

# Návod GM328A Přístroj k určení a měření elektronických součástek, a elektrických hodnot.

Verse 1.13k  
ale také  
1.52m

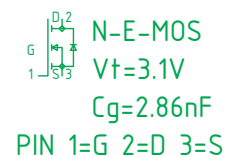
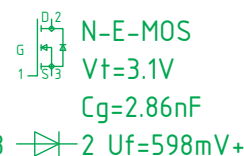
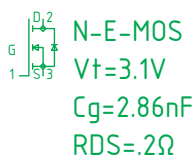
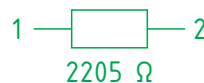
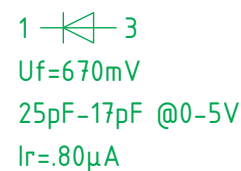
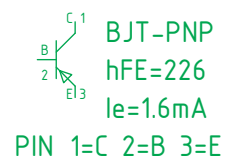
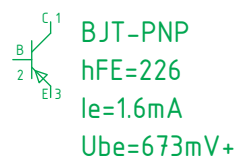
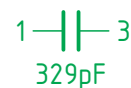
Karl-Heinz Kübbeler  
kh\_kuebbeler@web.de

&

Markus Reschke  
madires@theca-tabellaria.de

zkompileováno  
od bm-magic

12. května 2024



13.05.2019/MOR

---

## *Obsah*

---

<b>1</b>	<b>Základy</b>	<b>6</b>
1.1	Původ . . . . .	6
1.1.1	UPOZORNĚNÍ! . . . . .	6
1.2	Bezpečnost . . . . .	6
1.3	Licence . . . . .	6
1.3.1	Dostatečné licenční upozornění . . . . .	6
1.4	Rozdíly . . . . .	6
1.4.1	Upřesnění . . . . .	6
<b>2</b>	<b>Hardware</b>	<b>7</b>
2.1	Popis . . . . .	7
2.2	Ovládání . . . . .	8
2.2.1	Tlačítko . . . . .	8
2.2.2	Rotační kodér . . . . .	8
<b>3</b>	<b>Možnosti</b>	<b>9</b>
3.0.1	Výběrové menu . . . . .	11
3.1	Důležité poznámky pro použití . . . . .	11
3.2	Problemové součástky . . . . .	11
<b>4</b>	<b>Menu funkce v k verzi</b>	<b>12</b>
4.1	Volitelné funkce menu pro ATmega328 . . . . .	12
4.2	Autotest a kalibrace . . . . .	15
<b>5</b>	<b>Menu funkce v m verzi</b>	<b>16</b>
5.0.1	Zapnutí . . . . .	16
5.0.2	Hledání komponentů . . . . .	16
5.0.3	Monitorování baterie . . . . .	16
5.0.4	Vypínání, . . . . .	17
5.1	Výběrové menu . . . . .	17
5.1.1	PWM-Generátor . . . . .	17
5.1.2	Jednoduchý PWM . . . . .	17
5.1.3	Rozšířený PWM . . . . .	17
5.1.4	Obdélníkový signální generátor . . . . .	17
5.1.5	Zjištění Zenerového napětí (hardwarová úprava) . . . . .	18
5.1.6	Logický tester (hardwarová úprava) . . . . .	18
5.1.7	tester kontinuity (hardwarová úprava) . . . . .	19
5.1.8	ESR-Tool . . . . .	19
5.1.9	Unikající proud kondenzátoru . . . . .	19
5.1.10	R/C/L Monitory . . . . .	19
5.1.11	L/C-Měřič (hardwarová úprava) . . . . .	20
5.1.12	Čítač kmitočtů (hardwarová úprava) . . . . .	20
5.1.13	Jednoduchý čítač . . . . .	20

5.1.14	Rozšířený čítač . . . . .	20
5.1.15	Kroužkový tester (hardwarová úprava) . . . . .	21
5.1.16	Počítadlo událostí (s hardwarovou úpravou) . . . . .	21
5.1.17	Rotační kodér . . . . .	22
5.1.18	Kontrast . . . . .	22
5.1.19	Detektor/Dekodér pro IR dálkové ovládání . . . . .	22
5.1.20	IR dálkové ovládání . . . . .	23
5.1.21	Test optických spojek . . . . .	24
5.1.22	Test fotodiod . . . . .	24
5.1.23	Rychlý test diod/LED . . . . .	25
5.1.24	Test servopohonů pro modely . . . . .	25
5.1.25	OneWire skenování . . . . .	25
5.1.26	Snímače teploty DS18B20, DS18S20 . . . . .	25
5.1.27	Senzory teploty a vlhkosti DHTxx . . . . .	26
5.1.28	Převodník termočlánků . . . . .	26
5.1.29	BH1750 Snímač okolního světla . . . . .	27
5.1.30	Svítilna . . . . .	27
5.1.31	Autotest . . . . .	27
5.1.32	Samočinné nastavení . . . . .	27
5.1.33	Ušchovat/Použít . . . . .	28
5.1.34	Ukázat hodnoty . . . . .	28
5.1.35	Znaková sada/symboly . . . . .	28
5.1.36	Vypnout . . . . .	28
5.1.37	Konec . . . . .	28
5.2	config.h . . . . .	29
5.2.1	Hardwarová obsluha . . . . .	29
5.2.2	Možnosti softwaru . . . . .	32
5.2.3	Řešení obcházení pro některé testery . . . . .	35
5.2.4	Řešení pro některá IDE . . . . .	35
5.2.5	uživatelské rozhraní . . . . .	35
5.2.6	Správa napájení . . . . .	38
5.2.7	Nastavení a kompenzace měření . . . . .	39
5.2.8	R&D - určeno pro vývojáře firmwaru . . . . .	40
5.2.9	Busse . . . . .	40
<b>6</b>	<b>Programový kód</b>	<b>41</b>
6.1	V k-verzi . . . . .	41
6.2	V m-verzi vypadá ta situace úplně jinak. . . . .	41
6.3	Makefile . . . . .	41
6.3.1	MCU-Typ . . . . .	41
6.3.2	MCU-Taktfrequenz . . . . .	41
6.3.3	Oszillator-Typ . . . . .	41
6.3.4	Avrdude MCU-Typ . . . . .	42
6.3.5	Avrdude ISP-Programmierer . . . . .	42
6.4	config.h . . . . .	43
6.4.1	Pro tento tester musí být změněno . . . . .	43
6.4.2	Nyní začíná utrpení nutného výběru . . . . .	43
6.5	Config_328.h . . . . .	46
6.5.1	Nutné změny . . . . .	46
6.5.2	informace . . . . .	47
<b>7</b>	<b>Programování testeru</b>	<b>48</b>
7.1	Konfigurace testeru . . . . .	48
7.2	Programování testeru . . . . .	48
7.3	Operační system Linux . . . . .	48

7.4	Použití s Linuxem . . . . .	49
7.5	Instalace programových balíčků . . . . .	49
7.6	Stáhnutí zdrojů . . . . .	49
7.7	Používání rozhraní . . . . .	49
7.8	Členství ve skupině . . . . .	50
7.9	pracovní prostředí . . . . .	50
7.10	Přeložení Firmware . . . . .	51
7.11	Hardware k programování . . . . .	52
7.11.1	Programátor . . . . .	52
7.11.2	Možnosti nákupu: . . . . .	52
<b>8</b>	<b>Technické údaje</b>	<b>53</b>
8.1	Pomoc a otázky . . . . .	53
8.2	A pro chvíli oddechu . . . . .	53
8.3	Schema GM 328 A . . . . .	54

## Úvod

**Hlavní motivy** Každý z nás zná tento problém: vymontuje transistor nebo ho najde mezi svými poklady, když je jeho označení čitelné a technické údaje nebo náhrada dostupné, je všechno v pořádku. Pokud ale ne, nastává otázka, co je to za součástku. S konvenčními měřicími metodami je těžké a zdouhavé typ součástky a její parametry zjistit. Může se jednat o NPN, PNP, N- nebo P-Kanal-MOSFET atd. Nápad Markuse F., je, aby tuto práci za nás udělal AVR-Mikrokontrolér.

**Nadcházející věty** jsou opsány z návodu na Tranzistor Tester od Karl-Heinz Kübbelera. Ostatně, velký díl tohoto návodu pochází z výtažků díla tohoto autora,

...kterému bych rád tímto poděkoval...

**Můj kontakt s tímto testerem** byl čistě náhodný. Při hledání laciných součástek na můj nový projekt jsem ho našel za cenu patřičně nižší, než je cena samotného displeje. Poté, co tester přišel a já naletoval připojení baterie, jsem vůbec nebyl překvapený, když zůstal displej tmavý. Prodáváč, kterého jsem kontaktoval, mě poslal adresu ...

<https://www.youtube.com/watch?v=0bfxyy1K3po>, a když jsem tam viděl, jak ten zkoušeč mačká kódér... jsem se moc styděl a zároveň ...

**žasnul** co všechno tato destička umí. Při prvním zapnutí jsem byl doveden ke kalibraci, která je tak organizovaná, že nelze udělat žádnou chybu. Zároveň jsem byl upozorněn na: ...  
[svn://mikrocontroller.net/transistortester](http://mikrocontroller.net/transistortester)

kde jsem dostal kompletní dokumentaci, ve které jsem se dočetl, že existují různé ovládací řeči.

**V mé mladistvé lehkomyšlnosti** jsem se rozhodl ho naučit německy.

Na tomto místě bych měl dodat, že při koupi, začátkem roku 2018, mě bylo 73 let.

- Začalo to vlastně již v mých 72 letech, kdy jsem se rozhodl, že je na čase se naučit programovat. Již před více než 30 lety jsem si postavil (tenkrát narychlo z nouze) počítač událostí, který mě ještě dělá dobré služby. Jak to tak chodí, žijí provizoria nejdéle... přesto jsem zvolil, jako první projekt, postavit tento čítač v softwaru. Koupil jsem AVR kurs se stavebnicí, ale jako většina výuk jsem se ale nedostal dál než k zapínání LED a z nich vyrobený semafor pro chodce. Doposud jsem nenašel žádnou pomůcku, která vysvětlí, jak nahradit logické IC jako (FF, SRT) softwarem. Návod na softwarový, událostí ovladatelný, čítač/odčítač, který bych mohl na můj projekt použít, také ne.

- Kromě toho jsem neměl do té doby žádný kontakt ani zkušenosti s AVR.

Při prvních potížích, jsem se obrátil na autora (dále khk), který mě trpělivě (během asi tří měsíců) asi v kolem 50 mailech dostal tak daleko, že ten tester umí i česky.

- Z vděčnosti jsem slíbil, že přeložím jeho dokumentaci do češtiny. V dalších mailech mě khk představil LaTeX, ve kterém je tato dokumentace napsaná.

**Dodatek pro český překlad...** ..., to jsem si ale představoval, asi jako Hurvínek válku. Již po prvních pěti větách jsem si uvědomil, že před 50 lety co žiji v Německu, žádná informatika neexistovala a že já znám veškeré technické výrazy JEN v němčině nebo v angličtině. Zkouška překládat pomocí Google dopadla velmi špatně.

- Tak jsem na 130 stránek potřeboval skoro rok. Když jsem byl "hotový", tak khk z osobních důvodů nemá čas, tak že mojí práci doposud nezveřejnil...

- **Mezi tím jsem kontaktoval** druhého vývojáře, Markuse Reschke, který na tom testeru vyvíjí paralelně od počátku. Jeho software je moc zajímavá, ale jeho konfigurace ne tak dobře popsána. (Jen jako \*.txt a to ještě většinou v angličtině).

- Abych tomu lépe rozuměl, převedl jsem ji na \*.pdf. Autor můj převod vydal na své webové stránce [3] a kromě toho přidal do své verze počítaadlo. (To není sice takové jaké potřebuji, ale mám konečně základ, na kterém budu, až budu mít zase čas, pokračovat).

...také tomuto vývojáři bych tímto, rád poděkoval...

- Jeho verzi jsem nyní také přeložil do (Staro) češtiny.

**Tyto řádky** slouží k ušetření tvých nespavých nocí při výběru vhodné verze.

### 1.1. Původ

Původem je zkoušečka tranzistorů založena na projektu Markuse Frejka [1] s pokračováním Karl-Heinze Kübbelerem [2] a Markusem Reschkem [3].

Oba vývojáři napsali k svým testerům opravdu dobrou dokumentaci.

Následovně jsou použity výtažky z jejich dokumentací.

Určitě si přečti jejich originály!

**1.1.1. UPOZORNĚNÍ!** původní čínská softwarová verze má v ATmega328 P nastaveny bezpečnostní bity, proto není možné tento stav zálohovat.

Tím pádem ... bohužel ... nevede žádná cesta, zpět k původní verzi softwaru.

Návrh: Tento model používá lehce vyměnitelný ATmega 328 P, takže můžeš vypálit rozdílné konfigurace, podle potřeby je měnit a přesto originál zachovat. ;-)

### 1.2. Bezpečnost

Tester není multimetr!

Je to jednoduchý tester součástek, který dokáže měřit různé věci.

Vstupy nejsou chráněny a napětím nad 5V budou poškozeny.

Nepoužívej tester pro obvody v provozu, ale pouze pro jednotlivé součástky! U kondenzátorů se ujisti, že jsou vybité **před zapnutím** testeru.

Používáš na vlastní nebezpečí!

### 1.3. Licence

Autor původní verze má pouze dvě licenční podmínky.

Za prvé je projekt otevřený zdrojový kód,

a za druhé by měli komerční uživatelé kontaktovat autora.

- Bohužel, ani Karl-Heinz, ani Markus se dosud k autorovi nedostali.

K vyřešení problému s nedostačující "open source" licencí, vybral Markus 1.1.2016 standardní licenci pro otevřený zdroj, poté co měl původní autor dost času oznámit svá přání k licenci.

Vzhledem k tomu že tyto verze firmwaru jsou zcela nové verze, které zabírají jen několik nápadů původního firmwaru, ale nesdílí žádný kód, by to mělo být odůvodněno.

