ve sdružené skupině. Pak seřadíme více řídicích slov do jednoho řetězce, přičemž před každé další řídicí slovo předřadíme oddělovací znak, který předepisuje výrobce přístroje (u některých přístrojů je nepovinný). Potom můžeme zadat pomocí jednoho příkazu *pristroj.write* i více příkazů přístroji najednou. Budou vykonávány ve stejném pořadí, v jakém jsou do výsledného řídicího řetězce seřazeny.

Je třeba, aby programová jednotka visa obsahovala proceduru startu měření, kdy se údaj na vstupních svorkách přístroje zaznamená do paměti přístroje, kde bude čekat na přenos do paměti počítače. Pro start měření použijeme instrukci

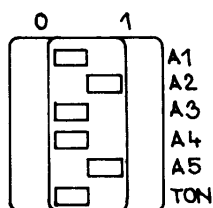
pristroj.read(retezec),

kde proměnná *pristroj* je definovaný výše a změřený údaj bude zapsán do paměti počítače do proměnné *retezec* (typu string).

Adresa přístroje

Před připojením přístroje k počítači je třeba, buď pomocí mikropínače, který se nachází na zadní straně přístroje, nebo pomocí tlačítek na přístroji, nastavit jeho GPIB adresu. Pokud nastavujeme mikropínačem, mikropínač sestává z několika prepínačů A_1 až A_n s polohou 0 nebo 1. Číselná hodnota nastavená na mikropínači se čte dvojkově, tj. podle pravidla,

$$a = a_0 + a_1 \cdot 2 + a_2 \cdot 2^2 + a_3 \cdot 2^3 + a_4 \cdot 2^4 + \dots + a_n \cdot 2^{n-1}, \quad (4)$$



kde $a_i = 2$ jeli nastaven prepínač A_i polohy 1, v opačném případě $a_i = 0$. Například na obr. 14 je nastaveno $A_1 = 0, A_2 = 1, A_3 = 0, A_4 = 0, A_5 = 1$, po dosazení do rovnice (4) vyjde $a = 2^1 + 2^4 = 18$. Mikropínač obsahuje někdy prepínač TON, který musíme při manuálním ovládání přístroje přepnout do polohy 0, v režimu GPIB může být 0 nebo 1.

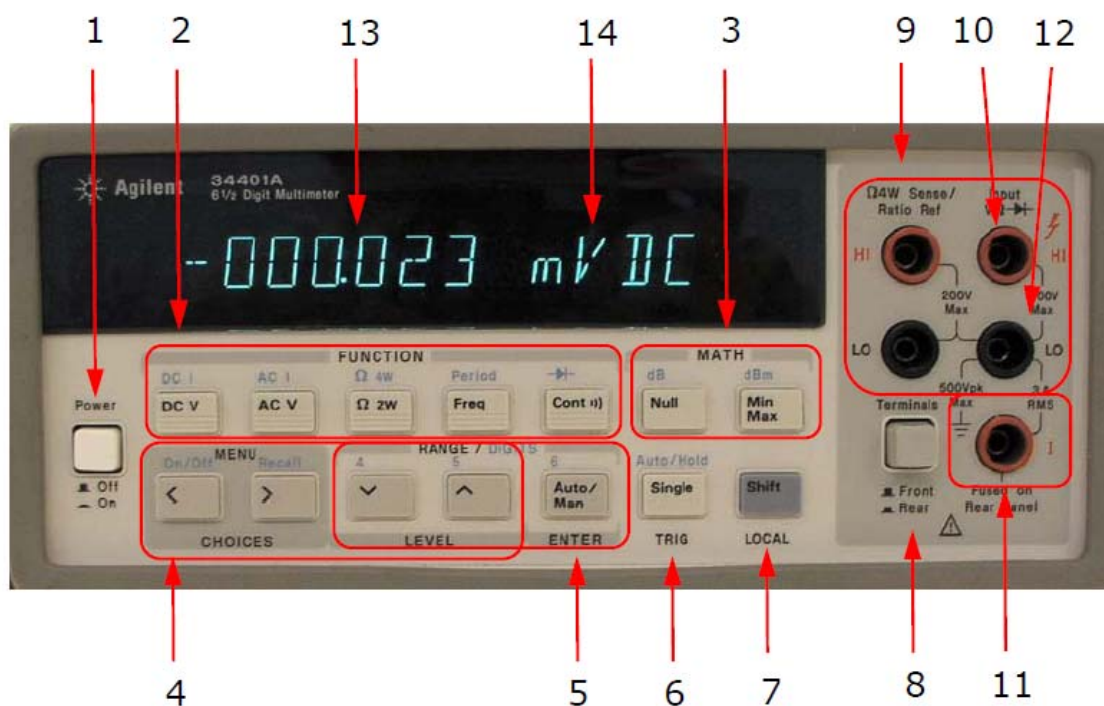
obr. 14 Mikropínač k nastavování adresy přístroje

Popis přístrojů se sběrnici GPIB od firmy Agilent

Multimetr Agilent 34401A

Popis

Přístroj 34401A je číslicový multimetr umožňující měření stejnosměrného a střídavého napětí a proudu, odporu dvou vodičovou i čtyřvodičovou metodou, měření periody a frekvence. Dále je možno kontrolovat polaritu a funkci PN přechodů a zjišťovat vodivá spojení v obvodech s indikací akustickým signálem. Přístroj je vybaven displejem s rozlišením $6\frac{1}{2}$ digitů, měřicí rozsah je nastavován automaticky nebo manuálně. Pro zobrazení výsledků je možno využít matematických funkcí – zobrazení v dB, dBm, průměrná, minimální, maximální hodnota. Přístroj je vybaven sériovým rozhraním RS-232 a rozhraním GPIB pro připojení do měřicího systému nebo ovládání z počítače.



1. Síťový vypínač
2. Tlačítka volby měřené veličiny
3. Tlačítka matematických funkcí pro volbu zobrazení měřené veličiny
1. (dB, dBm, Min, Max)
4. Tlačítka pro volbu a pohyb v menu
5. Tlačítka pro nastavení rozlišení displeje a volbu rozsahu
6. Tlačítko spuštění
7. Tlačítko pro volbu druhé funkce tlačítka
8. Tlačítko pro volbu předních/zadních měřicích svorek
9. Přední měřicí svorky – čtyřvodičové měření odporu
10. Svorka pro měření napětí, odporu, vodivého spojení, PN přechodu
11. Svorka pro měření proudu
12. Zemnicí svorka
13. Hodnota měřené veličiny

Funkce

Měřená veličina se volí tlačítky Function (2), modře vyznačené veličiny se vybírají pomocí klávesy Shift. Měření napětí připojte na svorky Input V Ω Hi a Lo umístěné zcela vpravo; pozor na záměnu se svorkami Hi a Lo určenými pro čtyřvodičové měření odporu (Ω 4W Sense). Měřená hodnota je zobrazena na displeji včetně jednotky. Desetinná místa jsou oddělena tečkou, zatímco zobrazené čárky pouze oddělují trojice čísel kvůli lepší čitelnosti.

Technické údaje

Měření stejnosměrného napětí

vstupní odpor:

rozsah 0,1 V, 1 V, 10 V volitelně 10 M Ω nebo > 10 G Ω

rozsah 100 V, 1000 V 10 M Ω \pm 1%

Měření stejnosměrného proudu

snímací rezistor:

rozsahy 10 mA, 100 mA 5 Ω

rozsah 1 A, 3 A 0,1 Ω

Testování vodivého spojení

citlivost: nastavitelná od 1 Ω do 100 Ω

Měření střídavého napětí

měřicí metoda: měření TRMS se střídavou vazbou

vstupní impedance: 1 M Ω \pm 2% + 100 pF

Měření střídavého proudu

měřicí metoda: měření TRMS se střídavou vazbou

snímací rezistor:

rozsah 1 A, 3 A 0,1 Ω

Měření kmitočtu

rozsah: 3 Hz – 300 kHz

Řídící slova multimetru Agilent 34401A

Obecný tvar měřicího nebo nastavovacího řetězce je

PŘÍKAZ:PARAMETRY?

Koncový otazník se používá **pouze u měření nebo získávání jiných informací**, ne u nastavování, konfigurace a pod., kde zpětnou informaci nedostáváme.