Licencováno v rámci EUPL V.1.1

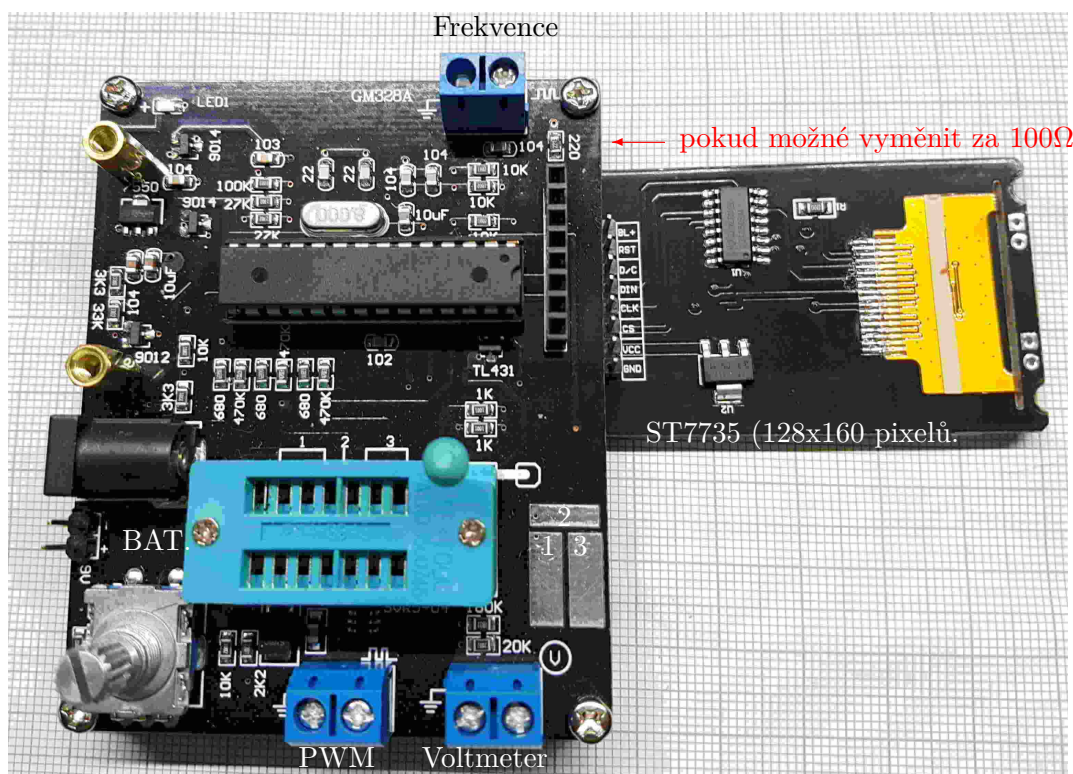
**1.3.1. Dostatečné licenční upozornění** Názvy produktů nebo společností mohou být registrované ochranné známky příslušných vlastníků.

### 1.4. Rozdíly

Zatímco je firmware od Karl-Heinze oficiální verzí, která podporuje i starší typy MCU ATmega, slouží verze od Markuse k vyzkoušení a testování nových nápadů, nabízí některé změny uživatelského rozhraní a postupy měření a je omezena na ATmegasy s minimálně 32kB Flash.

**1.4.1. Upřesnění** obě verze jsou koncipovány pro použití v různých testerech s rozličnou hardware. Některé možnosti nelze v tomto testeru použít, alespoň ne bez hardwarové úpravy. Na druhé straně nabízí software tolik možností, že to přesahuje kapacitu ATmega paměti, takže není možné, současné vyzkoušení.

## 2.1. Popis



Obrázek 2.1. Pohled s odklopeným LCD

- Jak je již naznačeno ve jménu používá tento tester ATmega 328 P s DIP patičí, což umožňuje externí programování, což je také nutné, proto že nemá ISP konektor.
- Takt je obsazen 8 MHz krystalem. Dodatečná výměna 22pF Kondenzátoru za trimr z důvodu optimalizování frekvence je možná.
- K zobrazení je použitý barevný displej s ST7735 řadičem (128x160 pixelů). Jak je na obrázku 2.1 vpravo dobře viditelné, používá display vyrovnávací paměť CD4050 (IC1), pro nastavení úrovně signálu a 3,3V regulátor napětí (IC2) pro napájení. Výměna odporu podsvětlení zvýší čitelnost displeje.
- Externí 2,5V reference je realizovaná s TL431.
- Souprava má zásuvku pro napájení. Kromě toho je připravené místo k připojení 9V baterie.
- K ovládání slouží rotační kodér s tlačítkem.
- Testovací porty jsou přístupné přes 14 kolíkovou Textool zásuvku (X1), pro SMD součástky je připravena zkušební podložka.
- Jak je vidět na plánu v podkapitole 8.3 na stránce 54 jsou testovací vstupy částečně chráněny diodovým IC SRV05-4 (IC2).
- Tester nabízí frekvenční výstup přes svorku (X2). Ten je ale pouze paralelně připojený k TP2.
- Další svorka (X3) je k měření pozitivního DC napětí do 50V. Tento vchod neposkytuje žádnou ochranu!
- Třetí svorka (X4) nabízí vstup pro měření frekvence. Také zde není žádná ochrana vstupu.

## 2.2. Ovládání

je realizováno rotačním kodérem s tlačítkem.

Obsluha testeru je jednoduchá. Nicméně zde je pár rad pro jeho použití.

V každém případě můžeš do tří testovacích bodů připojit tříbodové součástky v libovolném pořadí. U dvoupólových součástek můžeš použít kterékoliv dva testovací porty.

Nezáleží ani na polaritě, to znamená že i etyly mohou být připojeny libovolně. Měření kapacity se však provádí tak, že záporný pól je na měřicím portu s nižším číslem.

Protože měřicí napětí leží mezi 0,3V a maximálně 1,3V, nehraje zde polarita důležitou roli.

Je-li součástka připojena, nesmíš se jí během měření dotýkat. Dej ji na izolační podklad, pokud není v zásuvce. Nedotýkej se se ani izolace měřicích kabelů, výsledek měření tím může být ovlivněn.

Poté stiskni tlačítko start. Po úvodním hlášení se zobrazí výsledek měření asi do dvou sekund.

Při měření kondenzátorů může, v závislosti na kapacitě, trvat mnohem déle.

**Co se stane poté, závisí na použité softwarové verzi a konfiguraci.**

Aby bylo možné srovnání, byly v obou verzích zvoleny, pokud možno, stejné konfigurace a časy.

**2.2.1. Tlačítko** **zapíná** tester a slouží k **obsluze**. V automatickém režimu čeká tester 30 vteřin na součástku...poté vypne pro úsporu baterie.

Je-li v tom čase vložena součástka chovají se tyto verze **rozličně**.

**V m verši** vypne tester automaticky po 30 vteřinách. Dříve vypneš delším stiskem tlačítka.

- Nové měření docílíš krátkým stiskem, nebo otočením kodéru doprava.
- Menu dosáhneš dvojitým stiskem nebo točením kodéru doleva.

Tester rozlišuje mezi:

1. **krátkým stisknutím**, které se obvykle používá k pokračování funkce nebo k výběru další položky nabídky,
2. **dlouhým zmáčknutím** (> 0,3s), které provádí kontextovou akci a
3. **dvojitým stisknutím**, které akci ukončí.

Pokud tester očekává stisknutí klávesy pro pokračování v aktuální akci, je to označeno kurzorem v pravé dolní části modulu LCD.

Statický kurzor signalizuje, že následuje více informací  
a blikající kurzor znamená, že hledání součástek pokračuje.

U nabídek a některých dalších funkcí se kurzor nezobrazí, protože očekávaný vstup by měl zde být jasný.

**V k verši** tester nevypne, ale čeká na další součástku, kterou měří po vložení automaticky.

K vypnutí je nutné zmáčknutí tlačítka nebo rychlé točení kodérem libovolným směrem.

- Menu dosáhneš po zapnutí **bez součástky** a to dlouhým stiskem (> 0.5s) nebo rychlým točením kodéru. Po vložení součástky zde menu nedosáhneš.

**2.2.2. Rotační kodér** **dodává** další funkce, závislé na kontextu.

Podrobnosti jsou vysvětleny v následujících částech.





Některé funkce umožňují větší změny díky rychlosti otáčení nebo skoky hodnot.



Detekce rychlosti otáčení měří dobu dvou kroků. Proto bys měl s kodérem udělat alespoň dva kroky pro střední rychlost. Pro vyšší rychlost to jsou tři kroky.

Jediný krok vede vždy k nejnižší rychlosti.

Detaily se dozvíš v dokumentaci, přímo od autorů příslušné verze. Viz [3].



1. Automatická detekce NPN a PNP bipolárních tranzistorů, N- a P-KANÁLOVÝ MOSFET, JFETs, diody, dvojité diody, N- a P-IGBT, tyristory a triaky. Pro tyristory a triaky musí být dosaženo dostatečné zapalovací a udržovací napětí a proudy. U IGBT musí být prahové napětí brány nižší než  $5V$ .
2. Znázornění rozložení pinů testovacích součástek.
3. Měření stávajícího zesilovacího činitele a prahového napětí báse-emitor pro bipolární tranzistory.
4. Darlingtonovy tranzistory jsou charakteristické vyšším prahovým napětím a vysokým proudovým zesílením.
5. Automatická detekce ochranné diody v bipolárních tranzistorech a MOSFETů.
6. Měření prahového napětí, vstupní kapacity a  $R_{DSon}$  s hradlovým napětím těsně pod  $5V$  u MOSFETů.
7. Jsou měřeny a zobrazeny až dva odpory jako  symboly a jejich hodnoty jsou až na čtyři desetinná místa ve správné hodnotě. Všechny symboly jsou zarámovány s testovacími čísly, jak byly nasazeny do zkoušečky (1-3). Proto lze také měřit potenciometry. Když ale potenciometr dosáhne koncové polohy, není možné rozlišit mezi prostředním a koncovým kontaktem.
8. Odpory lze nyní měřit od  $0,01\Omega$ , do  $50M\Omega$ .
9. Kondenzátor je také detekován a změřen. Je označen symbolem  Jeho kapacita je určena a zobrazena až na čtyři desetinná místa přesně. Hodnota může být v rozmezí od  $25pF$  (při  $8MHz$  taktu,  $50pF$  při  $1MHz$  taktu) do  $100mF$ . Rozlišení je  $1pF$  (u  $8MHz$  taktu).
10. U kondenzátorů s kapacitou větší než  $20nF$  je kromě toho měřen ještě ekvivalentní sériový odpor (ESR) kondenzátoru s rozlišením  $0,01\Omega$  a zobrazen na dvě desetinná místa. Tato funkce je k dispozici pouze tehdy, pokud má ATmega nejméně 16K flash paměti.
11. U kondenzátorů s kapacitní hodnotou nad  $5000pF$  lze po nabíjecím impulsu určit ztrátovou hodnotu Vloss. Ztrátová hodnota v procentech indikuje kvalitu kondenzátoru.
12. Až dvě diody jsou označeny symbolem  nebo symbolem  a jsou zobrazeny ve správném pořadí. Kromě toho jsou zobrazeny úbytky napětí na diodách.
13. LED dioda je rozpoznána jako dioda, úbytek napětí je ale mnohem vyšší než u normální diody. Dvojité diody jsou rozpoznány jako dvě diody.
14. Zenerovy diody lze detekovat, když je Zenerovo napětí pod hodnotou  $4,5V$ . Zobrazují se jako dvě diody, rozpoznat je lze jen přes zobrazené napětí. Vnější čísla zkušebního kontaktu obklopující symboly diod jsou v tomto případě totožné. Skutečnou anodu diody lze nalézt pouze pro diodu, jejíž prahové napětí je blízké napětí  $700mV$ !
15. Pokud se zjistí více než 3 diody, zobrazí se spolu s chybovou zprávou počet nalezených diod. K tomu může dojít pouze v případě diod na všech třech zkušebních pinech a jsou spojeny a alespoň jedna z nich je Zenerova dioda. V tomto případě je třeba připojit pouze dva testovací kontakty a restartovat skenování a měřit jednu diodu za druhou.
16. Kapacita diody v závěrném směru je určena automaticky. Bipolární tranzistory lze také testovat, pokud je připojena pouze báze a buď kolektor nebo emitor. Pro ATmega s více než 8k flash paměti je kromě toho měřen ještě zpětný proud s rozlišením  $2nA$ . Hodnota je zobrazena pouze tehdy pokud je rozdílná od nuly.
17. Zapojení usměrňovacího můstku lze zjistit pouze jedním měřením.
18. Kondenzátory s hodnotami kapacity pod  $25pF$  není možné běžně rozpoznat, ale mohou být použity společně s diodou zapojenou paralelně nebo s paralelně připojeným kondenzátorem

- kapacity nejméně  $25pF$ . V tomto případě musí být od výsledku měření odečtena hodnota kapacity součásti zapojené paralelně. U procesorů s minimální pamětí 32K flash se tester změni pomocí kondenzátoru  $> 25pF$  mezi TP1 a TP3 v cyklické měření kondenzátoru, která také přímo měří kapacity od  $1pF$ .
19. Pro odpory pod  $2100\Omega$  se také provádí měření indukčnosti pokud má ATmega nejméně 16K flash paměti. Kromě symbolu odporu  se zobrazí symbol indukčnosti . Rozsah zobrazení je asi  $0,01mH$  až přes  $20H$ , ale přesnost není vysoká. Výsledek se zobrazuje pouze pro jeden rezistor společně s hodnotou odporu.
  20. Doba měření je asi dvě sekundy, měření kapacity a indukčnosti mohou trvat déle.
  21. Software lze konfigurovat pro sérii měření s předem definovaným počtem opakování, než se automatické vypne.
  22. Vestavěná funkce automatického testování včetně volitelného frekvenčního generátoru  $50Hz$  pro přesnost kontroly frekvence a časové prodlevy (pouze s minimálně 16 kB flash pamětí).
  23. Volitelná možnost kalibrace pro měření kondenzátoru a vnitřní odpor pro automatické určování portů během samočinného testu (pouze s minimálně 16 kB flash pamětí). Externí kondenzátor s kapacitou mezi  $100nF$  a  $20\mu F$  na testovacích kontaktech TP1 a TP3 je nutný, pro kompenzaci vyrovnávacího napětí analogového komparátoru. To může snížit chybu měření při měření kapacity až na hodnotu  $40\mu F$ . Stejným kondenzátorem je korekční napětí pro nastavení správného zesílení pro výpočet měření ADC pomocí vnitřního referenčního napětí  $1,1V$ .
  24. Zobrazení kolektor - emitor zbytkového proudu  $I_{CE0}$  při odpojené bázi ( $1\mu A$  přesnost) a zbytkový proud kolektor - emitor  $I_{CES}$  s bází připojenou na potenciál emitoru (pouze s minimálně 16K flash pamětí). Tyto hodnoty se zobrazují pouze v případě, že nejsou nulové (zejména pro germaniové tranzistory).
  25. Pro ATmega s minimálně 32K flash pamětí se tester přepne z multifunkčního testu na režim měřiče odporu, pokud je v automatickém režimu rozpoznání součástek, pouze jeden odpor na testovacích kontaktech (TP1) a (TP3). Pokud je v souboru Makefile zapnuto pomocí volby RMETER\_WITH\_L při měření odporů také měření indukčnosti, měří se také. Provozní režim je indikován s [R] nebo [RL] na pravé straně 1 řádku displeje. Přesně tak, jak se tester přepne na měřič kapacit, když byl mezi TP1 a TP3 detekován kondenzátor. Tento provozní režim je označen symbolem [C] na pravé straně 1 řádku displeje. V tomto režimu lze měřit kondenzátory od  $1pF$ . Pouze pro automatické spuštění funkce potřebujete kondenzátor s více než  $25pF$ . Obě speciální funkce lze opět ukončit stisknutím tlačítka. Tester poté funguje v normálním režimu.
  26. U procesorů s min. 32 kB flash pamětí je přístupné po dvousekundovém stisku tlačítka menu, což zprovozní další funkce. Menu lze samozřejmě použít také k návratu k funkci testeru tranzistoru.
  27. Pomocí funkce menu lze provést měření frekvence na portu PD4 ATmega. Rozlišení je  $1Hz$  na vstupních frekvencích nad  $33kHz$ . Při nižších frekvencích může být rozlišovací schopnost až  $0,001mHz$ . Přečti si prosím podkapitulu 2.2.4 v dokumentaci [2], jak musí být frekvenční signál připojen.
  28. Pomocí funkce menu a při vypnutí UART módu lze měřit externí napětí do 50 V přes 10:1 dělič napětí na PC3 kontaktu. U PLCC-ATmega328 varianty je možné jeden z těch dvou přidaných kontaktů dohromady s UART rozhraním použít na měření napětí. Pokud je připojené rozšíření pro měření Zenerovy diody (převodník DC-DC), je možné v této větvi, při současném podržení tlačítka, testovat Zenerovy diody.
  29. Pomocí další funkce menu lze na kontaktu TP2 (PB2 port ATmega) zapnout výstup frekvence. V současné době lze frekvence nastavit od  $1Hz$  do  $2MHz$ .
  30. Pomocí další funkce menu lze zapnout na pinu TP2 (PB2 port ATmega) pevně danou frekvenci s nastavitelnou šířkou impulsu. Šířku impulsu lze zvýšit o 1% krátkým stiskem klávesy a o 10% s delším stiskem.
  31. Pomocí další funkce menu lze spustit speciální měření kondenzátoru s měřením ESR. Tato funkce se při výběru nazývá C+ESR@TP1:TP3. Kapacity od přibližně  $2\mu F$  až do  $50mF$  mohou být měřeny pro nízké měřicí napětí okolo  $300mV$  v zapájeném stavu.

32. Pro procesory s alespoň 32K flash pamětí (Mega328) lze ADC použít s metodou vzorkování, která umožní měřit kondenzátory pod  $100pF$  s rozlišením  $0,01pF$ . Stejným způsobem je možné měřit také cívky pod  $2mH$  čímž lze dosáhnout výrazně lepší rozlišení než rezonanční frekvence s paralelním kondenzátorem známé velikosti.
33. Monitorování baterie je možné nastavit podle tvých představ. Každý cyklus hledání součástek začíná zobrazením napětí baterie a jejího stavu (ok, slabý, prázdný). Při poklesu pod její prahové napětí tester vypne. Baterie je kontrolována i během provozu.

**3.0.1. Výběrové menu** nabízí další volitelné možnosti testeru. Tyto jsou rozlišné podle použité verze. Některé funkce se stejným nebo podobným názvem nabízí jiné možnosti a má také jiné ovládání. Důkladnější popis najdeš přímo u popisu verze. Viz [3].

### 3.1. Důležité poznámky pro použití

**Pozor:** Vždy zkontroluj, zda jsou **kondenzátory** před připojením ke zkoušečce, nejlépe zkratováním, **vybité** ! V opačném případě by mohlo dojít k poškození přístroje ještě před jeho zapnutím. ! ATmega nabízí jen málo vlastní ochrany. ! Zvláštní pozornost je třeba věnovat také při měření v zapojení. Přístroj by měl být vždy předem odpojen od napájení a měli byste se přesvědčit, že v přístroji není **žádné zbytkové napětí**.

Všechna elektronická zařízení uvnitř obsahují kondenzátory!

Při měření malých odporů je třeba věnovat zvláštní pozornost odporu měřicích kabelů a přechodových odporů kontaktů.

Kvalita a stav konektorů hraje velkou roli, stejně jako odpor měřicích kabelů.

Totéž platí pro měření hodnoty ESR kondenzátorů.

Se špatnými měřicími kabely s krokosvorkami se může ESR odpor z  $0,02\Omega$  dosáhnout lehce hodnoty  $0,61\Omega$ .

Pokud je to možné, připájej měřicí kabely s krokosvorkami k testovacím portům paralelně s existujícími konektory. Pak nemusí být tester, při měření malých kapacit pokaždé kalibrován, pokud měříte pomocí zkušebních kabelů, nebo bez nich.

Při kalibraci nulového odporu je však rozdíl, pokud jsou testovací piny připojeny ke zkušebním svorkám přímo na základně nebo přes kabel.

Pouze ve druhém případě je odpor kabelu a svorek kalibrován.

Pokud máš pochybnosti, proved kalibraci pomocí zkratu na zkušební zásuvce a poté změř odpor zkratovaných měřicích kabelů.

### 3.2. Problemové součástky

Ve výsledcích měření bys měl mít vždy na paměti, že byl tester navržen pro citlivé součástky.

Obvykle je maximální měřicí proud pouze  $6mA$ .

Výkonové polovodiče často způsobují problémy při zjišťování, nebo měření vysokých zbytkových proudů malým měřicím proudem.

Pro tyristory a triaky nejsou často dosaženy spínací, nebo přídržné proudy.

To je důvod, proč je občas tyristor detekován jako NPN tranzistor, nebo dioda.

Stejně tak se může stát, že některý tyristor, nebo triak nebude vůbec rozpoznán.

Další Problém vzniká s detekováním polovodičů obsahujících integrované odpory, takže dioda báze-emitor BU508D tranzistoru nebyla v důsledku paralelně zapojeného vnitřního  $42\Omega$  odporu detekována. Z toho plyne, že zde funkce tranzistoru nemůže být testována.

Problémy s rozpoznáním jdou často také u výkonových tranzistorů Darlington. Tady je také často vestavěný odpor mezi bází a emitorem, které komplikují detekci kvůli nízkým měřicím proudům, které se zde používají.

**Po delším zmáčknutí tlačítka** ( $> 0.5s$ ) se ukáže výběrové menu.

Nabízená funkce se nachází ve třetím řádku displeje. Přitom je předcházející funkce ve druhém a následující ve čtvrtém řádku. Krátkým zmáčknutím se postupuje k další volbě.

Delším zmáčknutím startuje nabízená funkce.

Po poslední možné funkci „vypnout” se ukáže zase funkce první. (Cyklické udání).

Výběrové menu lze také dosáhnout rychlým otočením kodéru během ukázky předcházejícího měření.

Pomalým točením je možné vybrat kteroukoli funkci libovolným směrem.

Uvnitř funkce je možné změnit pomalým točením její parametry.

Rychlým otáčením se vrátí tester zpět do výběrového menu.

**Jednotlivé měření** Pokud je tester konfigurován pro jedno měření (POWER\_OFF-volba), Pokud nespustíte nové měření, vypne se tester kvůli úspoře baterie automaticky za 28 sekund (konfigurovatelné). Po vypnutí lze samozřejmě spustit nové měření, buďto se stejnou součástí, nebo také s jinou. Pokud chybí vypínací elektronika, zůstane zobrazen poslední výsledek měření.

**Kontinuální měření** Zvláštním případem je konfigurace bez funkce automatického vypnutí. V tomto případě je nutno nastavit možnost POWER\_OFF v makefile. Tato konfigurace se obvykle používá pokud nejsou osazeny tranzistory pro automatické vypnutí. Místo toho je zapotřebí externí vypínač pro zapnutí / vypnutí. Zde tester opakuje měření až do vypnutí.

**Opakované měření** V tomto případě se testovací přístroj nevypne přímo po měření, ale až po zvoleném počtu. Chcete-li to nastavit, je volbě POWER\_OFF v makefile přiřazeno číslo opakování (například 5). Ve standardním případě se přístroj vypne po pěti měřeních bez rozpoznání komponentu. Pokud je detekována další měřená součástka, vypne se po dvojnásobku, tj. Deset měření. Jediné měření s nerozpoznanou komponentou vynuluje počet zjištěných kusů na nulu. Stejně tak jediné měření s detekovanou komponentou vynuluje počet nerozpoznaných komponent na nulu. To má za následek, že i bez stisku startovacího tlačítka lze měřit další a další kusy, pokud se součástky pravidelně obměňují. Změna součásti s prázdnými měřicími svorkami mezitím provede měření bez detekované součásti. V tomto provozním režimu je pro zobrazení času speciální funkce. Při krátkém stisku start tlačítka, jsou výsledky měření zobrazeny pouze 5 sekund. Pokud držíte tlačítko start, až do zobrazení první zprávy, je doba zobrazení 28 sekund, jako u jednotlivého měření. Další měření v době zobrazování, je umožněno následujícím stiskem Start tlačítka.

#### 4.1. Volitelné funkce menu pro ATmega328

Když je zapnuta funkce menu, začne tester po dlouhém stisku tlačítka ( $> 0.5s$ ) volbu dalších funkcí. Tato funkce je k dispozici i pro jiné procesory s minimálně 32K flash pamětí. Volitelné funkce se zobrazují na řádku 2 dvouřádkového displeje nebo na řádku 3 čtyřřádkového displeje. Předchozí a následující funkce jsou zobrazeny v řádcích 2 a 4. Po delším čekání bez odezvy tlačítka se program vrátí k normální funkci testeru. Krátkým stiskem tlačítka můžete přepnout na další volbu. Dlouhým stisknutím tlačítka se spustí zobrazená doplňková funkce. Po zobrazení poslední funkce "Vypnout" se znovu zobrazí první funkce. Pokud je Tester vybaven pulzním enkodérem lze výběr nabídky docílit rychlým otáčením enkodéru. Funkcemi nabídky lze listovat pomalým otáčením voliče v libovolném směru. Zvolenou funkci nabídky, lze spustit pouze stiskem tlačítka. V rámci vybraných nastavení funkce jsou další parametry volitelné pomocí pomalého otáčení enkodéru. Rychlým otočením enkodéru se vrátíte do nabídkového menu.


**Frekvence** Přídavná funkce "frekvence" (frekvenční měření) používá jako vstupní pin PD4 AT-



mega, který je také připojen k LCD. Nejdříve je vždy měřena frekvence, při frekvencích pod  $25\text{kHz}$  je také měřena střední hodnota vstupního signálu a z toho je vypočtena frekvence frekvence s rozlišením až  $0,001\text{Hz}$ . Pokud je v souboru Makefile nastavena v makefile volba `POWER_OFF` je doba měření frekvence omezena na 8 minut. Měření frekvence je ukončeno stisknutím tlačítka a návratem do nabídky menu.


**f-generátor** Pomocí doplňkové funkce "f-generátor"(frekvenční generátor) lze zvolit frekvenci mezi 1Hz a 2MHz. Nastavení frekvence lze měnit pouze v nejvyšším řádu. Pro frekvenci 1Hz až 10kHz jsou volitelná čísla 0-9. Od 100kHz je možné volit 0-20. V prvním řádku oznámí symbol  $>$  nebo  $<$  angezeigt, zde je možné delším tiskem ( $> 0.8\text{s}$ ) přepnout na vyšší nebo nižší místo. Na nižší místo ( $<$ ) lze přepnout pouze tehdy, je-li aktuální číslice nastavena na hodnotu 0 a pokud nebyl zvolen kroky nižší než 1Hz. V řádu 100kHz je symbol  $>$  nahrazen znakem R (reset). Delší tisk způsobí návrat frekvence na počáteční hodnotu 1Hz. Je-li v Makefile nastavená volba `POWER_OFF`-musí být stisk tlačítka pro změnu frekvence delší. Krátké stisknutí tlačítka ( $< 0,2\text{s}$ ) pouze resetuje sledování času 4 minut. Uplynulý čas je zobrazen v 1 řádku tečkou za každých 30 sekund. Pravidelným krátkým stiskem tlačítka lze zabránit předčasnému vypnutí generování kmitočtu. Dlouhé stisknutí tlačítka ( $> 2\text{s}$ ) způsobí návrat do menu.

**10-bit PWM** Přídavná funkce "10bitové PWM"(šířková modulace impulsů) generuje pevnou frekvenci s nastavitelnou šířkou impulsu na pinu TP2. Při krátkém stisknutí klávesy ( $< 0,5\text{s}$ ) se šířka impulsu zvýší o 1%, s delším stisknutím klávesy o 10%. Při překročení 99% bude 100% odečteno (zpátky na 0). Při zvolené možnosti `POWER_OFF`-v makefile, bude po 8 minutách bez stisku tlačítka, tester vypnutý. Konec generování je také možné dosáhnout dlouhým ( $> 1,3\text{s}$ ) stiskem.

**C+ESR@TP1:3** Pomocí rozšiřující funkce "C + ESR @ TP1: 3" se na TP1 a TP3 spustí samostatné měření kondenzátoru s měřením ESR. Měřitelné jsou kondenzátory s více než  $2\mu\text{F}$  až k  $50\text{mF}$ . Vzhledem k nízkému měřicímu napětí asi 300mV by mělo být ve většině případech možné měření v obvodu bez předchozího vypájení. Pokud je v souboru Makefile nastavena možnost `POWER_OFF`-, je počet měření omezen na 250, může být ale znovu spuštěn. Opakované měření může být ukončeno delším stiskem tlačítka.

**Měření odporů** Ikona 1——3 promění tester na ohmmetr mezi TP1 a TP3. Tento režim je indikován zobrazením textu [R] v pravém rohu prvního řádku displeje.

Protože se při této měřicí funkci nepoužívá ESR, platí pro odpory s hodnotou nižší než  $10\Omega$  rozlišení pouze  $0.1\Omega$ . Pokud je funkce ohmmetru nakonfigurovaná i s měřením indukci, zobrazí se zde ikona 1———3. Funkce ohmmetru poté zahrnuje měření indukčnosti pro odpory pod  $2100\Omega$ . V pravém rohu prvního řádku displeje se zobrazí text [RL]. Pokud nebyla detekována žádná indukčnost pro odpory pod  $10\Omega$  tak je použita ESR metoda měření. To zvyšuje rozlišení rezistorů s hodnotou nižší než  $10\Omega$  na  $0.01\Omega$ . V tomto měřicím režimu se měření opakuje bez stisku tlačítka. Stisknutím tlačítka opustíme tento režim a tester se vrátí do nabídky menu. Pokud je mezi TP1 a TP3 připojený odpor je tento měřicí režim také automaticky spuštěn stiskem tlačítka. Po stisku tlačítka se tester vrátí ke své normální funkci.

**Měření kondenzátorů** Ikona 1——3 mění tester na klasický měřič kondenzátorů na TP1 a TP3. Tento režim je označen znakem [C] v pravém rohu prvního řádku displeje. V tomto režimu mohou být kondenzátory měřeny od  $1\text{pF}$  do  $100\text{mF}$ . V tomto provozním režimu se měření opakuje bez stisku tlačítka. Stiskem tlačítka se tato speciální operace ukončí a tester se vrátí do nabídky menu. Stejně tak jako u měření odporů se tento provozní režim automaticky zapne, pokud byl mezi TP1 a TP3 detekován kondenzátor. Po stisku tlačítka se tester vrátí ke své normální funkci.