Seznam vybraných příkazů:

*CLS	nulování všech stavových registrů
MEAS	změřit
CONF	konfigurovat
SENS	volba

Seznam vybraných parametrů:

VOLT : DC	stejnoseměrné napětí
VOLT : AC	střídavé napětí
CURR : DC	stejnoseměrný proud
CURR : AC	střídavý proud
RES	elektrický odpor
FREQ	frekvence

PER perioda

Příklady

MEAS:VOLT:DC?	Změř stejnosměrné napětí, automatický rozsah.
MEAS:CURR:DC?	Změř stejnosměrný proud, automatický rozsah.
MEAS:RES?	Změř elektrický odpor, dvou-svorkově, automatický rozsah
CONF:FREQ 1000,0.1	Nastav funkci měření frekvence, rozsah 1kHz a rozlišení 0.1Hz

Formát zobrazení na displeji přístroje

-H.DDD,DDD_EFFF

Kde:

_ je mezera (aby jste ji lépe viděli)

- se v případě kladné hodnoty nezobrazí

H je číslice v rozsahu 0 až 1

. povinný znak v povinné poloze

, povinný znak v povinné poloze

D je číslice v rozsahu 0 až 9

E je povinný znak v povinné poloze

F je znak jednotky, může být zepředu doplněn mezerami

Ukázky formátu zobrazení na displeji přístroje

10.216,5 VDC

Ukázka pro rozsah 10 V stejnosměrné napětí

-045.23 mVDC

Ukázka pro rozsah 100 mV stejnosměrné napětí

113.325,6 OHM

Ukázka pro měření odporu rozsah 100 ohm

OVL.D mVDC

Ukázka pro rozsah 100mV stejnosměrné napětí, měřená hodnota přetekla

Formát zobrazení na displeji pro různá nastavení

po nastavení	prístroj na read() vrati
MEAS:CAP?	-H.DDD,DDD__F
MEAS:CURR:AC?	-H.DDD,DDD__AAC
MEAS:CURR[:DC]?	-H.DDD,DDD__AAC
MEAS:FREQ?	-H.DDD,DDD__HZ
MEAS:PER?	-H.DDD,DDD__ms
MEAS:TEMP?	-H.DDD,DDD__K
MEAS[:VOLT]:AC?	-H.DDD,DDD__VAC
MEAS[:VOLT][:DC]?	-H.DDD,DDD__VDC

Formát výstupu dat do Vašeho programu

Výstupní údaje budou v jednom z formátů uvedených v následující tabulce. Pro jednoduchost budeme používat jen první řádek (jedno čtení)

Typ výstupních dat	Výstupní formát dat
Jedno čtení (IEEE-488)	SD.DDDDDDDDESDD<nl>
Vícenásobné měření (IEEE-488)	SD.DDDDDDDDESDD, ..., ..., <nl>

S záporné nebo kladné znaménko

D číslice

E exponent

<nl> znak nového řádku

Číslicově řízený zdroj Agilent E3631A

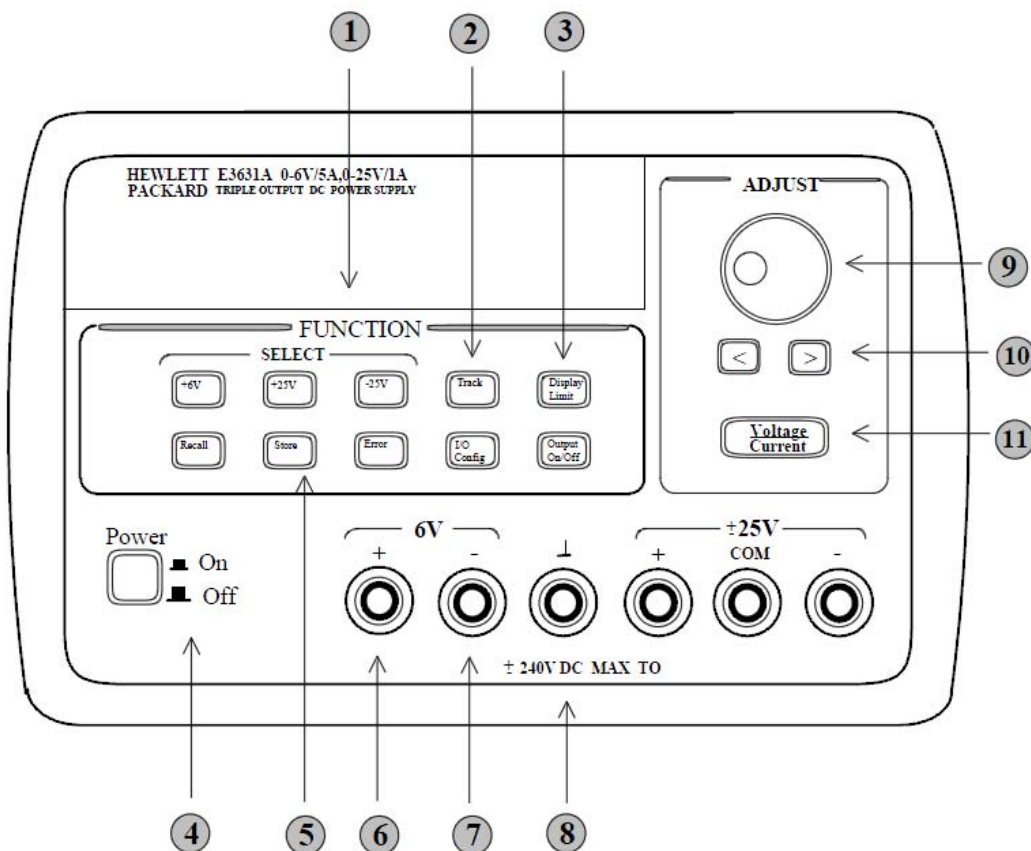
Popis

Základní provedení stejnosměrných zdrojů Agilent poskytuje nejdůležitější napájecí vlastnosti pro finančně nenáročné aplikace. Trojitý zdroj E3631A, 80W má tři výstupy, GPIB, má malé a kompaktní provedení určené pro práci na pracovním stole, má malé výstupní zvlnění a malý šum, obsahují vnitřní měření a umožňují základní programovatelné funkce. Tento dokonalý a spolehlivý zdroj je určen pro všeobecné aplikace a nabízí pohodlnou práci na pracovním stole jako kompaktní a flexibilní napájecí systém s více výstupy.



Funkce

Přední panel číslcového zdroje HP E3631A



1. Klávesy pro výběr zdroje. Po výběru zdroje je možné nastavit napětí nebo proud pomocí otočného knoflíku 9 a kláves se šípkami 10
2. Klávesa pro zapnutí nebo vypnutí "Track modu". Jedná se o režim ve kterém jsou +25V a -25V zdroj v "závěsu". Nastavením parametrů jednoho zdroje se automaticky nastaví stejné parametry u zdroje druhého. Pro aktivaci funkce je potřeba držet klávesu "Track" stisknutou minimálně 1 sekundu. "Track mod je výhodný zejména v případě, kdy zdroj využíváme jako symetrický 0 až ±25V.
3. Stisknutím klávesy Display Limit se na displeji zobrazí nastavené limitní hodnoty napětí a proudu. Pomocí otočného knoflíku 9 a kláves se šípkami je možné tyto hodnoty upravit. Opětovným stisknutím klávesy se stav displeje vrátí do předchozího stavu. Pokud je zdroj ve stavu CV (jako zdroj konstantního napětí), pak nastavováním limitní hodnoty napětí se nastavuje zároveň hodnota výstupního napětí, neboť tyto hodnoty si odpovídají. Podobně pokud je zdroj ve stavu CC (jako zdroj konstantního proudu) nastavujeme limitní hodnotou proudu zároveň hodnotu výstupního proudu.
4. Klávesa Recall slouží pro vyvolání nastavení parametrů zdroje uložených v paměti. Paměti jsou tři a výběr se provádí pomocí otočného knoflíku 9. Opětovným stisknutím klávesy Recall dojde k novému nastavení zdroje.
5. Klávesa Store/Local slouží k uložení nastavených parametrů zdroje do paměti 1 až 3 (výběr se provede pomocí otočného knoflíku 9). K uložení dojde opětovným stisknutím klávesy

Store / Local. Dále lze klávesy použít při přechodu z dálkového do místního ovládání.

6. Klávesa Error/Calibrate. Slouží jednak ke zobrazení kódu chyb vzniklých během provozu, automatického testu nebo kalibrace a jednak k nastavení kalibračního módu. K tomu, aby se zdroj dostal do stavu kalibrace je nutné současně při zapnutí zdroje držet stisknutou klávesu Error/Calibrate po dobu zhruba 5 sekund.
7. Klávesa I/O Confing/Secure - Tato klávesa slouží jednak pro nastavení konfigurace (sběrnice HP - IB nebo RS 232, adresa přístroje, přenosová rychlost a parita a jednak pro odjištění nebo zajištění kalibračních parametrů.
8. Klávesa Output On/Off- odpojení nebo připojení nastaveného výstupního napětí na výstupní svorky zdroje.
9. Otočný knoflík - slouží ke zvyšování či snižování hodnoty blikajícího digitu nebo k výběru z nabídky.
10. Klávesy se šípkami - slouží k přesunutí blikajícího digitu doprava či doleva
11. Klávesa Voltage/Current - klávesa pro výběr nastavované veličiny (napětí nebo proudu)

Technické údaje

- Výstupní parametry:
- výstup 1: 0 až 6 V ss., 0 až 5 A
- výstup 2: 0 až +25 V ss., 0 až 1 A
- výstup 3: 0 až -25 V ss., 0 až 1 A
- Přesnost programování při 25°C ±5°C
- napětí: 0,05% + 10 mV, 0,05% + 20 mV, 0,1% + 5 mV
- proud: 0,15% + 4 mA, 0,15% + 4 mA, 0,2% + 10 mA
- Zvlnění a šum: (20 Hz až 20 MHz)
- napětí (normal mode-mode / mezi sebou): <350 μVef/2 mVš-š, <350 μV ef/2 mV š-š, <350 μVef/2 mV š-š
- proud (normal mode-mode / mezi sebou): <500 μA ef, <500 μA ef, <2 mA ef
- proud (common-mode / proti zemi): <1,5 μA ef, <1,5 μA ef, <1,5 μA ef
- Přesnost měření při 25°C ±5°C
- napětí: 0,05% + 10 mV, 0,05% + 10 mV, 0,1% + 5 mV
- proud: 0,15% + 4 mA, 0,15% + 4 mA, 0,2% + 10 mA

Řídicí slova zdroje Agilent E3631A

Označení zdrojů:

P6V	zdroj +6V
P25V	zdroj +25V, polarita + na aktivní svorce, – na společné svorce COM
N25V	zdroj -25V, polarita – na aktivní svorce, + na společné svorce COM

Obecný tvar měřícího nebo nastavovacího řetězce je

PŘÍKAZ:PARAMETRY?