**Pulzní Enkodér** Pulzní enkodér lze testovat pomocí dodatečné funkce "Pulsní rotační snímač". Tři kontakty pulzního enkodéru libovolně připojíme ke třem zkušebním pinům před startem této doplňkové funkce. Po spuštění funkce nesmí být otočným knoflíkem otáčeno příliš rychle. Po úspěšném dokončení testu je na druhém řádku zobrazen symbol přiřazení kontaktů. Tester indikuje společný kontakt obou přepínačů a indikuje zda jsou v aretované poloze oba kontakty otevřené, ("o") nebo zavřené ("C"). Impulzní snímač s otevřenými kontakty v aretované pozici se zobrazí na řádce 2 „1-/-2-/-3 o" po dobu dvou sekund. Sa-

možřejmě je správné číslo pinu společného kontaktu zobrazeno uprostřed namísto "2". Dokonce i když je uzavřená poloha spínače v aretovaných pozicích, je také zobrazen na řádku 2, "1—2—3 C" po dobu dvou sekund. Neznám žádný pulsní snímač, který má vždy pouze uzavřené kontakty v každé pozici zámku. Polohy kontaktů mezi aretačními polohami se jen krátce ( $< 0,5s$ ) zobrazí bez kódových písmen "o" nebo "C" v 2 řádku. Pokud má být impulsní kódér použit k ovládání testeru, musí být v makefile volba `WITH_ROTARY_SWITCH=2` pro kódéry s pouze otevřenými kontakty ("o") a volba `WITH_ROTARY_SWITCH=1` pro snímače s otevřenými ("o") a uzavřenými ("C") kontakty v aretačních pozicích.

**C( $\mu F$ )-korekce** Pomocí této funkce lze měnit korekční hodnotu pro měření kapacit velkých hodnot. Stejnou korekci můžete také nastavit pomocí volby Makefile `C_H_KORR`. Hodnoty nad nulou snižují výstupní hodnotu kapacity o tuto procentuální hodnotu. Hodnoty pod nulou výstupní hodnotu zvyšují. Krátké stisknutí tlačítka snižuje korekční hodnotu o 0.1%, delší stisk tlačítka zvýší opravnou hodnotu o 0.1%. Velmi dlouhým stiskem tlačítka se hodnota uloží. Vlastností této metody měření je, že u nekvalitních elektrolytických kondenzátorů je naměřena kapacita výrazně vyšší než skutečná. Kvalitu lze rozpoznat parametrem `Vloss`. Kvalitní kondenzátory nemají žádný `Vloss`, nebo pouze 0,1%. Pro nastavení tohoto parametru je třeba použít pouze kondenzátory s vyšší hodnotou než  $50\mu F$  s vysokou kvalitou. Mimochodem, považuji za zbytečné, určit přesnou hodnotu kapacity elektrolytických kondenzátorů, protože kapacita závisí jak na teplotě, tak na výši stejnosměrného napětí.

**Autotest** Pomocí přídatné funkce "autotest" se provádí kompletní autotest s kalibrací. Všechny testovací funkce T1 až T7 (pokud tomu nebrání možnost `NO_TEST_T1_T7`) Pokaždé se provádí kalibrace s externím kondenzátorem.

**Napětí** Přídatná funkce "napětí" (měření napětí) je možná pouze tehdy, když je deaktivován UART výstup. nebo má ATmega nejméně 32 pinů (PLCC) a jeden z dalších pinů ADC6 nebo ADC7 je použit pro měření. Vzhledem k tomu, že je na portu PC3 (nebo ADC6 / 7) připojen dělič napětí 10:1, lze měřit napětí do hodnoty 50V. Připojený měnič DC-DC pro měření Zenerovy diody se zapíná stiskem tlačítka. Tak lze také měřit i připojené Zenerovy diody. Je-li v makefile volba `POWER_OFF-Option` nastavena a není-li stisknuto tlačítko, skončí měření po 4 minutách. Měření lze předtím ukončit dlouhým stiskem tlačítka ( $> 4$  vteřiny).

**Kontrast** Tato funkce je k dispozici řadičům se softwarovým řízením kontrastu. Nastavenou hodnotu lze snížit velmi krátkým stisknutím tlačítka nebo levým otočením impulzního snímače. Dlouhým stiskem tlačítka, nebo otáčením pulzního enkodéru ve směru hodinových ručiček se hodnota kontrastu zvýší. Pokud je tlačítko stisknuto déle, je funkce ukončena a nastavená hodnota je trvale zapsána do paměti EEPROM.

**Barva pozadí** Pro barevné displeje, může být tato položka zapnuta volbou Makefile `LCD_CHANGE_COLOR`, sloužící pro nastavení barvy pozadí. K tomu musí být nainstalováno rozšíření impulzního enkodéru. Můžete zvolit jednu ze tří barev červenou, zelenou a modrou pomocí delšího stisku tlačítka. Intenzitu vybrané barvy, označené znakem  $>$  ve sloupci 1, lze změnit otáčením enkodérem impulzů.

**Barva popředí** Pro barevné displeje, může být tato položka zapnuta volbou Makefile `LCD_CHANGE_COLOR`, pro úpravu barvy popředí. K tomu musí být nainstalováno rozšíření impulzního enkodéru. Můžete zvolit jednu ze tří barev červenou, zelenou a modrou pomocí delšího stisku tlačítka. Intenzitu vybrané barvy, označené znakem  $>$  ve sloupci 1, lze změnit otáčením knoflíku impulzů.

**Zobrazit údaje** Funkce "Zobrazit data" kromě údajů o verzi softwaru zobrazuje také údaje o kalibraci. Jedná se o přechodové odpory R0 kombinace pinů 1:3, 2:3 a 1:2. Také je změřen výstupní odpor měřících pinů proti 5V-(RiHi) a také proti 0V (RiLo). Také jsou zobrazeny hodnoty parazitní kapacity ( $C_0$ ) ve všech Pinových kombinacích (1:3, 2:3, 1:2 a 3:1, 3:2 2:1). Poté se také zobrazují korekce napětí komparátoru (`REF_C`) a pro referenční napětí (`REF_R`). U grafických displejů se zobrazí symboly použité pro součástky a font písma. Každá stránka se zobrazí 15 sekund. Na další stránku se také dostaneme stiskem tlačítka nebo otáčením enkodéru impulzů ve směru hodinových ručiček. Při otočení vlevo impulzního kódéru se zobrazení bude opakovat nebo přejdeme zpět na předchozí stránku.

**Vypnout** Pomocí dodatečné funkce "vypnout" se dá tester vypnout.

**Transistor** Samozřejmě že je možné s funkcí "tranzistor"(tranzistorový tester) vrátit zpět na normální funkci testeru tranzistorů.

Když je nastavena volba Makefile POWER\_OFF jsou všechny přídatné funkce z důvodu úspory baterie časově omezené.

## 4.2. Autotest a kalibrace

Je-li v software konfigurovaná funkce Autotestu a kalibrace, může se provést samočinný test zkratováním všech tří testovacích portů a stiskem tlačítka Start. Pro zahájení autotestu musí být během 2 sekund znovu stisknuto tlačítko start, jinak začne tester s normálním měřením. Autotest provádí testy popsané v kapitole 5 Transistor testeru [2]. Je-li tester konfigurován s funkcí menu (volba WITH\_MENU), provádí se úplný samočinný test automaticky jen při prvním použití a dále pak pouze během "autotestu", který lze vybrat jako funkci menu. Pro kalibraci jsou zkoušeny T1 až T7. Navíc se při volání funkce přes menu provádí nastavení s externím kondenzátorem. Normálně se provádí pouze při první kalibraci, tímto způsobem lze autotest se zkratovanými vstupy provádět rychleji. Čtyřnásobnému opakování testu na T1 až T7 je možné se vyhnout, pokud je trvale stisknuté tlačítko start. Takže můžete rychle ukončit nezájímavé testy a interaktivní testy můžete opakovat čtyřikrát uvolněním tlačítka start. Čtvrtý test pokračuje automaticky pouze, pokud je uvolněný zkrat mezi testovacími porty. Je-li v Makefile vybrána funkce AUTO\_CAL provede autotest kalibraci nulové hodnoty pro měření kondenzátorů. Pro tuto kalibraci je důležité, aby během čtvrté zkoušky bylo zrušen zkrat mezi testovacími piny. Během kalibrace (po zkoušce 6) byste se neměli dotýkat testovacích portů, nebo připojených kabelů. Měřicí kabely, by měly být stejné, které budou poté použity k měření. V opačném případě nebude vynulování správně provedeno. Při této volbě je kalibrace vnitřního odporu měřících portů provedena před každým měřením. Je-li v makefile nastavena funkce vzorkování volbou (WITH\_SamplingADC = 1) jsou během kalibrace provedeny navíc dva speciální kroky. Po normálním měření nulových hodnot kapacity budou změřeny také nulové hodnoty metodou odběru vzorků (C0samp). Poslední částí kalibrace je připojení zkušební kondenzátoru pro měření cívk mezi pinem 1 a 3 což je oznámeno požadavkem na vložení kapacity 1—||—3 10-30nF(L). Hodnota kapacity by měla být mezi 10nF a 30nF, k dosažení měřitelné rezonanční frekvence v pozdějším paralelním spojení s cívkou ( $< 2mH$ ). U cívk s indukčností větší než 2mH by měla být použita běžná zkušební funkce bez připojeného paralelního kondenzátoru. Paralelně připojený kondenzátor zde nezlepšuje výsledky měření. Po měření nulové kapacity je nezbytné připojit kondenzátor s kapacitou mezi 100nF a 20μF mezi Pin 1 a Pin 3. Z tohoto důvodu se v prvním řádku zobrazí požadavek na vložení kondenzátoru 1—||—3 >100nF. Kondenzátor byste měli připojit pouze po výstupu hodnot C0 nebo po zobrazení této hlášky. S tímto kondenzátorem je proveden offset analogového komparátoru, k určení přesnějších hodnot při měření kapacit. Navíc je tímto kondenzátorem také nastaven zisk ADC s vnitřním referencí k získání lepších výsledků při měření odporů s možností AUTOSCALE\_ADC. Pokud byla na testeru vybrána funkce menu (volba WITH\_MENU) a autotest nebyl spuštěn ve funkci menu, nastavení provede se kalibrace s externím kondenzátorem pouze při prvním zapnutí přístroje. Kalibrace s externím kondenzátorem se opakuje pouze v případě, že se provádí autotest funkcí menu. Offset pro měření ESR je přednastaven jako volba Option ESR\_ZERO v makefile. Při každém autotestu je nulová hodnota ESR znovu určena pro všechny tři kombinace měřících pinů. Metoda ESR měření se používá také pro odpory s hodnotami pod 10Ω, zde slouží k dosažení rozlišení 0,01Ω.

**5.0.1. Zapnutí** Dlouhým stisknutím při zapnutí se aktivuje režim automatického přidržování. V tomto režimu čeká tester na krátké stisknutí tlačítka k pokračování testování.

Jinak běží tester v nepřetržitém režimu. Výběr můžeš s (UI\_AUTOHOLD) v config.h obrátit.

Po zapnutí se krátce zobrazí verze firmwaru.

S velmi dlouhým stiskem tlačítka (2s) při zapnutí, se vrátí tester na uložené standardní hodnoty.

- To může být výhodné, když je např. kontrast LCD modulu tak nastaven, že nic nevidíš.

Pokud tester zjistí problém s uloženými hodnotami nastavení (problém s EEPROM), zobrazuje chybu kontrolního součtu a místo toho používá také výchozí hodnoty.

U testeru s vypínačem aktivuj POWER\_SWITCH\_MANUAL v config.h. V tomto případě to nemůže udělat tester sám.

**5.0.2. Hledání komponentů** Po zapnutí tester automaticky vyhledá komponenty.

V nepřetržitém režimu tester opakuje hledání po krátké čekací době.

Pokud není, několikrát za sebou, nalezeno žádné zařízení, tester se sám vypne.

V Auto Hold režimu (signalizováno kurzorem) provede tester jednu operaci a čeká, před zahájením dalšího vyhledávání, na stisknutí tlačítka nebo otočení rotačního kodéru doprava.

Čekací přestávka a automatické vypnutí v nepřetržitém režimu mohou být změněny prostřednictvím CYCLE\_DELAY a CYCLE\_MAX na straně 35 v config.h.

Režim automatického přidržování POWER\_OFF\_TIMEOUT (strana 36) má volitelné automatické vypnutí, které je aktivní pouze při vyhledávání komponentů, výstupu výsledku a v hlavní nabídce.

V obou režimech můžeš vyvolat hlavní nabídku (viz níže).

Pokud je k dispozici možnost bzučáku/pípáku, můžeš aktivovat krátký potvrzovací tón pro ukončení vyhledávání komponentů (UI\_PROBING\_DONE\_BEEP). Kromě toho je k dispozici také možnost dočasného přepnutí do režimu automatického podržení po rozpoznání součásti (v nepřetržitém režimu, UI\_AUTOHOLD\_FOUND). To pomáhá při čtení výsledného výstupu.

**5.0.3. Monitorování baterie** je možné nastavit podle tvých představ na stránce 43. Každý cyklus hledání součástek začíná zobrazením napětí baterie a jejího stavu (ok, slabý, prázdný). Při poklesu pod její prahové napětí tester vypne. Baterie je kontrolována i během provozu.

Výchozí konfigurace monitorování baterie je navržena pro 9V baterii, ale lze ji přizpůsobit v sekci „power management“ v config.h na jakékoli jiné napájení.

Monitorování baterií může být deaktivováno pomocí BAT\_NONE, přímé měření baterie menšího napětí než 5V, lze konfigurovat pomocí BAT\_DIRECT, nebo nepřímé měření pomocí děliče napětí (definovaného BAT\_R1 a BAT\_R2).

Ačkoli některé testery podporují volitelné externí napájení, neumožňují jeho sledování.

V tomto případě můžeš podle BAT\_EXT\_UNMONITORED problémy s automatikou, při nízkém stavu napětí, vypnout.

Při externím napájení je stav baterie nastaven na „ext“ (externí).

Mezní hodnoty pro slabou a prázdnou baterii se nastavují pomocí BAT\_WEAK a BAT\_LOW, zatímco BAT\_OFFSET definuje ztrátu napětí v okruhu, např. ochranná dioda s obrácenou polaritou a PNP tranzistor pro přepínání napájení.

Můžeš aktivovat zobrazení malé ikony baterie pro stav baterie místo textové varianty (UI\_BATTERY).

K dispozici je také možnost zobrazit stav baterie v posledním řádku za výstupem výsledků (UI\_BATTERY\_LASTLINE).



**5.0.4. Vypínání,** když při zobrazení výsledku posledního hledání součásti, stiskneš dlouze tlačítko, ukáže tester krátce „sbohem“ nebo „Ciao!“ a vypne se. Po dobu stisknutí zůstane však stále zapnutý. To je příčinou konstrukce obvodové části napájecího zdroje.

## 5.1. Výběrové menu

docílíš krátkým dvojitým stisknutím testovacího tlačítka po výstupu posledního výsledku. (Možná budeš muset na začátku trochu cvičit. ;-)

Máš-li rotační kodér, spustí se menu otočením vlevo.

S aktivováním UI\_SHORT\_CIRCUIT\_MENU (na stránce 35), lze také aktivovat starou metodu (zkratování tří zkušebních pinů).

V nabídce vybereš další položku krátkým stisknutím. Dlouhým stisknutím ji zvolíš.

U dvouřádkového LCD modulu je vpravo dole zobrazena navigační pomůcka. Šipka „>“, pokud následují další body, nebo „<“ u posledního bodu. Jdeš-li dále, dostaneš se na začátek.

Pro LCD modul s více než 2 řádky je vybraný bod označen “\*“.

Pokud máš rotační kodér, dosáhneš jeho otáčením předchozí nebo následující bod. I zde dochází k přetečení, tzn. od posledního k prvnímu bod.