Koncový otazník se používá **pouze u získávání zpětných informací**, ne u nastavování, konfigurace a pod., kde zpětnou informaci nedostáváme.

Seznam vybraných příkazů:

*CLS	nulování všech stavových registrů
APPL	výběr a nastavení napětí a proudu
INST	výběr zdroje na kterém nastavovat jednotlivé parametry
OUTP	odpojit nebo připojit zdroj k výstupním svorkám
COUP	spřáhnout zdroje

Seznam vybraných parametrů:

SEL P6V VOLT : AC	výběr zdroje s polaritou plus do +6V střídavého (místo P6V také P25V nebo N25V, kde N je polarita minus na aktivním vodiči)
ON	zapnout
OFF	vypnout

Příklady

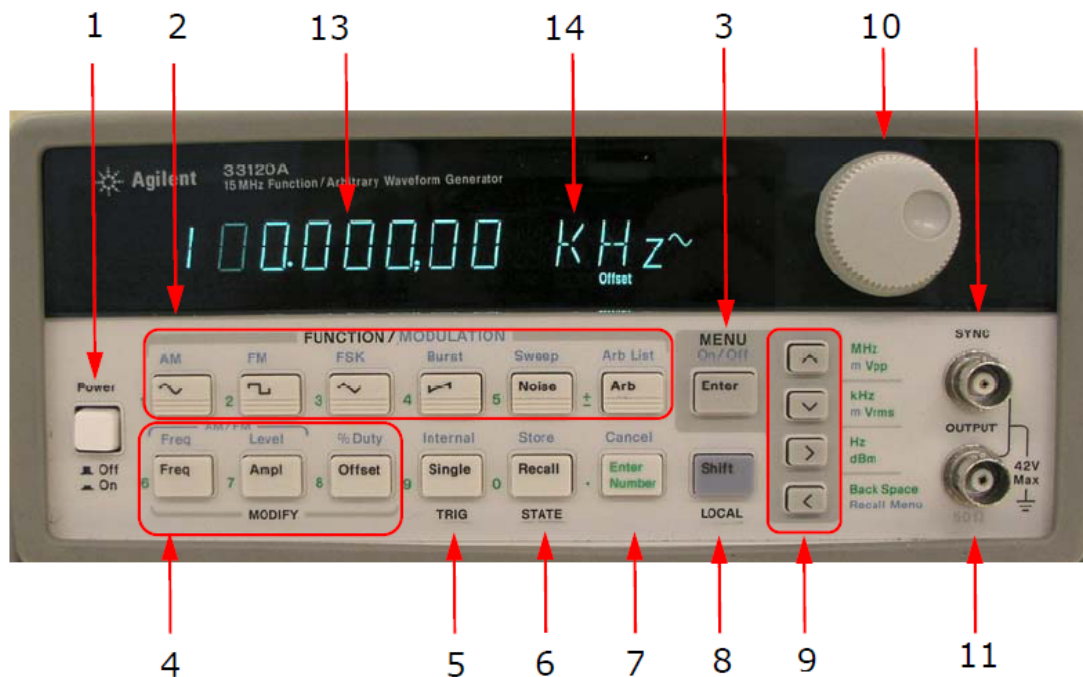
INST P25V	výběr zdroje s rozsahem 25V
VOLT 1.2	nastav napětí 1,2 V na vybraném zdroji (výběr viz INST)
OUTP ON	výstupní svorky jsou otevřeny (OUTP OFF - zavřeny)
COUP P6V, P25V	spřažení zdrojů 6V a +25V

Programovatelný funkční generátor Agilent 33120A

Popis

Agilent 33120A je funkční generátor pracující na principu přímé kmitočtové syntézy. Přístroj tak umožňuje generovat nejen základní průběhy (sinus, obdélník, pila, trojúhelník, šum), ale i libovolný průběh, který si může uživatel uložit do paměti přístroje. Nastavovat lze kmitočet, amplitudu, ofset a pro obdélníkový průběh i činitel plnění impulsů. Je možné generovat i spouštěné sekvence impulsů uvedených průběhů. Výstupní signál lze modulovat amplitudovou modulací (interně nebo externě), kmitočtovou modulací nebo kmitočtovým klíčováním (obdélníkový průběh). Kmitočet výstupního signálu lze lineárně nebo logaritmicky rozmítat do maximální hodnoty s dobou rozmítání až 500 sekund. Přístroj je vybaven sériovým rozhraním RS-232 a rozhraním GPIB, je tak možno ho zapojit do měřicího systému nebo ho ovládat z počítače.

Funkce



1. Síťový vypínač
2. Tlačítka pro volbu tvaru výstupního signálu / volbu modulační výstupního signálu
3. Tlačítka pro vstup a výstup z menu a potvrzení volby
4. Tlačítka pro volbu parametrů výstupního signálu / volbu parametrů modulační
5. Tlačítka spouštění
6. Tlačítka nastavení přístroje z paměti / uložení nastavení přístroje do paměti
7. Tlačítka pro zadání číselné hodnoty
8. Tlačítka pro volbu druhé funkce tlačítka
9. Tlačítka pro pohyb v menu a na displeji
10. Otočný prvek pro nastavování hodnot na displeji
11. Výstupní konektor generovaného signálu
12. Výstupní konektor signálu synchronizace

Volba tvaru výstupního signálu se provádí stiskem tlačítek v horní řadě Function (2). Důležitá jsou první dvě – harmonický a obdélníkový signál. U obdélníkového signálu je možné změnit činitel

plnění pomocí volby Shift a % Duty.

Kmitočet generovaného signálu se nastavuje po stisku tlačítka Freq, amplituda pak po stisku tlačítka Ampl. Pro nastavování parametru signálu lze použít směrová tlačítka (9) spolu s otočným prvkem (10), nebo lze hodnotu zadat číselnými klávesami (zelená barva) po stisknutí tlačítka Enter number (7). Ukončení vkládání čísla současně s volbou násobku jednotky se děje pomocí tlačítek (9), čímž se zároveň určí násobek jednotky \wedge (MHz, Vpp), \vee (kHz, Vrms) či $>$ (Hz, dBm). Pokud stisknete předem ještě Shift, je vložena namísto V hodnota mV.

Poznámka: Vpp je napětí špička-špička (tedy dvojnásobek amplitudy), Vrms je efektivní hodnota napětí.

Nastavovaná veličina je zobrazena na displeji včetně jednotky.

Desetinná místa jsou oddělena tečkou, zatímco zobrazené čárky pouze oddělují trojice čísel kvůli lepší čitelnosti.

Výstup generátoru je označen Output, pozor na záměnu se synchronizačním výstupem (Sync). Výstupní impedance je 50 Ω .

Technické údaje

Generované signály: sinus, obdélník, trojúhelník, pila, šum, stejnosměrné napětí, sinc, obrácená pila, exponenciální průběh klesající, exponenciální průběh rostoucí, EKG

Průběhy v uživatelské paměti:

délka 4 x 8 až 16 000 bodů (čtyři paměti)

rozlišení 12 bitů

vzorkovací kmitočet 40 MSa/s

Parametry generovaného signálu:

amplituda 50 mV – 10 V ($\pm 1\%$ na 1 kHz)

ofset ± 5 V $\pm 2\%$

kmitočet

- sinus 100 μ Hz – 15 MHz
- obdélník 100 μ Hz – 15 MHz
- trojúhelník 100 μ Hz – 100 kHz
- pila 100 μ Hz – 100 kHz
- šum (gaussovský) šířka pásma 10 MHz
- uživatelský průběh 100 μ Hz – 200 kHz až 5 MHz podle délky průběhu

Modulační parametry:

amplitudová modulace

- kmitočet mod. signálu 10 mHz – 20 kHz
- hloubka modulace 0 % - 120 %

kmitočtová modulace

- kmitočet mod. signálu 10 mHz – 10 kHz

Řídící slova generátoru Agilent 33120A

Obecný tvar měřicího nebo nastavovacího řetězce je

PŘÍKAZ:PARAMETRY?

Koncový otazník se používá **pouze u získávání zpětných informací**, ne u nastavování,

konfigurace a pod., kde zpětnou informaci nedostáváme.

Seznam vybraných příkazů:

*CLS	nulování všech stavových registrů
APPL	příkaz pro nastavení průběhu, frekvence, amplitudy a stejnosměrné složky
OUTP	práce s výstupem
FUNC	nastavení nebo dotaz
[SOUR]	příkaz pro nastavení, toto slovo může být v příkazu vypuštěno

Seznam vybraných parametrů:

SIN	sinusový průběh
SQU	čtvercový průběh
VOLT {cislo}	amplituda signálu ve voltech, viz příklad
LOAD {cislo}	nastavení použité zatěžovací impedance, viz příklad
ON	zapnout
OFF	vypnout

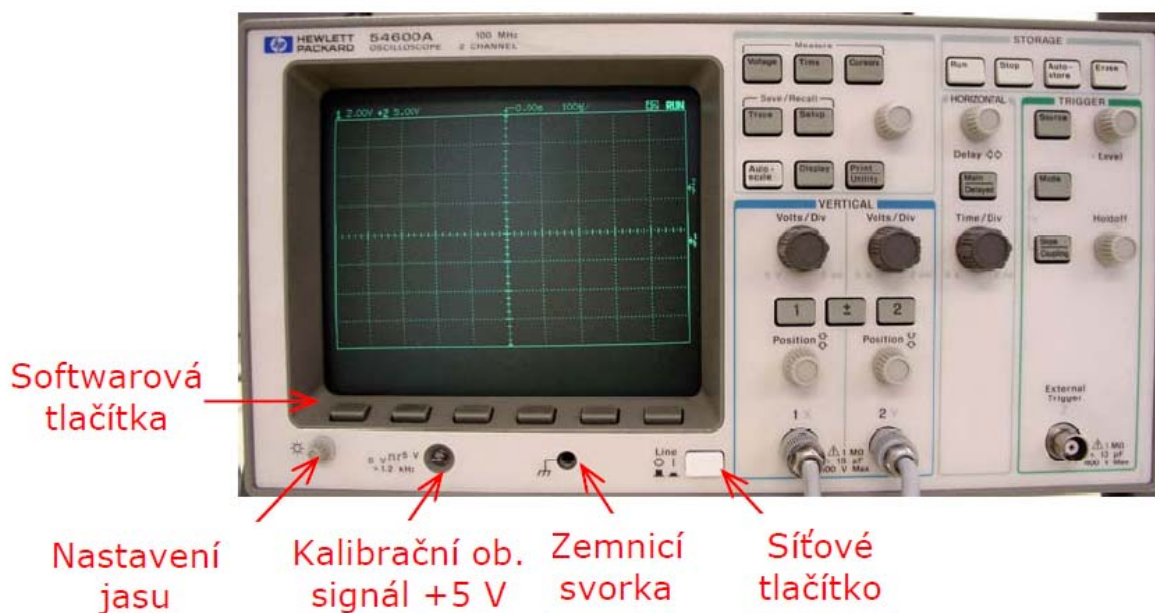
Příklady

FUNC SIN	nastaví na výstupu sinusový signál.
FREQ 440	nastaví frekvenci výstupního signálu 440Hz
VOLT 5	nastaví amplitudu výstupního signálu 5V
OUTP ON	výstupní svorky se otevřou (OUTP OFF - zavřou)
OUTP:LOAD 50	nastavení impedance na výstupu 50Ω
FUNC : SHAP?	dotaz na použitý průběh

Osciloskop Agilent 54600A

Popis

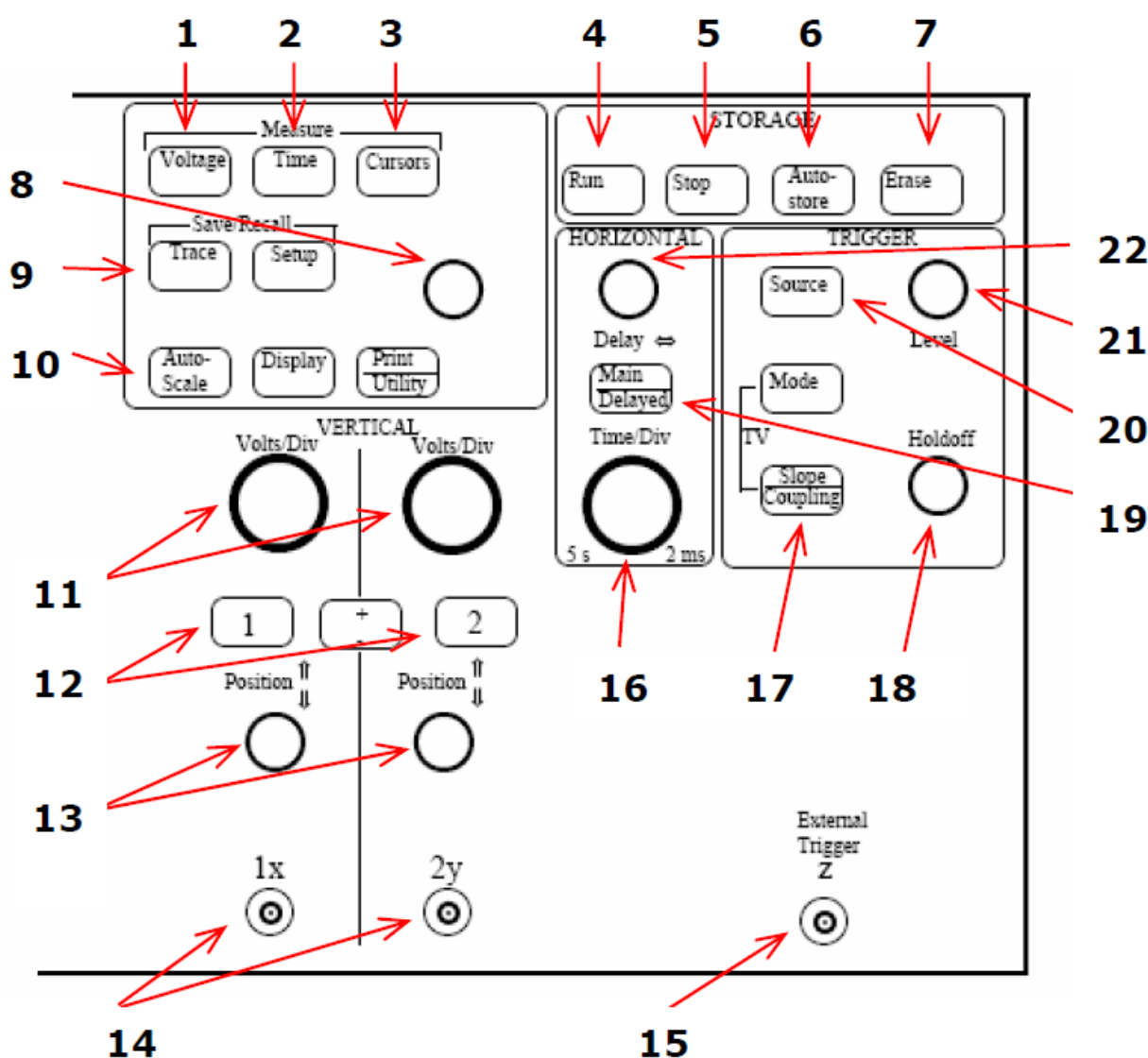
Jedná se o dvoukanálový osciloskop s číslicovou pamětí zobrazených průběhů a parametrů nastavení. Maximální vzorkovací kmitočet je 20 MHz a šířka kmitočtového pásma je 100 MHz. Osciloskop umožňuje kurzorové měření frekvence, času a napětí. Pomocí sběrnice RS 232 nebo GPIB (volitelné připojitelné moduly) lze použít dálkové řízení osciloskopu nebo připojit tiskárnu.