Na rozdíl od předešle, zde vybírá položku **krátký stisk** tlačítka.

Po provedení funkce skočí tester normálně zpět k opětovnému vyhledávání součástek. Pokud chceš zůstat v nabídce, můžeš toto chování pomocí UI\_MAINMENU\_AUTOEXIT vypnut.

Nabídka se poté zavře až výslovným pokynem “Exit“.

Některé body/doplňky ukazují krátce na začátku rozložení použitých testovacích pinů.

Informace se zobrazí na několik sekund, ale můžeš ji krátkým stiskem tlačítka přeskočit.

Funkce, které generují signály, vysílají jejich signál standardně na pinu # 2.

Přitom budou piny # 1 a # 3 uzemněny.

Je tvůj tester nakonfigurován pro výstup signálu na svém vlastním výstupu (OC1B) nebudou zkušební piny použity.

**5.1.1. PWM-Generátor** dělá přesně to, co čekáš :-). Za předpokladu, že před přeložením firmwaru vybereš buď PWN generátor s jednoduchým nebo s rozšířeným ovládáním, který vyžaduje rotační kodér a větší display.

Rozložení testovacích pinů:

Pin # 2: výstup (přes 680Ω odpor k omezení proudu)

Pin # 1 a # 3: uzemněny

**5.1.2. Jednoduchý PWM** Nejprve musíš z daného seznamu vybrat možnou frekvenci. Krátké zmáčknutí volí další frekvenci, dlouhým stisknutím jí spustíš, stejně jako u volby menu.

S rotačním kódérem spouštíš krátkým stisknutím.

Pracovní cyklus začíná na 50 % a lze jej měnit v krocích po 5 %.

Krátký stisk pro +5 % a dlouhý pro -5 %.

Chceš-li program ukončit, stiskni dvakrát za sebou zkušební tlačítko.

Pokud je k dispozici rotační kodér, můžeš být pracovní cyklus měnit v krocích po 1 %.

**5.1.3. Rozšířený PWM** Krátkým stiskem klávesy se zde přepíná mezi frekvenčním a pracovním cyklem. Vybraná hodnota je označena hvězdičkou.

Točením rotačního kodéru vpravo zvolenou hodnotu zvyšuješ, doleva jí snižuješ.

Dlouhý stisk tlačítka zde obnoví výchozí hodnoty (frekvence: 1 kHz, pracovní cyklus: 50 %).

Dvojitým stisknutím tlačítka PWM generátor ukončíš.

**5.1.4. Obdélníkový signální generátor** vydává signál s čtvercovou vlnou s proměnnou frekvencí až do 1/4 MCU taktu (2MHz při 8MHz taktu). Počáteční frekvence je 1 kHz a pomocí otočného kodéru se nechá měnit. Stupeň změny určuje rotační rychlost, tzn. pomalé otáčení způsobí malé změny a rychlé otáčení velké.

Protože je generování signálu založeno na interní PWM funkci MCU, není možné generovat libovolné frekvence, ale pouze v krocích. Pro nízké frekvence je velikost kroku je poměrně malá, pouze při vysokých frekvencích se stává významnou.

Dlouhé stisknutí tlačítka vrátí frekvenci na 1 kHz a dvojitě stisknutí ukončí generátor.

Rozložení testovacích pinů:

Pin # 2: výstup (přes 680Ω odpor k omezení proudu)

Pin # 1 a # 3: uzemněny

**Poznámka:** *Poznámka: Nutný Rotační enkodér nebo jiná volba ovládání!*

**5.1.5. Zjištění Zenerového napětí (hardwarová úprava)** s pomocí DC-DC převodníku je možné generovat testovací napětí do 50V k testování Zenerových diod. Připojení se provádí pomocí svých vlastních testovacích pinů.

Dokud je testovací tlačítko stisknuto, generuje měnič testovací napětí a zobrazí proudové napětí.

Po uvolnění tlačítka se zobrazí nejmenší naměřené napětí, pokud běžel test dostatečně dlouho pro stabilní zkušební napětí. Tento proces lze libovolně často opakovat.

Pro ukončení stiskni krátce testovací tlačítko dvakrát po sobě.

Zesilovací měnič lze také přepínat přes pevný I/O pin (ZENER\_SWITCHED), k prodloužení výdrže baterie.

Pokud je tvůj tester vybaven pouze děličem napětí 10:1 bez zesilovacího převodníku pro měření externího napětí nebo bez měniče externího napětí, nebo je měnič boost v provozu po celou dobu, můžeš aktivovat alternativní režim (ZENER\_UNSWITCHED), který měří napětí pravidelně, bez stisknutí testovacího tlačítka. Pokud tester zobrazí kurzor v pravém dolním rohu mezi měřeními, můžeš deaktivovat zenerovo měření dvojitým stisknutím tlačítka.

Další možností je automatické spuštění Zenerova testu během běžného vyhledávání součástek. (HW\_PROBE\_ZENER) během běžného vyhledávání komponentů. Pokud není nalezena žádná komponenta na normálních testovacích vývodech, zkontroluje tester napětí na zenerových testovacích vývodech. Tato možnost je k dispozici pouze v případě, když je aktivován buď ZENER\_UNSWITCHED nebo ZENER\_SWITCHED.

V případě, že má tvůj tester nestandardní dělič napětí (ne 10:1), aktivuj ZENER\_DIVIDER\_CUSTOM a nastav hodnoty rezistorů (ZENER\_R1 a ZENER\_R2).  
Obvod pro Zenerovou diodu:

Pin +: katoda

Pin -: anoda

**5.1.6. Logický tester (hardwarová úprava)** kontroluje stav logických signálů pomocí pevného ADC- pinu plus děliče napětí. Dělič napětí by měl mít poměr 4:1, aby bylo možné dosáhnout napětí do až 20 V (15 V CMOS).

Vývod ADC je TP\_LOGIC (v config\_<MCU>.h) a dělič napětí je nastaven v config.h (LOGIC\_PROBE\_R1 a LOGIC\_PROBE\_R2).

Možná by nebylo špatné zlepšit ochranu vstupu, např. pomocí ochranných diod.

Po spuštění tester automaticky přečte napětí, porovná hodnoty napětí s prahovými hodnotami logických úrovní a poté zobrazí logický stav napětí:

0 pro nízký

1 pro vysoký

Z pro nedefinovaný/HiZ

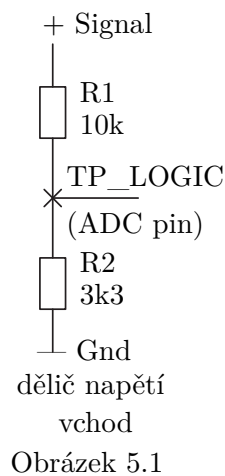
K dispozici je jednoduchá nabídka pro nastavení logické rodiny nebo Vcc/Vdd a prahů logických úrovní.

Pomocí testovacího tlačítka vyber požadované nastavení (označené hvězdičkou). Pak pomocí otočného ovladače můžeš změnit prostředí. Při změně logické rodiny nebo Vcc/Vdd jsou prahové hodnoty automaticky upraveny. Tyto lze měnit podle potřeby. V případě neobvyklého napětí Vcc/Vdd zvol vyšší možnost. Výchozí nastavení hodnoty pro Vcc/Vdd jsou:

- TTL : 5V

- CMOS : 3,3 V, 5 V, 9 V, 12 V, 15 V

A jako obvykle se dvěma krátkými stisky kláves tester logiky ukončí.



**5.1.7. tester continuity (hardwarová úprava)** kontroluje spojitost obvodů a vydává akustický signál prostřednictvím bzučáku a zároveň výstupní napětí mezi testovacími piny. Testovací proud je omezen na přibližně 7 mA pomocí  $R_l$  ( $680\ \Omega$ ). Prahové hodnoty pro bzučák jsou:

Napětí:	aktivní bzučák:	pasivní bzučák:
< 100 mV	nepřetržitý zvuk	opakované pípání s vysokou frekvencí
100-700 mV	opakované krátké pípnutí	opakované pípání s nízkou frekvencí
> 700 mV	žádný zvuk	

Krátké pípnutí signalizuje možný polovodičový přechod. V případě přerušení nebo velmi vysokého odporu se napětí blíží 5 V.

Po spuštění testeru spojitosti tester na několik sekund zobrazí zapojení testovacích kolíků. To lze přeskočit zmáčknutím tlačítka test.

Dvěma krátkými stisky kláves test ukončíš.

Zapojení zkoušecích vývodů:

Pin #1:	Vcc ( $680\ \Omega$ odpor pro omezení proudu)
Pin #3:	Zem

**5.1.8. ESR-Tool** (měřič ekvivalentního sériového odporu) může měřit a zobrazovat kondenzátor v obvodu a kromě kapacity měří také ESR pokud v obvodu kondenzátor najde.

**Před připojením se přesvědč, že je kondenzátor vybitý!**

Měřené hodnoty mohou mít odchylky způsobené paralelními komponenty v obvodu.

Pro zahájení měření krátce stiskni testovací tlačítko.

Chceš-li program ukončit, stiskni dvakrát za sebou zkušební tlačítko.

Obvod pro měření ESR:

Pin #1:	+
Pin #3:	-

**5.1.9. Unikající proud kondenzátoru** Test svodového proudu nabíjí kondenzátor a zobrazuje proud a napětí na měřicím odporu. Načítání začíná s  $R_l$  ( $680\ \Omega$ ) a přepne na  $R_h$  ( $470\ k\Omega$ ), jakmile dosáhne tok určitého limitu. Každý zkušební cyklus začíná zobrazením přiřazení zkušebních pinů. Po připojení kondenzátoru začíná ládování stisknutím testovacího tlačítka (nebo točením rotačního kodéru doprava).

Další stisknutí ukončí nabíjení a tester vybíjí kondenzátor a zobrazuje zbytkové napětí.

Po dosažení limitu vybití začne tester nový testovací cyklus.

Chceš-li test ukončit, stiskni dvakrát krátce testovací tlačítko.

Poznámka: **Věnuj pozornost polaritě Elkos!**

Zapojení kondenzátoru:

Pin #1:	+
Pin #3:	-

**5.1.10. R/C/L Monitor** Monitorové funkce měří neustále pasivní součástky na pinech #1 a #3. Po startu ukazuje tester na pár vteřin obsazení pinů, což je možné stisknutím tlačítka přeskočit.

Mezi měřeními je krátká dvouvteřinová přestávka což je označeno kurzorem vpravo dole.

Během pauzy lze monitor dvojitým stiskem testovací klávesy ukončit.

Dostupné monitory:

R-monitor	(odpor)
C-monitor	(kapacita plus volitelně ESR)
L-monitor	(indukce)
R/C/L-monitor	(odpor plus volitelně L, nebo C plus volitelně ESR)
R/L-monitor	(odpor plus volitelně indukce)

Pro monitory C a L existují možnosti pro uchování poslední platné hodnoty měření.

(SW\_MONITOR\_HOLD\_ESR, SW\_MONITOR\_HOLD\_L). Ta se zobrazí ve třetím řádku.

**Poznámka** k bit-bang SPI:

Hodnoty kapacit elektrolytických kondenzátorů mohou být o něco nižší než při normálním hledání součástek (*způsobené opakovaným měřením se stejnosměrným signálem*).

**5.1.11. L/C-Měřič (hardwarová úprava)** je založen na jednoduchém obvodu LC oscilátoru, jaký se používá u více levných PIC L/C měřičů. Obvyklý design ( $82\mu\text{H}$  a  $1\text{nF}$ ) má základní frekvenci kolem 595 kHz a připojení dalšího kondenzátoru nebo indukčnosti tuto frekvenci snižuje. S pomocí referenčního kondenzátoru známé hodnoty, měření frekvence a trochu matematiky je možné, hodnotu neznámého kondenzátoru určit, resp. lze vypočítat Indukčnost.

PIC-L/C měřiče mají obvykle měřicí rozsah od 10nH do 100mH, a 0,1pF až 900nF. Zdá se, že pro frekvenci používají dobu brány 100 ms.

m-Firmware zde ale používá automatický výběr rozsahu s časy brány 100 ms a 1000 ms ke zvýšení rozlišení malých hodnot L/C. Měřicí rozsah tedy začíná přibližně na 1 nH a 10 fF.

Rozsahy měření tedy začínají přibližně na 1nH a 10fF (0,01pF). Maximální měřitelná indukčnost je zhruba 150mH. Co se týče kapacity, narazil jsem na problém s mojí deskou s výstupním signálem od cca 33nF. Tam se na hranách objevují malé ostruhy, takže čítač frekvence vidí více pulsů, než kolik jich ve skutečnosti je. Jeden z uživatelů uvedl, že se jedná o známý problém obvodu oscilátoru založeného na LM311. Při testování po několika úpravách nedošlo k výraznému zlepšení. Proto, se nabízí, použít další komparátor nebo logické hradlo se Schmittovým spouštěčem pro vyčištění výstupního signálu oscilátoru. Čtyřnásobný CMOS NAND 4093 se k tomuto účelu skvěle hodí. S vyčištěným výstupním signálem, leží maximální kapacita přibližně u 120nF (nad touto hodnotou se LC oscilátor stává nestabilním). Tester omezuje měření na dolní hodnotu 10kHz, tj. teoretické maximální hodnoty jsou 250mH resp. 3,5fF, pokud by LC oscilátor pracoval stabilně.

Po spuštění L/C měřiče je první věcí, kterou tester provede, samo test, označený zprávou „samo-test ...“. Potom lze připojit kondenzátor nebo cívku k měření. Krátkým Stisknutím tlačítka přepínáte mezi měřením C a L (standard: měření C). Frekvence LC oscilátoru se v průběhu času poněkud mění (až do 100 Hz) a pak potřebuje novou úpravu. Pokud máte rostoucí nulovou hodnotu nebo vidíš „-“ bez připojené součástky, proveď automatické nastavení dlouhým stisknutím tlačítka. V případě, že nastal problém s párováním nebo se zruší stisknutím klávesy, tester opustí L/C metr a udává chybu. Dvě krátké stisknutí klávesy ukončí L/C metr.

**Tipy:**

- Referenční kondenzátor by měl být typu filmu 1nF s malou tolerancí. Můžeš ale také použít konvenční filmový kondenzátor kolem 1nF.

Měř dobrým LCR měřičem a podle toho změň `LC_METER_C_REF`.

- Pokud tě zajímá frekvence LC oscilátoru a jeho drift, pak aktivuj `LC_METER_SHOW_FREQ`.

**5.1.12. Čítač kmitočtů (hardwarová úprava)** je k dispozici ve dvou verzích.

Jednoduchý sestává z pasivního vstupu na pinu T0 MCU (F-in).

Rozšířený má kromě vstupní vyrovnávací paměti také dva oscilátory pro testování krystalu (pro nízké a vysoké frekvence) a další frekvenční děliče.

Oba okruhy jsou popsány v dokumentaci Karla-Heinze [2].

**5.1.13. Jednoduchý čítač** pokud je nainstalován přídatný obvod pro jednoduchý čítač kmitočtu, můžeš zjistit frekvence přibližně od 10 Hz až 1/4 taktu frekvence MCU s rozlišením na 1Hz při frekvencích pod 10 kHz. Frekvence se neustále měří a zobrazuje.