1. Měření napětí (špička–špička, ef. hodnota, střední hodnota, max. a min hodnota)
2. Měření času (frekvence, perioda, činitel plnění, šířka kladného a záporného impulsu, doba nástupné a sestupné hrany)
3. Zapnutí kurzoru
4. RUN - spuštění měření a uvolnění časové základny
5. STOP - zastavení časové základny a uložení obrazu do paměti
6. AUTOSTORE - obraz se průběžně ukládá
7. ERASE - vymazání paměti
8. Otočný regulátor pro posun kurzorů
9. Tlačítko TRACE vybírá nabídku funkcí pro ukládání obrazu do energeticky nezávislých paměti a tlačítko SETUP vybírá nabídku pro uložení parametrů nastavení osciloskopu
10. 10.AUTOSCALE - tlačítko pro automatické zobrazení průběhu
11. Vertikální změna měřítka (citlivost) pro jednotlivé kanály
12. Volba zobrazení kanálů
13. Vertikální posun zobrazovaných signálů na obrazovce
14. Vstupy kanálů 1 a 2, tlačítkem ± se umožní výběr matematických operací v menu
15. Vstup externího signálu pro synchronizaci
16. Časová základna
17. Odvození spouštěcího impulsu
18. HOLDOFF - doba zákazu spouštění časové základny v rozsahu 0,2 ms až 13.5 s
19. MAIN – DELAYED - nastavení dvojí časové základny (Zoom)
20. Výběr zdrojového signálu pro synchronizaci je tlačítkem SOURCE a tlačítkem MODE se vybírá režim spouštění

21. Otočný regulátor pro nastavení úrovně spouštění

22. Horizontální posun na obrazovce



Technické údaje

Vertikální systém

Šířka pásma (-3 dB): 0 až 100 MHz

Střídavá vazba: 10 Hz až 100 MHz

Kanály 1 a 2: 2 mV/dílek až 5 V/dílek $\pm 1,5\%$

Činitel potlačení souhlasného napětí: 20 dB při 50 MHz

Vstupní impedance: 1 M Ω , 13 pF

Max. vstupní napětí: 400 V (ss + špičková hodnota)

Horizontální systém

Časová základna: 5 s/dílek až 2 ns/dílek

Přesnost: $\pm 0,01\%$

Předstih spouštění ČZ: ≥ 10 dílků

Vnější spouštění

Rozsah: ± 18 V

Vstupní impedance: 1 M Ω , 13 pF

Max. vstupní napětí: 400 V (ss + špičková hodnota)

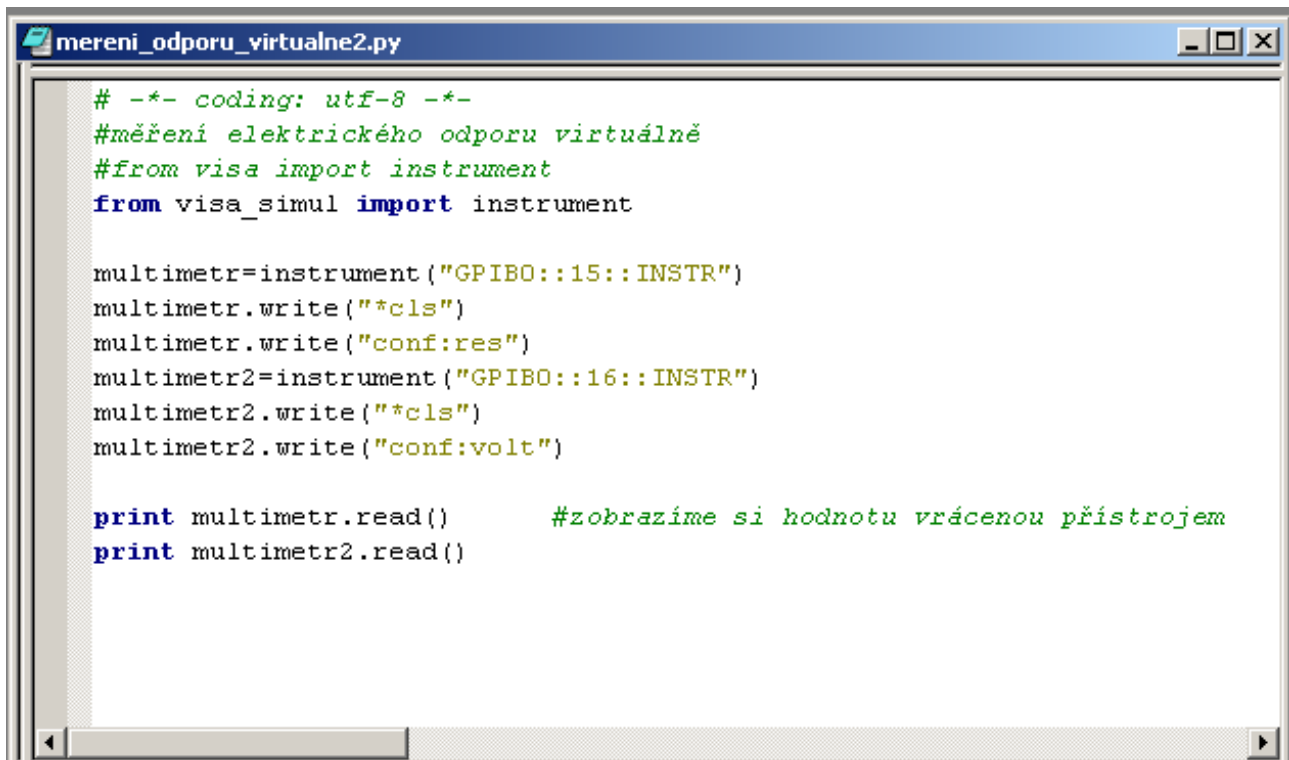
Vzorkovací systém

Maximální rychlost vzorkování: 20 M vzorků/s
Přesnost: 8 bitů

Program (skript) pro ovládání jednoduchého měřicího celku s možností simulace

Simulace měření

Protože ne vždy bude možnost tvořit aplikaci za použití měřících přístrojů, nahradíme si přístroje jednoduchou softwarovou funkcí, která nám bude posílat smyšlené (náhodné) údaje. Provedeme několik úprav v našem programu. Použijeme skript (program) z ostrého měření, kde upravíme zdrojový kód podle obr. 15a uložíme pod novým názvem.



```
# -*- coding: utf-8 -*-
#měření elektrického odporu virtuálně
#from visa import instrument
from visa_simul import instrument

multimetr=instrument("GPIBO::15::INSTR")
multimetr.write("*cls")
multimetr.write("conf:res")
multimetr2=instrument("GPIBO::16::INSTR")
multimetr2.write("*cls")
multimetr2.write("conf:volt")

print multimetr.read()      #zobrazíme si hodnotu vrácenou přístrojem
print multimetr2.read()
```

obr. 15 Zdrojový kód "mereni_odporu_virtualne2.py"

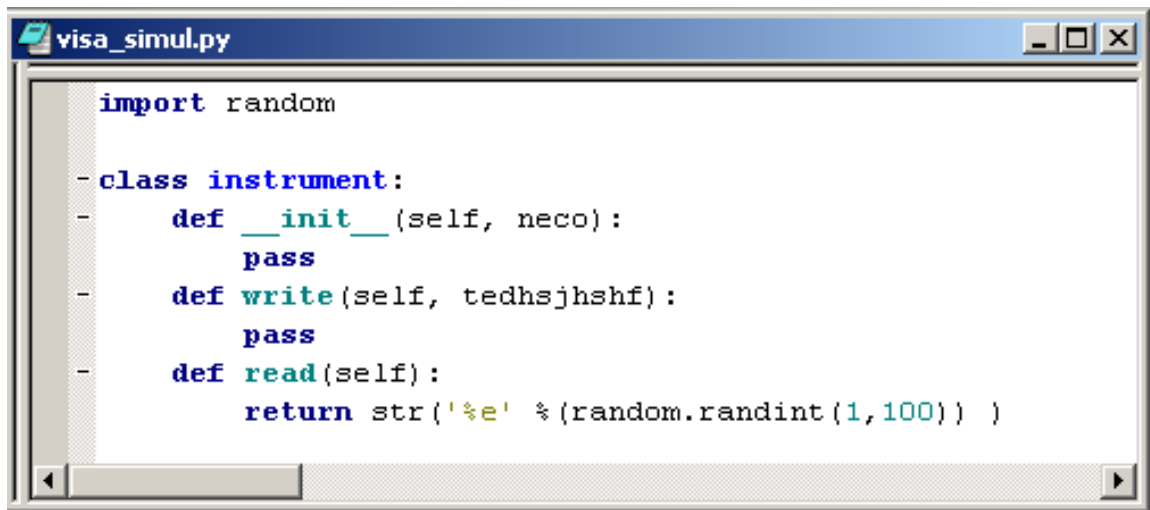
Změny oproti „ostrému“ měření jsou ty, že zakomentujeme

#from visa import instrument

A naopak použijeme (odkomentujeme) import třídy

from visa_simul import instrument.

Nyní nebudeme nic zavírat a otevřeme nový skript. Nazveme jej „visa_simul.py“ a vepíšeme do něj řádky podle . Uložíme.