Automatické nastavení rozsahu nastavuje dobu brány na hodnoty mezi 10ms a 1000ms, v závislosti na frekvenci. Měření ukončíš dvojitým stiskem tlačítka.

**5.1.14. Rozšířený čítač** Počítadlo rozšířené frekvence má další předzesilovač, který povoluje měření vyšších frekvencí.

Teoretické maximum je 1/4 taktu MCU vynásobených prescalerem (16:1 nebo 32:1).

Ovládací signály jsou definovány v `config<mcu>.h`, a prosím nezapomeň v `config.h` na výběr správného předzesilovače.

Vstup signálu (vyrovnávací vstup, křemenný oscilátor pro nízké hodnoty frekvence,) nebo (křemenný oscilátor pro vysoké frekvence) změníš pomocí testovacího tlačítka nebo rotačního kodéru.

Počítadlo kmitočtů zastaví dva krátké stisky tlačítka.

**5.1.15. Kroužkový tester (hardwarová úprava)** (LOPT/FBT tester) kontroluje cívky a transformátory na přítomnost zkratu. Obvod generuje spouštěcí impuls a tester pak jednoduše počítá počet kmitů, který udává hodnotu Q. Obvod může být připojen přes pevný pin (RING\_TESTER\_PIN) nebo přes konektor testovacích kolíků (RING\_TESTER\_PROBES). V obou případech je kolík T0 ATmega vstupem čítače (počítá se při klesající hraně). Vývod T0 může být také použit paralelně k ovládání displeje.

Po spuštění kroužkového testeru se zobrazí přiřazení testovacích pinů, pokud je aktivováno ovládání prostřednictvím testovacích pinů. Poté tester automaticky kontroluje cívky/transformátory a zobrazuje počet kmitů. Jako obvykle, dvěma krátkými stisky kláves test ukončíte.

Interpretace počtu oscilací pro obvod s Darlingtonovým obvodem. na základě kroužkového testeru Boba Parkera:

Oscilace	Q
0	zkrat nebo rozpojení
1 - 3	nízké Q (špatné)
4 - 5	střední Q (nejasně)
$\geq 6$	vysoké Q (dobré)

Kromě výše uvedeného jednoduchého zapojení můžete použít i složitější varianty, pokud pracují s napětím přibližně 5 V, vyžadují malý proud ( $< 20 \text{ mA}$ ) a mají generovat počítací impulsy s klesající hranou.

Zapojení pro ovládání přes testovací kolíky:

Pin #1:	Vcc (5 V)
Pin #2:	impulsní výstup (s $680\Omega$ odporem pro omezení proudu)
Pin #3:	Zem
T0:	Vstup čítače

**5.1.16. Počítadlo událostí (s hardwarovou úpravou)** Čítač událostí používá pin T0 (F-in) jako pevný vstup a reaguje na náběžnou hranu signálu. Pin T0 není možné použít současně pro display. Doporučuje se jednoduchá úprava vstupu.

Čítač je řízen pomocí malého menu, které také hodnoty čítače zobrazí. Položky menu a jejich změna jsou vybírány krátkým stiskem, pomocí otočného kodéru nebo dalších tlačítek.

První položka nabídky je režim počítadla:

- Počítání            počítá čas a události
- Čas                počítá události za daný čas
- Události          počítají čas pro daný počet událostí

Druhá položka nabídky „n“ je počet událostí. V režimu počítadla „Události“ zobrazí hodnotu zastavení, kterou lze změnit. Dlouhým stisknutím nastavíš hodnotu stop na výchozí hodnotu (100). V jiných režimech počítadla je tato položka nabídky blokována.

Další nabídka je „t“ časový interval ve vteřinách (výchozí: 60 s). Stejná hra, pouze pro režim času.

Poslední položka nabídky spustí nebo zastaví čítač dlouhým stisknutím tlačítka. V době provozu je počet události a uplynulý čas každou vteřinu a poté co skončí měření aktualizován.

Časový limit je 43200s (12h) a pro události  $4 \cdot 10^9$ .

Jakmile je jedna z mezních hodnot překročena, počítadlo se automaticky zastaví.

Limit nebo hodnota zastavení událostí se kontroluje každých 200 ms. Proto pokud hodnota překročí 5 událostí/s, lze tuto hodnotu překročit.

- **Spouštěcí výstup** můžeš aktivovat s (EVENT\_COUNTER\_TRIGGER\_OUT) k ovládání dalšího zařízení pomocí zkušebních kolíků.

Výstup spouště je během počítání aktivován, tj. náběžná hrana při startu a klesající hrana při zastavení.

Zapojení výstupu:

Pin #1:	uzemění
Pin #2:	výstup (přes $680\Omega$ odpor k omezení proudu)
Pin #3:	uzemění

**5.1.17. Rotační kodér** testuje rotační enkodéry a určuje rozložení pinů. Tvým úkolem je připojit testovací piny k rotačnímu kodéru (A, B, Common) a točit enkodérem doprava (ve směru hodinových ručiček). Algoritmus vyžaduje pro detekci 4 kroky Grey kódu.

Směr otáčení je pro detekci A a B nutný, protože nesprávný směr by způsobil kroucení pinů. Když je rotační kodér detekován, vydá tester rozložení pinů a čeká (v automatickém režimu) na stisknutí tlačítka nebo (v nepřetržitém režimu) čeká chvíli.

Pro ukončení stiskni během vyhledávání krátce tlačítko testu.

**5.1.18. Kontrast** je možné, u některých grafických LCD modulů upravit.

Krátké stisknutí hodnotu zvýší, dlouhý stisk ji sníží. Pokud máš k dispozici otočný kodér, lze změnit hodnotu kontrastu točením. K ukončení, stiskni dvakrát za sebou zkušební tlačítko.

**5.1.19. Detektor/Dekodér pro IR dálkové ovládání** detekuje a dekoduje signály z IR ovladačů a vyžaduje IR přijímací modul, např. ze série TSOP.

Při překladu firmwaru si můžeš vybrat mezi dvěma variantami připojení.

V první variantě je modul připojen k normálním testovacím pinům (SW\_IR\_RECEIVER). Druhou variantou je pevný modul, připojený ke konkrétnímu MCU pinu (HW\_IR\_RECEIVER).

Pokud je známý protokol zjištěn, poskytne tester protokol, adresu (pokud je k dispozici), příkaz a případně hexadecimálně další informace.

Výstupní formát je: <Protokol> <Datová pole>

Pokud je datový paket vadný, ohlásí „?“.

Pokud je protokol neznámý, zobrazí tester počet pauz & pulsů a trvání prvního pulzu a první pauzy v jednotkách 50μs: ? <Pulse>: <první impuls> - <první pauza>

Pokud je počet pulzů na různých tlačítkách dálkového ovladače stejný, jedná se s největší pravděpodobností o PDM nebo PWM modulaci.

Měnící se počet pulzů naznačuje bi-fázovou modulaci.

K zastavení stiskni jednou testovací tlačítko.

Podporované protokoly a jejich datová pole:

- JVC <Adresa>: <příkaz>
- Kaseikyo (japonský kód, 48 bitů) <Code Vendor>: <System> - <Produkt>: <funkce>
- Matsushita (Panasonic MN6014, C6D6 / 12 bitů) <Code Zařízení>: <Data code>
- Motorola <příkaz>
- NEC (standardní a pokročilé) <Adresa>: <příkaz> R pro opakování sekvence
- Proton / Mitsubishi (M50560) <Adresa>: <příkaz>
- RC-5 (standardní) <Adresa>: <příkaz>
- RC-6 (standardní) <Adresa>: <příkaz>
- Samsung / Toshiba (32 bitů) <Code Zařízení>: <Data code>
- Sharp <Adresa>: <příkaz>
- Sony SIRC (12, 15 a 20 bitů) 12 & 15: <příkaz>: <adresa>  
20: <příkaz>: <adresa>: <rozšíření>

Volitelné protokoly (SW\_IR\_RX\_EXTRA):

- IR60 (SDA2008 / MC14497) <příkaz>
- Matsushita (Panasonic MN6014, C5D6 / 11 bitů) <Code Zařízení>: <Data code>
- NEC μPD1986C <Code dat>
- RECS80 (standardní a pokročilé) <Adresa>: <příkaz>
- RCA <Adresa>: <příkaz>
- Sanyo (LC7461) <Code Zařízení>: <key>
- Thomson <Zařízení>: <funkce>

Nosná frekvence přijímacího modulu TSOP IR nemusí přesně odpovídat dálkovému ovládání.

Ve skutečnosti pouze snižuje rozsah, což pro náš účel ale nepředstavuje problém.

Pokud máš k dispozici možnost bzučáku/pípáku, můžeš aktivovat krátký potvrzovací tón pro platné datové pakety (SW\_IR\_RX\_BEEP).

IR přijímací modul na testovacích pinech

- jako první připoj IR přijímací modul k IR detektoru dálkového ovládání!

Administrace pro modul TSOP:

Sonda # 1: uzemnění / Gnd  
Sonda # 2: Vs (680Ω omezovač proudu)  
Sonda # 3: Data/Out

Pokud je to nutné, lze firmware změnit na jiné vývody. by to bylo nutné. To je zejména u testerů s nulovými zásuvkami užitečné:

- alternativní zapojení vývodů SW\_IR\_RX\_PINOUT\_D\_G\_V

Sonda #1: Data/Out  
Sonda #2: Gnd  
Sonda #3: Vs (680Ω omezovač proudu)

- alternativní zapojení vývodů SW\_IR\_RX\_PINOUT\_D\_V\_G

Sonda #1: Data/Out  
Sonda #2: Vs (680Ω omezovač proudu)  
Sonda #3: Gnd

**Poznámky:** Odpor pro omezení proudu nastavuje IR přijímací modul s a předpokládá rozsah napájecího napětí asi 2,5 - 5V. Pokud máš 5V modul, můžeš na vlastní nebezpečí odpor v config.h deaktivovat. Zkrat však může MCU zničit.

- U pevného přijímacího modulu nastav port a data v config<MCU>.h

**5.1.20. IR dálkové ovládání** odešle kódy dálkového ovládání, které jsi dříve zadal a používá se k testování IR přijímačů nebo zařízení s IR dálkovým ovládáním.

Tato funkce vyžaduje další možnost vstupu, například např. rotační kodér, displej s více než čtyřmi řádky textu a jednoduchý obvod ovladače pro IR-LED.

Tester vám ukáže protokol, nosnou frekvenci, pracovní cyklus dopravce a několik datových polí.

Krátkým stisknutím testovacího tlačítka přepínáš tam a zpět mezi body.

Vybraný bod je označen znakem „\*“.

Pomocí otočného enkodéru (nebo jiné možnosti vstupu) měníš nastavení nebo hodnotu bodu.

Tester odesílá IR kód tak dlouho, jak je testovací tlačítko stisknuto. A jako obvykle, dvě krátké stisknutí tlačítka funkci zastaví.

Pokud změníš protokol, nastaví se nosná frekvence a pracovní cyklus na výchozí hodnoty příslušného protokolu.

Tyto můžeš ale libovolně změnit.

Nosnou frekvenci lze nastavit na 30 až 56 kHz a pracovní cyklus zapnout na 1/2 (50%), 1/3 (33%) nebo 1/4 (25%).

Datová pole jsou části kódu dálkového ovládání, které můžeš nastavit.

Jsou níže vysvětleny a většinou jde pouze o adresu a příkaz.

Podporované protokoly a jejich datová pole:

- JVC <Adresa: 8> <příkaz: 8>
- Kaseikyo (japonský kód) <Výrobce: 16> <Systém: 4> <Produkt: 8> <Funkce: 8>
- Matsushita (Panasonic, MN6014 12 bitů) <Zařízení: 6> <tlačítko: 6>
- Motorola <Command: 9>
- Norma NEC <Adresa: 8> <příkaz: 8>
- NEC Extended <Adresa: 16> <příkaz: 8>
- Proton / Mitsubishi (M50560) <Adresa: 8> <příkaz: 8>
- RC-5 standard <Adresa: 5> <příkaz: 6>
- RC-6 standard, režim 0 <Adresa: 8> <příkaz: 8>
- Samsung / Toshiba (32 bitů) <Zařízení: 8> <tlačítko: 8>
- Sharp / Denon <Adresa: 5> <příkaz: 8> <maskování: 1>
- Sony SIRC-12 <Příkaz: 7> <Adresa: 5>
- Sony SIRC-15 <Příkaz: 7> <Adresa: 8>
- Sony SIRC-20 <Příkaz: 7> <Adresa: 5> <Pokročilé: 8>

Volitelné protokoly (SW\_IR\_RX\_EXTRA):

- Thomson <Zařízení: 4> <funkce: 7>

Datová pole jsou oddělena mezerami a jejich syntaxe je: <Název pole>: <počet bitů>



Rozložení testovacích pinů:

Pin # 2: výstup (přes 680Ω odpor k omezení proudu)

Pin # 1 a # 3: uzemněny

Signální výstup (testovací pin # 2) má odpor pro omezení proudu a může proto spínat pouze asi 5 mA, což pro typickou IR LED s  $I_f$  100mA nestačí.

Obrázek 5.2 ukazuje ovladač, který lze pro IR-LED ( $V_f$  1,5V,  $I_f$  100mA) použít.

**Poznámka:**

*Pokud se načasování pulsů/pauzy nehodí, použij alternativní metodu čekající smyčky `SW_IR_TX_ALTDELAY` na straně 32. To může být nutné, pokud tvůj C kompilátor optimalizuje, přesto že je nastaven na zachování kódu vloženého v assembleru.*

**5.1.21. Test optických spojek** kontroluje optočlen a dává  $V_f$  LED, hodnotu CTR (také  $I_f$ ) a  $t_{on}$  a  $t_{off}$  časy (pro tranzistorové typy). Podporovány jsou standardní NPN tranzistory, NPN Darlington fáze a TRIAC. Pro CTR měření je MCU I/O pin, po dobu přibližně 3 ms, krátce přetížen. Datový list udává maximální výstupní proud 20 mA, pin je ale přetížen asi až na 100 mA. Proto je maximální hodnota CTR omezená a hodnoty nad 2000 % by měly být zpracovány s opatrností. Maximální proud pro LED je 5 mA, což by mělo být zvažováno u typů TRIAC.

Typy relé (MOSFET zády k sobě) jsou rozpoznány jako tranzistor a Hodnota CTR pak nemá smysl. Typy s antiparalelními LED budou ignorovány.

K testování potřebujete jednoduchý adaptér s následujícími třemi testovacími body:

typu tranzistoru:

- anoda LED
- Katoda LED a emitoru propojená tranzistorem
- Sběratel z tranzistoru

typu TRIAC:

- anoda LED
- Katoda LED a MT1 připojená pomocí TRIAC
- MT1 od TRIAC

Adaptér můžete libovolně spojit se třemi testovacími piny testeru. Tester pak automaticky najde přiřazení pinů. Po spuštění připoj adaptér k testovacím pinům a krátce stiskni tlačítko.

Pokud byla opto-spojka nalezena, zobrazí tester typ a různé informace.

Pokud ji nenajde, zobrazí se na displeji „žádný“.

Blikající kurzor označuje, že se při příštím testu očekává stisknutí tlačítka.

Dvě krátké stisknutí ukončí jako obvykle test.

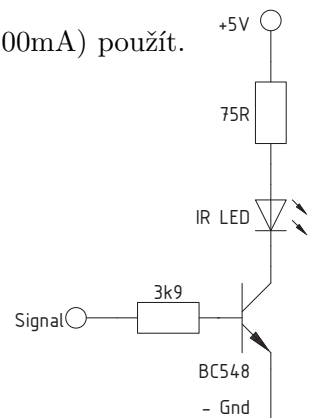
**5.1.22. Test fotodiod** ti umožní sledovat proud fotodiody. Nejprve se na několik vteřin zobrazí přiřazení testovacích kolíků, které lze předčasně ukončit stisknutím tlačítka. Poté tester přejde k pozorování proudu  $I_P$  s diodou v reverzním režimu, označeném "rev". Krátké stisknutí klávesy přepne provozní režim do přímého směru (fotočlánek), označeno "no". Další stisknutí klávesy přepne režim zpět. A jak jsi již jistě uhodl, dva krátké stisky tlačítka ukončí testovací funkci. Chceš-li fotodiodu otestovat, tak jí dočasně ručně zatemni nebo ji z různých vzdáleností ozáří vhodným zdrojem světla. Méně světla vede k menšímu proudu a naopak. Je možné, že tester může zachytit nějaké EM rušení, které může vést k malému proudu, zvláště v režimu obráceného směru. To lze velmi dobře pozorovat, pokud není fotodioda připojena. Je třeba také zdůraznit, že fotodiody mají v režimu zpětného vychýlení temný proud.

**Upozornění:** Nezkoušej solární články!

Zapojení testovacích vývodů:

Pin #1: anoda

Pin #3: katoda



Obrázek 5.2.  
IR-ovladač



**5.1.23. Rychlý test diod/LED** je určen k testování diod/LED a určení jejich polarity na testovacích pinech #1 a #3. Během testování LED diody blikají. Po spuštění tester na několik sekund zobrazí zapojení testovacích vývodů, které lze přeskočit stisknutím testovacího tlačítka. Pokud je dioda nebo LED dioda nalezena, zobrazí se její přiřazení vývodů a Vf. Anoda je vždy vlevo a katoda vpravo. V případě dvou antiparalelních diod/LED je zobrazena i druhá z nich. A jak již víš: k dokončení stiskni dvakrát krátce testovací tlačítko.

**5.1.24. Test servopohonů pro modely** Tato funkce generuje PWM signál pro serva pro výrobu modelů, která jsou ovládána s PWM 1–2 ms dlouhými pulsy.

Podporovány jsou typické PWM frekvence 50, 125, 250 a 333 Hz, s nastavitelnou délkou pulsu od 0,5 až do 2,5 ms.

Kromě toho existuje režim rozmítání pro impulzy 1 - 2ms s volitelnou rychlostí. Šířku pulzu nastavíš pomocí otočného kodéru. Doleva pro kratší pulsy, doprava pro delší.

Dlouhým stisknutím tlačítka se šířka pulzu nastaví na (střední polohu serva), to je na 1,5 ms. Krátkým stiskem tlačítka přepínáš mezi výběrem pulzu a frekvencí. (označené hvězdičkou).

Ve volbě frekvence přepínáš rotačním kóděrem mezi kmitočty. Dlouhé stisknutí zapne nebo vypne „Sweep-mod“ (označený „<->“). Pokud je „Sweep-mod“ zapnutý, je délka impulsu nahrazena „Sweep“ časem, který lze změnit pomocí otočného kodéru. Funkci jako obvykle zastaví dvojitý stisk tlačítka.

Rozložení testovacích pinů:

Pin # 2: PWM výstup (přes 680Ω odpor k omezení proudu)

Pin # 1 a # 3: uzemněny

**Poznámka:** Servo potřebuje svoje napájení.

Výrobce	Pin 1	Pin 2	Pin 3
Airtronics	PWM bílá/černá	Gnd černá	Vcc červená
Futaba	PWM bílá	Vcc červená	Gnd černá
hitec	PWM žlutá	Vcc červená	Gnd černá
JR Radios	PWM Oranžová	Vcc červená	Gnd hnědá

Tabulka 5.1. Rozložení pinů pro typické 3kolíkové servopohony

**5.1.25. OneWire skenování** zobrazuje ROM-kódy, všech připojených uživatelů. Informace o nastavení sběrnice OneWire naleznete v části „Sběrnice a rozhraní“ na straně 40. Při použití testovacích kolíků tester informuje o zapojení a vyčkávání, dokud není detekován externí pull-up odpor. To lze přeskočit stisknutím tlačítka.

Po každém stisknutí tlačítka tester vyhledá dalšího účastníka sběrnice a vydá jeho ROM-kód (v šestnáctkové soustavě). První část vydání je Rodinný kód a druhá sériové číslo.

Hodnota CRC je vynechána.

U Rodinovského kódu  $\geq 0x80$  (nastavený 7bit) se jedná o zákaznický kód, ve kterém jsou ty horní (levé) tři číslice sériového čísla jeho zákaznické ID.

Tester tě informuje, když našel posledního účastníka sběrnice, ale také o CRC chybách i o chybách sběrnice.

V případě posledního účastníka sběrnice nebo chyby sběrnice, můžeš spustit úplně nové skenování stisknutím tlačítka. Zapojení je na straně 40.

Funkci ukončí, jako obvykle, dvě krátké stisknutí.

**5.1.26. Snímače teploty DS18B20, DS18S20** v tomto výběru lze použít tyto teplotní OneWire senzory ke čtení teplot. Nastavení sběrnice je na stránce 40, viz část „Busse“. Při použití testovacích pinů tester informuje o zapojení a čeká dokud není detekován externí pull-up odpor. To lze přeskočit stisknutím tlačítka. Po připojení DS18B20, DS18S20 jako jediného klienta na sběrnici, se stisknutím tlačítka začne číst teplota (která může trvat téměř sekundu).

Dlouhým stisknutím tlačítka volíš auto-mód (automatická aktualizace), která je signalizována hvězdičkou “\*” v prvním řádku.

Pro ukončení stiskni krátce dvakrát testovací tlačítko.  
Připojení zkušebních pinů:

Probe #1: Gnd  
Probe #2: DQ (Data)  
Probe #3: Vcc (Proud je ohraničen (přes 680Ω odpor)

**Nezapomeň:** *Kromě toho je 4k7Ω odpor, mezi DQ a Vcc nutný.*

Nápověda: - Možnost pro DS18B20: zaokrouhlení na 0,1 °C/F (UI\_ROUND\_DS18B20)  
- Možnost pro DS18S20: vysoké rozlišení (DS18S20\_HIGHRES)  
- Funkce DS18B20 dokáže snímat také čidlo DS1822.

**5.1.27. Senzory teploty a vlhkosti DHTxx** Pro čtení DHT11, DHT22 a kompatibilních snímačů teploty a vlhkosti. Nejprve tester ukáže, že testovací kolíky jsou připojeny a čeká na externí pull-up odpor. Poté se zobrazí vybraný typ senzoru (Standard: DHT11), který je načten, krátkým stisknutím testovacího tlačítka. Pokud je čtení úspěšné, vydá tester naměřené hodnoty, v případě chyby jen "-". Jedním dlouhým stisknutím změníš typ senzoru a dvě krátké stisknutí tlačítka ukončí funkci. Při změně typu senzoru máš možnost aktivovat automatický režim čtení (každou sekundu). Toto je za názvem senzoru označeno "\*". Podporované senzory:

DHT11: DHT11, RHT01  
DHT22: DHT22, RHT03, AM2302  
DHT21, RHT02, AM2301, HM2301  
DHT33, RHT04, AM2303  
DHT44, RHT05

Připojení zkušebních pinů:

Probe #1: Gnd  
Probe #2: Data  
Probe #3: Vdd (Proud není ohraničen)

Mezi Data (#2) a Vdd (#3) je vyžadován externí pull-up odpor 4k7Ω!

Některé moduly již integrovaly 10kΩ pull-up odpor, který také s kratšími kabely dobře funguje.

**Poznámka:**

*Vnitřní 680Ω testovací odpor nelze k omezení proudu použít, kvůli aktuální spotřebě senzoru. Buďte opatrní, zkrat může poškodit MCU.*

#### 5.1.28. Převodník termočlánků MAX6675/MAX31855

Oba MAX jsou termočláňkové převodníky s kompenzací studeného spoje a jsou řízeny prostřednictvím sběrnice SPI. MAX6675 je určen pouze pro termočláňky typu K, zatímco MAX31855 je k dispozici pro více typů. MAX31855 má provozní napětí 3,3 V, a proto vyžaduje převodníky úrovní.

Obsluha obou MAX je stejná. Stisknutím testovacího tlačítka se spustí měření a zobrazí se teplota. V případě chyb se místo toho zobrazí symbol "-". Pro režim automatického odečítání (každou sekundu) stiskni a podrž testovací tlačítko déle. (označeno symbolem "\*" za názvem MAX). Dalším dlouhým stisknutím tlačítka se režim automatického čtení opět ukončí.

Dvě krátká stisknutí klávesy funkci ukončí.

**Poznámky:**

- Přizpůsob MAX31855\_CS v config\_<MCU>.h!
- Potřebuje také funkci čtení SPI (SPI\_RW).

**5.1.29. BH1750 Snímač okolního světla** tento levný snímač okolního osvětlení pracuje s I2C sběrnici a 3,3 V provozním napětím. Díky 3,3V regulátoru napětí a pull-up rezistorům pro SCL a SDA (na 3,3 V) lze obvykle provozovat přímo na 5V ATmega pokud na sběrnici nezpůsobují žádné další obvody I2C problémy. V opačném případě je nutné použít převodníky úrovní.

Pro měření stiskni krátce testovací tlačítko. Případně můžeš dlouhým stisknutím tlačítka přepnout do automatického režimu, v kterém se objeví symbol “\*” v titulkovém pruhu; a zase zpět. V automatickém režimu se měření provádí každou vteřinu. V případě chyby se na displeji testeru místo světelné hodnoty zobrazí“-“. intenzita.

A jako obvykle se funkce ukončí dvěma krátkými stisky tlačítka.

**Poznámky:**

- Moduly BH1750 mají obvykle 3,3V LDO a 4,7kΩpull-up rezistory ( na 3,3 V) pro SCL a SDA.
- “Error“ přímo za voláním funkce znamená problém se senzorem, např. žádná komunikace I2C.

**5.1.30. Svítidla** Jedná se o obecný spínací výstup pro ovládání dalšího obvodu, jako je například LED světlo. Zkoušečka jednoduše přepíná výstup mezi nízkou a vysokou hodnotou při každém volání. Zatížení menší než 20 mA mohou být ovládány přímo.

**5.1.31. Autotest** Pokud jsi autotest spustil pomocí nabídky, vyzve tě tester ke zkratování zkušebních pinů a čeká, až je rozpozná.

V případě problémů můžeš čekání stiskem klávesy přerušit.

Autotest provádí každý test 5krát.

Krátké stisknutí tlačítka přeskočí aktuální test a dlouhý stisk tlačítka kompletní test.

V testu # 4 musí být zkrat odstraněn. Tester v tomto kroku tak dlouho čeká.

Kroky testu jsou:

- T1 interní referenční napětí (v mV)
- T2 Srovnání odporů Rl (offset v mV)
- T3 Srovnání Rh odporů (offset v mV)
- T4 Odstraň zkrat zkušebních pinů/kabelu
- T5 Test těsnosti pro zkušební piny s úrovní Gnd (napětí v mV)
- T6 Test těsnosti pro zkušební piny s úrovní Vcc (napětí v mV)

**Poznámky:**

- Pokud se hodnoty T2 liší o více než 5 mV mezi dvojicemi testovacích pinů, může to být způsobeno nestojnými testovacími odpory Rl.
- Totéž platí pro hodnoty T3 a testovací odpory Rh.
- Dlouhé zkušební kabely mají negativní vliv a zhoršují hodnoty u T2, T3, T5 a T6.
- Podivné hodnoty jsou obvykle známkou vadných pinů MCU, zkratů, znečištění nebo větších problémů s kontakty.

**5.1.32. Samočinné nastavení** měří odpor a kapacitu měřicích kabelů, tzn. z desky s obvodu, vnitřního zapojení a měřicího kabelu jako součet k určení nulového posunu.

Také je určen vnitřní odpor pinů MCU portů v režimu pull-up a pull-down.

Pokud je srovnání přeskočeno nebo pokud jsou změřené hodnoty nepravděpodobné, převezme tester výchozí hodnoty firmwaru.

Pokud běží všechno hladce, zobrazí se nové hodnoty, které ale **nebudou** automaticky do EEPROMu uloženy. (viz volbu „Uložit“).

Během měření kondenzátoru (při normálním vyhledávání součástí) se automaticky stane kompenzace napětí analogového komparátoru, pokud má Kondenzátor hodnotu mezi 100nF a 3,3μF.

Kromě toho se současně měří offset vnitřní referenční hodnoty napětí.

Před provedením automatického ladění bys měl min. 3krát za sebou změřit filmový kondenzátor s kapacitou mezi 100nF a 3,3μF, aby bylo možné, uvedenou kompenzaci určit.

První měření je obvykle příliš nízké, druhé příliš vysoké a až od třetího měření dosáhneš správnou hodnotu.

To je způsobeno offsetovými kompenzacemi.

V modelech s pevným kondenzátorem pro samo-ladění, je automatické seřízení pro měření

kapacity, nahrazeno vlastní funkcí, která během testu provede se vlastní nastavení.

Zde nemusíš žádný filmový kondenzátor měřit.

Pokud se kompenzace kapacity mezi páry testovacích pinů příliš liší, můžeš v config.h přepnout na specifické posuny podle právě zapojených zkušebních kabelů-párů (CAP\_MULTIOFFSET). To stejné platí i pro jejich odpor s (xR\_MULTIOFFSET) viz stana 39.

Samo-ladění je do značné míry autotestu provozem a obsluhou podobné.

Kroky seřízení jsou:

- A1 kompenzace pro interní referenční napětí a analogový komparátor  
(pouze s pevným vyrovnávacím kondenzátorem)
- A2 Odolnost zkušebních pinů/kabelů v (10 mΩ)
- A3 Odstranění zkratu zkušebních pinů/kabelů
- A4 vnitřní odpor portových pinů pro Gnd (napětí přes RiL)
- A5 vnitřní odpor pinů portu pro Vcc (napětí přes RiH)
- A6 kapacita testovacích pinů/kabelů (v pF)

Povolené maximální hodnoty:

- |   |                       |
|---|-----------------------|
| - zkušební kolík/kabel odpor                  | <1,50 Ω (dva v sérii) |
| - Zkušební pin/kabel kapacita                 | <100 pF               |
| - Vnitřní odpor IO pinu v nízkém režimu (RiL) | < 25 Ω                |
| - Vnitřní odpor vývodu IO v režimu high (RiH) | < 29 Ω                |

**Poznámky:**

- Pokud se hodnoty odporu zkušebních kolíků příliš liší, je možný kontaktní problém.
- Pokud je posunutí kapacity (A6) nulové, je to pravděpodobně způsobeno problémem s vybíjením. (zvýšit CAP\_DISCHARGED).