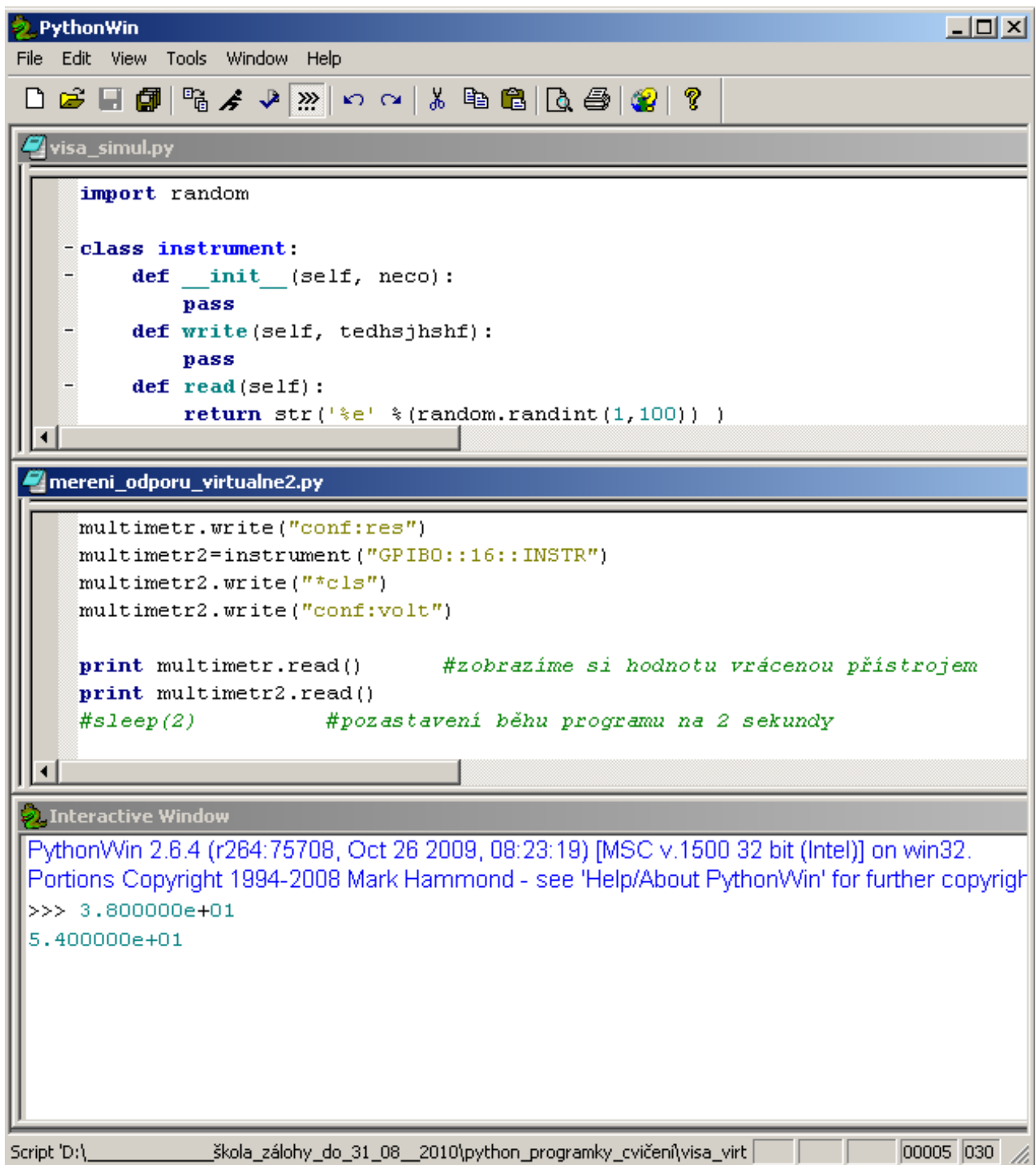


```
import random

class instrument:
    def __init__(self, neco):
        pass
    def write(self, tedhsjhshf):
        pass
    def read(self):
        return str('%e' % (random.randint(1,100)) )
```

obr. 16 Zdrojový kód modulu "visa_simul.py"

Po zarovnání oken bude vypadat PythonWin okno třeba jako na obr. 16.



obr. 17 PythonWin – třída, skript a interaktivní okno

Nyní se přesuneme zpět do okna skriptu “mereni_odporu_virtualne2.py” a spustíme náš program. Pokud se program vykonal úspěšně, uvidíme v Interactive Window jako poslední záznam dvě náhodná přirozená čísla od 1 do 100.

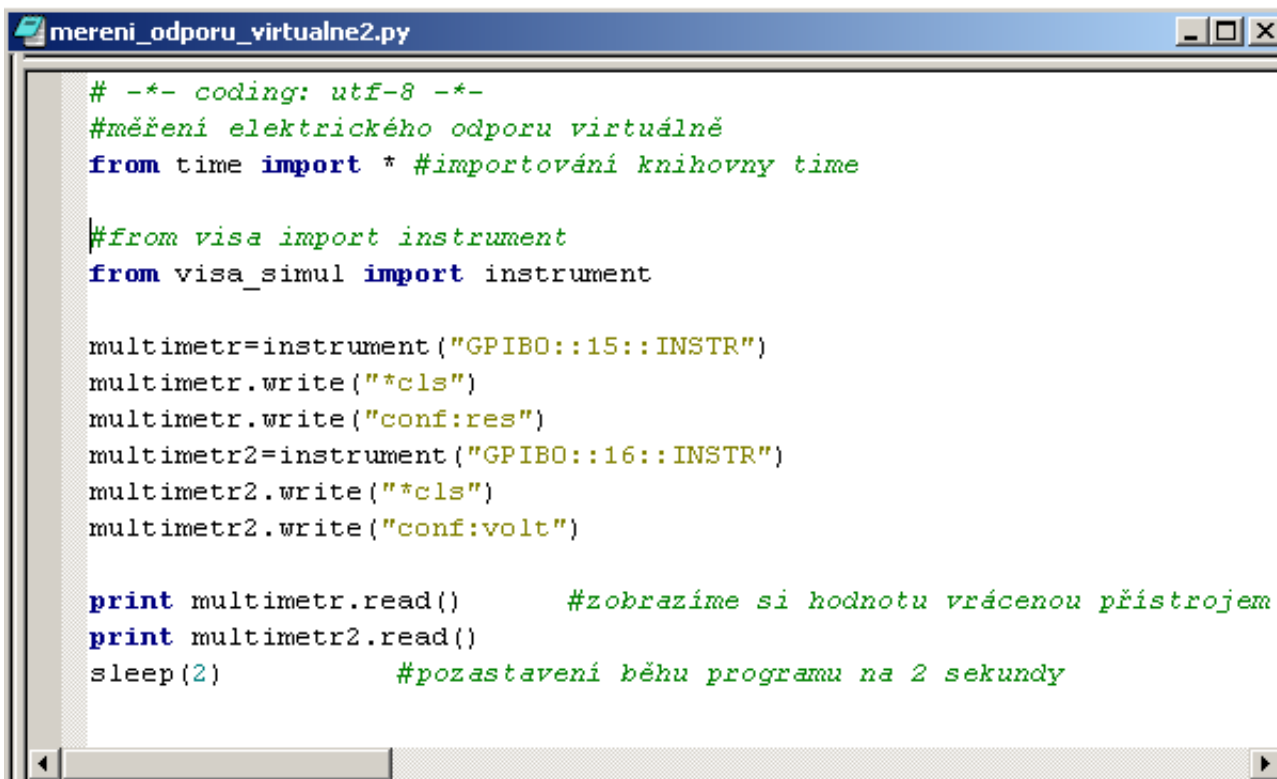
Navíc nám v místě uložení našeho programu přibyl soubor s názvem „visa_simul.pyc“, který bude třeba v následujícím odstavci.

Vytvoření „exe“ souboru z „pythonovského“ programu

Předpokládáme, že máme funkční předešlý program s názvem „mereni_odporu_virtualne2.py“. Nyní si ukážeme, jak transformovat náš program na program „mereni_odporu_virtualne2.exe“,

který si bude moct spustit i uživatel, který nemá na svém počítači nainstalovaný Python. Použijeme k tomu už nainstalované **py2exe**.

Nejprve doplníme program „mereni_odporu_virtualne2.py“ o dva řádky (v obr. 18 jsou to třetí a poslední řádek), otestujeme a pokud pracuje správně, uložíme.



```
# -*- coding: utf-8 -*-
#měření elektrického odporu virtuálně
from time import * #importování knihovny time

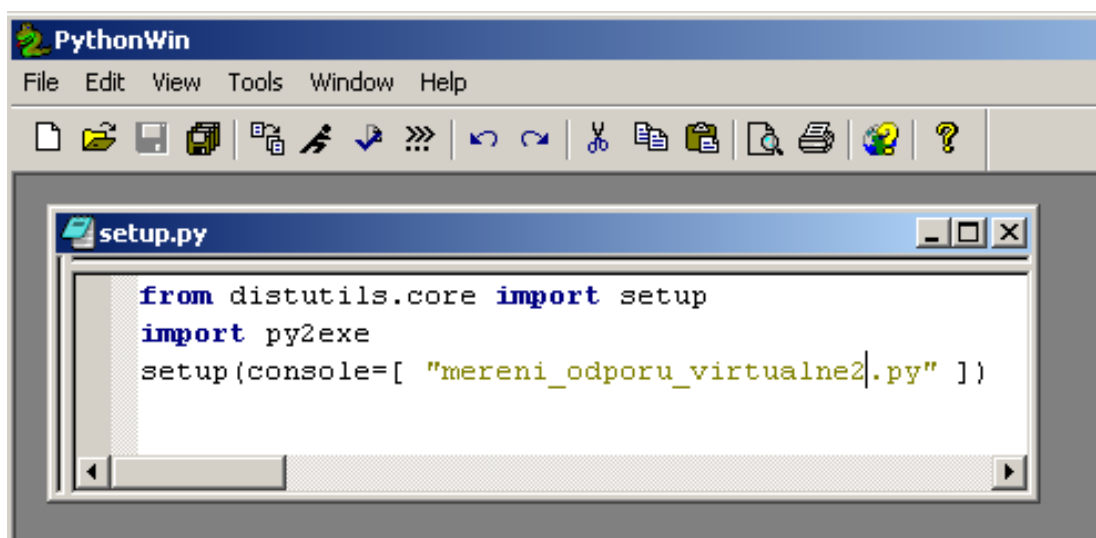
#from visa import instrument
from visa_simul import instrument

multimetr=instrument("GPIBO::15::INSTR")
multimetr.write("*cls")
multimetr.write("conf:res")
multimetr2=instrument("GPIBO::16::INSTR")
multimetr2.write("*cls")
multimetr2.write("conf:volt")

print multimetr.read()      #zobrazíme si hodnotu vrácenou přístrojem
print multimetr2.read()
sleep(2)                    #pozastavení běhu programu na 2 sekundy
```

obr. 18 Doplnění hlavního souboru

Zkopírujeme hotový program **mereni_odporu_virtualne2.py** do instalačního adresáře Python, např. C:\Python26\. Na stejné místo zkopírujeme soubor **visa_simul.pyc**. V prostředí **PythonWin** zavřeme okna skriptů a otevřeme zcela nový skript, do kterého vepíšeme řádky podle obr. 18.

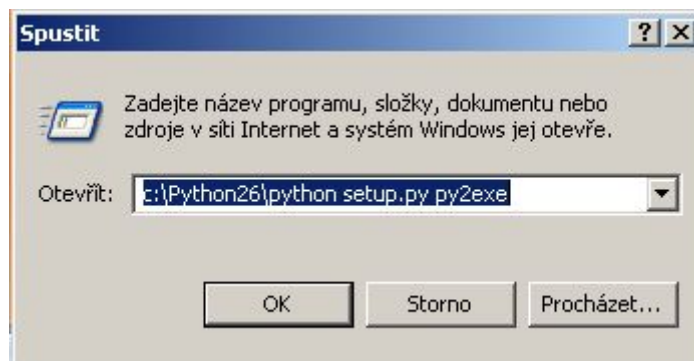


```
from distutils.core import setup
import py2exe
setup(console=[ "mereni_odporu_virtualne2.py" ])
```

obr. 19 Obsah setup.py souboru

Skript uložíme pod názvem **setup.py** také do C:\Python26\. Je důležité správně vyplnit název vašeho souboru uvnitř uvozovek. PythonWin můžeme zavřít.

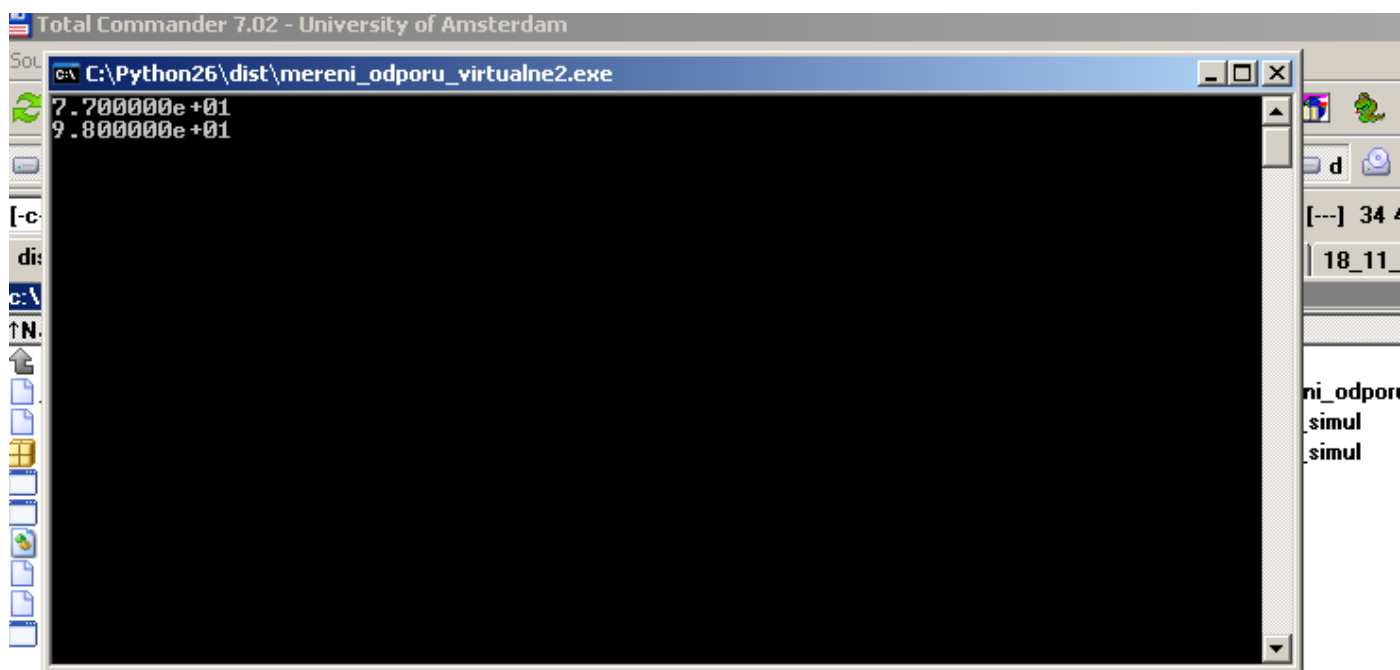
V prostředí Microsoft Windows klikneme na Start-spustit (nebo použijeme příkazovou řádku v Total Commanderu apod.) a do řádku vepíšeme text `c:\Python26\python setup.py py2exe`. Viz obr. 20.



obr. 20 Spuštění tvorby exe souboru

Stlačíme klávesu Enter nebo klikneme na OK. Pokud je všechno v pořádku, začne se nám v černém konzolovém okně velmi rychle vypisovat různý text, informující o probíhající tvorbě **exe** souboru. Po skončení uvidíme v instalačním adresáři `c:\Python26\` dvě nové složky **build** a **dist**. Uvnitř složky **dist** nalezneme vygenerovány soubory, mezi kterými je očekávaný soubor **mereni_odporu_virtualne2.exe**.

Pokud se rozhodneme přenést spustitelný soubor exe na jiný počítač, je nutné přenést celou složku **dist**.



obr. 21 Konzolové okno

Ukládání dat do souboru a cyklus

Už umíme vytvořit samostatný funkční program, nyní jej obohatíme o ukládání naměřených dat do textového souboru. A abychom mohli uložit efektivně víc než jednu sadu dat, seznámíme se s cyklem, přesněji jedním z možných cyklů.

Otevřeme nový skript nebo upravíme některý dřívější podle programu na obr. 22. Samozřejmostí by mělo být uložení nového programu pod jiným názvem, než byl předchozí.


```
mereni_odporu_virtualne3.py
# -*- coding: utf-8 -*-
#měření elektrického odporu virtuálně
from time import * #importování knihovny time

#from visa import instrument
from visa_simul import instrument

multimetr=instrument("GPIB0::15::INSTR")
multimetr.write("*cls")
multimetr.write("conf:res")

soubor=file("textovysoubor.txt",'w') #vytvoření souboru textovysoubor.txt
#soubor=open("textovysoubor.txt",'w')

soubor.write("namereno\n") #uložení textu do souboru

x=0 #nastavení proměnné x na nulu
while x<10: #cyklus se provádí, pokud je x menší než 10
    a=multimetr.read() #do lokální proměnné se vloží údaj z přístroje
    soubor.write(a+"\n") # do souboru se vloží řetězec složený z obsahu
                        # proměnné "a" a připojíme znaky pro skok na
                        # další řádek
    print a #na obrazovce uvidíme každou hodnotu
    sleep(1) #zastavíme chod programu na 1 sekundu
    x+=1 #zvýšíme hodnotu v proměnné x o jedna

soubor.close() #uzavření souboru
```

obr. 22 Program zahrnující ukládání do souboru a cyklus

Na 9. Řádku programu (prázdné řádky nezapočítány) se nachází instrukce

```
soubor=file("textovysoubor.txt",'w')
```

Tato instrukce vytváří soubor s příponou txt a do kterého se bude pouze zapisovat, proto je vedle názvu souboru uveden příznak **w**. Tento řádek lze nahradit použitím pouze

```
soubor=open("textovysoubor.txt",'w')
```

Tato instrukce otevírá soubor s daným názvem pro zápis do souboru. V případě, že soubor neexistuje, vytvoří se nový soubor. Po otevření souboru se začne zapisovat na začátek souboru. V našem případě se jako první zapíše slovo **namereno\n**, jenže poslední dva znaky nikdy v souboru běžně neuvídíme. Jedná se o formátovací znaky, které zajistí, aby se příští zapsané slovo (číslo) umístilo na nový řádek souboru.

Může se vám stát, že budete chtít otevřít soubor s daty, aniž byste chtěli přepisovat už uložená data. Zkrátka potřebujete pouze přidávat na konec souboru další a další údaje. K tomu slouží atribut **append**, který uvedete v podobě „a“ místo „w“ v řádku: **soubor=open("textovysoubor.txt",'a')**

Ještě nutná poznámka k práci se soubory: vždy po skončení používání souboru pro čtení nebo zápis je nezbytné zavřít soubor. To je provedeno posledním řádkem našeho programu na obr. 22.

Nyní se zaměříme na blok, uvozený syntaxí **while x<10:** . Než jsem k tomuto bloku přišli, nadefinovali jsme si proměnnou **x** a vložili do ní hodnotu „nula“. Pomocí syntaxe **while x<10:**

říkáme programu, aby obsah bloku prováděl opakovaně tak dlouho, dokud bude hodnota proměnné menší než 10. Co provádí další řádky bloku-cyklu, je popsáno v obrázku. Pro lepší pochopení je zde také výpis hodnot od multimetru na obrazovku (**print**) a zpomalení chodu programu (**sleep**) aby bylo patrné opakované provádění bloku. Na konci bloku je provedena inkrementace (navýšení hodnoty) proměnné x o „jedničku“. Kdybychom tento řádek vynechali, cyklus by se prováděl donekonečna a my bychom museli přerušit chod programu (vpravo dole v oznamovací oblasti systému Windows rozevřít případnou roletku a na ikoně aplikace PythonWin kliknout pravou myší a vybrat možnost **Break into running code**.)

Existují ještě další příkazy typu cyklus (např **for i in range(10):**) a podmínkové příkazy (**if podmínka1:**), kterým souhrnně říkáme příkazy řídící běh programu. Těm se budeme věnovat později.

Celý výsledek úspěšně spuštěného programu je pak uložen v souboru s názvem **textovysoubor.txt**. Název si může zvolit každý sám. Do budoucna je dobré nepoužívat pro názvy souborů mezery ani znaky diakritiky, vyhnete se pak komplikacím s načítáním jména souboru.

Ukázky

Poznámka k jednotce visa_simul.py

Pokud nepracujete s přístrojovou jednotkou **visa.py**, ale se simulační jednotkou **visa_simul.py**, na pozici **instrument** v instrukci `instrument("GPIB0::XX::INSTR")`, viz str. 12, se použijí klíčová slova podle simulovaného přístroje

- **multim** (pro Multimetr Agilent 34401A), např. `multim("GPIB0::5::INSTR")`
- **zdroj** (pro řízený zdroj Agilent E3631A), např. `zdroj("GPIB0::6::INSTR")`
- **generator** (pro funkční generátor Agilent 33120A), např. `generator("GPIB0::7::INSTR")`

U jednotky `visa.py`, pracující s reálnými přístroji, se používá universální slovo **instrument**, jednotka si sama zjistí jakému přístroji volaná adresa přístroje patří.

Ukázka programu, jednoduché měření s použitím jednoho přístroje a simulační jednotky

```
# -*- coding: utf-8 -*-
from time import *
from visa_simul import *
# from visa import *

voltmetr=multim("GPIB0::22::instr") #přístroj je multimetr a ma adresu 22
voltmetr.write("*cls")
voltmetr.write("conf:volt:DC")

soubor=open("textovysoubor.txt", 'w')

soubor.write("Namereno, nadpis pouzijte uplny\n")
soubor.write("=====\n")
soubor.write("velicina/jednotka\n")
soubor.write("=====\n")

x=0
while x<(10):

    Udisplay=voltmetr.read()
    Useznam=Udisplay.split()
    Ustr=Useznam[0]
    print Ustr
    U=float(Ustr)
    soubor.write(Ustr + "\n") # zde pracuji s retezcem
    print U/2 # zde pracuji s cislem, retezec nelze delit
    sleep(1)
    x+=1
soubor.write("=====\n")
soubor.close()
```

Výstup programu jednoduché měření s použitím jednoho přístroje do souboru

Namereno, nadpis použijte uplny

=====

velicina/jednotka

=====

-0.973946

-0.868094

-0.298831

1.113392

0.635827

-0.627286

0.831842

0.145844

1.135280

0.952993

=====

Úprava výstupu do souboru s využitím formátovacího řetězce

Předpokládejme, že chceme výstupní měření uspořádat do souboru takto:

```
Zmeril Jan Novotny, Cejchovni krivka termočládku  
*****
```

```
-----  
T/°C    U/V  
-----  
20      022.73  
21      023.64  
22      024.55  
23      025.45  
24      026.36  
-----
```

Definujeme výstupní soubor `soubor=open("nejaky_soubor.txt","w")` a zaměříme se na zápis do souboru, tedy na příkaz `soubor.write()`. Využijeme tzv. formátovací řetězec.

Funkce **write()** je funkcí, která dokáže vytisknout formátovaný výstup, tedy výstup podle našeho požadavku, přičemž první parametr je tzv. formátovací řetězec. Formátovací řetězec určuje, v jakém tvaru se přesně vypíše následující parametry.

Formátovací řetězec

Formátovací řetězec je řetězec typu `"%+ 6.2f"`. Obsahuje speciální sekvence znaků, které budou ve výstupu nahrazeny odpovídající hodnotou. Takováto sekvence začíná znakem `%` a dále následuje:

- Pořadí hodnoty v n-tici (nepovinné) či klíč, pod kterým je hodnota dostupná ve slovníku (v případě použití slovníku pochopitelně povinné).
- Znak specifikující další možnosti konverze:
 - `#` - před osmičkové číslo bude vložen znak `0`, před šestnáctkové `0x`.
 - `0` - čísla budou zarovnána nulami.
 - `-` - zarovnání doleva.
 - - mezeru na místě znaménka před kladnými čísly (správně to má být obyčejná ASCII mezera, ale napište ji rozumným způsobem do HTML)
 - `+` - znaménko `+` před kladnými čísly
- Minimální délka hodnoty na výstupu (nepovinná).
- Nepovinný znak `.`, za nímž následuje přesnost.
- Typ konverze:
 - `d` nebo `i` - celé číslo se znaménkem
 - `u` - celé číslo bez znaménka
 - `o` - celé číslo v osmičkové soustavě
 - `x` (resp. `X`) - celé číslo v šestnáctkové soustavě malými (resp. velkými) písmeny.
 - `f` - číslo s plovoucí řádovou čárkou
 - `e`, `E`, `g`, `G` - různé varianty formátů čísel s plovoucí řádovou čárkou.
 - `s` (resp. `r`) - Řetězcová reprezentace hodnoty získaná funkcí `str()` (resp. `repr()`)
 - `c` - První znak řetězce.
 - `%` - znak `%`

A nyní prakticky pro naši potřebu. Například místo teploty **T** ve výstupu **write()** napíšeme:

```
"%+02.0f" %T
```

kde

- + zajistí povinné znaménko
- 0 zajistí povinné úvodní nuly
- 2 zajistí celkově 2 znaky ve výstupu
- 0 zajistí 0 (žádná) desetinná místa
- f zajistí číslo float
- T označení výstupní proměnné

Ukázka části programu, který zajistí požadovaný výstup

```
#pred cyklem

soubor=open("nejaky_nazev_souboru.txt","w")
soubor.write("Zmeril Jan Novotny, Cejchovni krivka termoelánku\n")
soubor.write("*****\n")
soubor.write("\n")
soubor.write("-----\n")
soubor.write("T/oC    U/V\n")
soubor.write("-----\n")

#v cyklu

    soubor.write ("%+02.0f" %T+" "+"%06.2f" %U+"\n")

#po cyklu

soubor.write("-----")
soubor.close()
```

Ukázka měření elektrického odporu virtuálně

```
# -*- coding: utf-8 -*-
#měření elektrického odporu virtuálně
from time import * #importování knihovny time
#from visa import *
from visa_simul import * #importování simulace merici jednotky
multimetr=multim("GPIB0::15::INSTR") #pokud merim naostro, místo multim použiji instrument
multimetr.write("*cls")
multimetr.write("CONF:RES")
#soubor=file("textovysoubor.txt",'w') #vytvoření
soubor=open("textovysoubor.txt",'w')
soubor.write("Namereno, nadpis použijte uplny\n") #uložení textu do souboru
soubor.write("=====\n")
soubor.write("    R/ohm          R2/ohm\n")
soubor.write("=====\n")
x=0 # nastavení proměnné x na nulu
while x<10: # cyklus se provádí, pokud je x menší než 10
    multimetr.write("conf:res")
    Rstr = multimetr.read() # vezmeme bezpečně jen řetězec před bílými znaky
    print "z přístroje ",Rstr # na obrazovce uvidíme každou hodnotu
    R=float(Rstr)
    print "po převodu ",R # zobrazím na displeji řetězec po převodu na číslo
    soubor.write("%+ 8.2f" %R + "%+ 8.2f" %R + "\n") # do souboru se vloží řetězec složený z obsahu
    # proměnné "R" a připojíme znaky pro skok na
    # další řádek
    sleep(1) # zastavíme chod programu na 1 sekundu
    x+=1 # zvýšíme hodnotu v proměnné x o jedna
soubor.close() # uzavření souboru a jeho uložení na disk
```