**Pamatuj:**

**Nastavení není kalibrace!**

- Kalibrace je postup pro porovnání výsledků měření se sledovatelnými standardy a odchylky zaznamenat. Účelem je sledovat a odstraňovat časové odchylky.
- Nastavení je postup nastavit měřicího zařízení tak, aby dodržovalo svou danou přesnost a další parametry.

**5.1.33. Uchovat/Použít** Po vlastní úpravě můžeš pomocí této funkce aktualizovat hodnoty nastavení, uložené v paměti EEPROM. Při příštím restartu testeru se tyto hodnoty (profil č. 1) pak automaticky načtou a použijí.

Pro usnadnění jsou k dispozici dva profily pro uložení nebo načtení, např. pro dvě různé sady měřicích kabelů. Pokud jsou dva příliš málo lze aktivovat třetí profil (UI\_THREE\_PROFILES).

Myšlenka funkce manuálního ukládání je taková, že když dočasně změníš měřicí kabely a provedeš samočinné nastavení, tak máš po restartu opět hodnoty pro hlavní měřicí kabely. Jinak bys musel své standardní kabely znovu nastavit.

Volitelně můžeš, pomocí (UI\_CHOOSE\_PROFILE) při zapnutí testeru toto menu ukázat.

**5.1.34. Ukázat hodnoty** Tato funkce zobrazuje aktuální hodnoty nastavení.

Použití externí reference napětí 2.5V je signalizováno „\*“ po Vcc.

**5.1.35. Znaková sada/symboly** Tyto dvě položky nabídky vypisují kompletní sadu znaků nebo symboly složek pro testovací účely. symboly pro testovací účely. Řádky/bloky začínají adresou (v šestnáctkové soustavě). z prvního znaku/symbolu v daném řádku/bloku. Následuje buď 8 znaků nebo tolik symbolů, kolik se jich vejde do bloku.

**5.1.36. Vypnout** Za předpokladu, že jsi tuto funkci přes SW\_POWER\_OFF strana 43 aktivoval, můžeš zde tester vypnout.

**5.1.37. Konec** ti umožní opustit nabídku, když jsi do ni náhodou/nechtě vstoupil.

## 5.2. config.h

Tento soubor slouží k nastavení provozu a funkcí. Protože se zde jedná o normální soubor se záhlavím C, používají se zde, na rozdíl od „Makefile“ známá pravidla komentování v C. Chceš-li něco aktivovat, odstrañ znaky „//“ na začátku řádku a na deaktivování je zase na začátek řádku vlož. Některá nastavení vyžadují číselnou hodnotu, kterou můžeš případně upravit.

### 5.2.1. Hardwarová obsluha

#### rotační kodér k obsluze

- Standardní Piny: PD2& PD3 (ATmega 328)
- mohou být paralelně použity s LCD modulem
- viz ENCODER\_PORT pro portové piny (config-<MCU>.h)
- #define HW\_ENCODER \* - komentuj k odvolbě

#### počet pulzů Gray kódu na krok nebo aretaci

- rotační kódovací impuls je kompletní sekvence 4 Gray kódových impulsů
- typické hodnoty: 2 nebo 4, zřídka 1
- #define ENCODER\_PULSES...4 - Uprav hodnotu podle tvého rotačního kodéru.

#### počet zářezek nebo kroků

- používá se k záznamu rychlosti otáčení rotačního kodéru
- nemusí přesně odpovídat a umožňuje jemné doladění (vyšší hodnota: pomalejší, nižší hodnota: rychlejší)
- typické hodnoty: 20, 24 nebo 30
- #define ENCODER\_STEPS...24 - Uprav hodnotu podle tvého rotačního kodéru

#### více/méně tlačítek pro ovládání

- alternativa k rotačnímu kodéru
- viz KEY\_PORT pro portové piny (config-<MCU>.h)
- //#define HW\_INCDEC\_KEYS - odkomentuj k aktivaci

#### referenční napětí 2.5 V pro Vcc test

- standardní pin:PC4 (ATmega 328)
- mělo by být nejméně 10krát přesnější než regulátor napětí
- viz TP\_REF pro port pin (config-<MCU>.h)
- v případě potřeby uprav UREF\_25 níže podle referenční hodnoty napětí
- //#define HW\_REF25 - odkomentuj k aktivaci

#### typické napětí referenčního napětí 2,5 V (v mV)

- viz katalogový list reference napětí
- nebo změř napětí > = 5,5 číslic DMM
- #define UREF\_25...2495 \*\* - Pokud je třeba změnit hodnotu

#### ochranná relé pro vybíjení kondenzátorů

- standardní pin:PC4 (ATmega 328)
- nízký signál: kratuj zkušební piny
- vysoký signál prostřednictvím externí reference: eliminuj zkrat
- //#define HW\_DISCHARGE\_RELAY \* - odkomentuj k aktivaci

#### měření napětí do 50V DC, měření zenerových diod

- standardní pin:PC3 (ATmega 328)
- dělič napětí (standart je 10:1)
- DC-DC boostovací převodník ovládaný testovacím tlačítkem
- viz port TP\_BAT pro portový port
- //#define HW\_ZENER - odkomentuj k aktivaci

#### nestandardní dělič napětí při testu zenerových diod

- standardní dělič napětí je 10:1
- ZENER\_R1: horní odpor v  $\Omega$
- ZENER\_R2: spodní odpor v  $\Omega$
- //#define ZENER\_DIVIDER\_CUSTOM - odkomentuj k aktivaci
- #define ZENER\_R1 180000 \*\* - Pokud je třeba změnit hodnotu
- #define ZENER\_R2 20000 \*\* - Pokud je třeba změnit hodnotu

### alternativní mód při testu zenerových diod

- zesilovací převodník nespíná, pokud je DC/DC krokový převodník stále v chodu
  - při měření externího napětí (přepínání bez stupňového převodníku)
- ```
//#define ZENER_UNSWITCHED
```
- odkomentuj k aktivaci

### alternativní režim pro Zenerovu kontrolu: nepřepínat měnič boost

- . \* - pokud je DC-DC měnič v provozu po celou dobu nebo pouze měření vnějšího napětí (obvod bez měniče)
- ```
//#define ZENER_UNSWITCHED
```
- pro povolení

### alternativní režim pro Zenerovu kontrolu přepínání převodníku přes vyhrazený MCU pin

- posilovací měnič je řízen vyhrazeným I/O pinem.
  - viz BOOST\_PORT v config\_<MCU>.h pro pin portu
  - dvě kontrolní metody:  
ZENER\_BOOST\_HIGH / aktivuje se při vysoké hodnotě  
ZENER\_BOOST\_LOW / aktivuje se při nízké hodnotě
  - odstranění komentáře pro aktivaci a výběr způsobu řízení
- ```
//#define ZENER_SWITCHED  
//#define ZENER_BOOST_HIGH /* vysoká aktivita */  
#define ZENER_BOOST_LOW /* nízká aktivita */
```

### Zenerova kontrola při běžném sondování

- vyžaduje neustále běžící měnič (ZENER\_UNSWITCHED)
  - minimální/maximální napětí jsou určena pro detekci platného Zenerova napětí.  
min. napětí by mělo být vyšší než šumové dno, zatímco max. napětí by mělo být nižší než výstupní napětí zesilovacího měniče.
- ```
//#define HW_PROBE_ZENER
```
- odkomentuj k aktivaci
- 
- ```
#define ZENER_VOLTAGE_MIN 1000 /* min. napětí v mV */
```
- \*\* - změň hodnotu
- 
- ```
#define ZENER_VOLTAGE_MAX 30000 /* max. napětí v mV */
```
- \*\* - změň hodnotu

### výstup pevného signálu

- pokud je pin OC1B MCU zapojen jako vyhrazený výstup signálu namísto odporu R<sub>l</sub> zkušebního pinu #2
- ```
//#define HW_FIXED_SIGNAL_OUTPUT
```
- \* - odkomentuj k aktivaci

### jednoduchý čítač kmitočtů

- standardní pin:T0 (PD4 ATmega 328) přímo jako frekvenční vstup
  - počítá až do 1/4 taktu frekvence MCU
  - lze ho použít paralelně s LCD modulem
- ```
//#define HW_FREQ_COUNTER_BASIC
```
- \* - odkomentuj k aktivaci

### rozšířený čítač kmitočtů

- nízko a vysokofrekvenční krystalový oscilátor a vyrovnávací frekvenční vstup
  - Prescaler 1:1 a 16:1 (32:1)
  - piny portů najdeš v části COUNTER\_PORT (config-<MCU>.h)
  - vyžaduje LCD s více než 2 řádky textu
  - nastavení předvolebního nastavení obvodu: buď 16:1 nebo 32:1
- ```
//#define HW_FREQ_COUNTER_EXT
```
- odkomentuj k aktivaci
- 
- ```
#define FREQ_COUNTER_PRESCALER...16 / * 16:1 * /
```
- volba
- 
- ```
//# define FREQ_COUNTER_PRESCALER...32 / * 32:1 * /
```

### ring tester (LOPT/FBT tester)

- používá T0 přímo jako vstup čítače
  - výběr pulzní výstup: buď vyhrazený pin nebo sondy
  - viz RINGTESTER\_PORT v config-<MCU>.h pro vyhrazený pin
- ```
//#define HW_RING_TESTER
```
- odkomentuj k aktivaci
- 
- ```
#define RING_TESTER_PIN /* dedicated pin */
```
- volba
- 
- ```
//#define RING_TESTER_PROBES /* probes */
```

### počítadlo událostí

- standardní pin: T0 (PD4 ATmega 328)
- používá T0 přímo jako vstup události/pulsu (náběžná hrana)
- není možný žádný společný provoz s displeji pro T0
- vyžaduje další tlačítka (např. otočný kodér) a displej s více než 5 řádky
- pouze pro MCU takt 8, 16 nebo 20 MHz
- //# definiere HW\_EVENT\_COUNTER - odkomentuj k aktivaci

### spouštěcí výstup pro čítač událostí

- jako spouštěcí výstup se používá pin #2, piny #1 a #3 jsou Gnd
- nastaví spouštěcí výstup na vysokou hodnotu během počítání
- //# definiere EVENT\_COUNTER\_TRIGGER\_OUTR - odkomentuj k aktivaci

### IR detektor/dekodér (prostřednictvím vyhrazeného pinu MCU)

- vyžaduje IR modul přijímače, např. série TSOP
- modul je připojen k pevnému I/O pinu
- viz PIN\_PORT pro port pin (config-<MCU>.h)
- pro další protokoly aktivuj SW\_IR\_RX\_EXTRA
- //#define HW\_IR\_RECEIVER - odkomentuj k aktivaci

### pevný kondenzátor pro samonastavení

- viz TP\_CAP a ADJUST\_PORT pro piny portů (config-<MCU>.h)
- //#define HW\_ADJUST\_CAP - odkomentuj k aktivaci

### L/C měřič

- používá T0 přímo jako frekvenční vstup
- viz LC\_CTRL\_PORT v config-<MCU>.h pro Port-piny
- //#define HW\_LC\_METER - odkomentuj k aktivaci

### L/C měřič Hodnota referenčního kondenzátoru C<sub>p</sub> (v 0,1 pF)

- měl by mít kolem 1000pF
- #define LC\_METER\_C\_REF 10000 - komentuj k deaktivaci

### L/C měřič ukazuje také frekvenci LC-Oscilátoru

- pomůže rozpoznat úchylku frekvence Oscilátoru
- vyžaduje displej s více než dva řádky
- //#define LC\_METER\_SHOW\_FREQ - odkomentuj k aktivaci

### paralelní relé kondenzátoru (vzorkovací ADC)

- //# define HW\_CAP\_RELAY - odkomentuj k aktivaci

### logická sonda

- viz TP\_LOGIC v config-<MCU>.h pro vyhrazený port pin
- používá dělič napětí (standard: 4:1, R1=10k, R2=3,3k, až 20V)
- LOGIC\_PROBE\_R1: horní odpor v ohmech
- LOGIC\_PROBE\_R2: spodní odpor v ohmech
- vyžaduje přídatná tlačítka (např. otočný snímač) a displej s více než 4 řádky
- //#define HW\_LOGIC\_PROBE - odkomentuj k aktivaci
- #define LOGIC\_PROBE\_R1 10000 - zkontroluj nastavení hodnot odporů
- #define LOGIC\_PROBE\_R2 3300 - zkontroluj nastavení hodnot odporů

### bzučák

- viz BUZZER\_CTRL v config-<MCU>.h pro port pin
- typy bzučáků:
- //#define HW\_BUZZER - odkomentuj k aktivaci
- #define BUZZER\_ACTIVE /\* aktivní bzučák s oscilátorem/
- //#define BUZZER\_PASSIVE /\* pasivní bzučák \*/ - zvol tvůj

### MAX6675 převodník termočlánku

- viz MAX6675\_CS v config-<MCU>.h pro vyhrazený pin portu
- vyžaduje sběrnici SPI a podporu čtení SPI
- //#define HW\_MAX6675 - odkomentuj k aktivaci

### **MAX31855** převodník termočlánků

- viz MAX31855\_CS v config\_<MCU>.h pro vyhrazený pin portu
- vyžaduje sběrnici SPI a podporu čtení SPI
- //#define HW\_MAX31855 - odkomentuj k aktivaci

### **LED lampa** obecný spínací výstup pro volitelné ovládání do 20 mA

- viz FLASHLIGHT\_CTRL v config\_<MCU>.h pro tvůj port pin
- //#define HW\_FLASHLIGHT - odkomentuj k aktivaci

### **BH1750VFI** Snímač okolního světla

- vyžaduje sběrnici I2C a podporu čtení I2C
- odkomentuj pro povolení a také pro výběr správné adresy I2C
- //#define HW\_BH1750
- #define BH1750\_I2C\_ADDR 0x23 /\* I2C adresa 0x23 (ADDR low) \*/
- //#define BH1750\_I2C\_ADDR 0x5c /\* I2C adresa 0x5c (ADDR high) \*/

## **5.2.2. Možnosti softwaru**

### **snadno použitelný PWM generátor**

- výstup přes OC1B
- #define SW\_PWM\_SIMPLE \* - komentuj k odvolbě

### **PWM generátor s rozšířeným provozem**

- výstup přes OC1B
- vyžaduje další tlačítka a LCD s více než 2 řádky textu
- //# define SW\_PWM\_PLUS - odkomentuj k aktivaci

### **PWM Generátor: zobrazuje také délku trvání pulzů**

- trvání na základě rozlišení časovače
- //#define PWM\_SHOW\_DURATION - odkomentuj k aktivaci

### **měření indukčnosti**

- #define SW\_INDUCTOR \* - komentuj k odvolbě

### **měření ESR a měření ESR v obvodu**

- MCU takt> = 8 MHz nutný
- #define SW\_ESR \* - komentuj k odvolbě
- výběr SW\_OLD\_ESR pro starou metodu měření od 180nF
- //# define SW\_OLD\_ESR - odkomentuj k aktivaci

### **ESR-tool** měření ESR v obvodu

- vyžaduje aktivaci od SW\_ESR nebo SW\_OLD\_ESR
- //#define SW\_ESR\_TOOL - odkomentuj k aktivaci

### **test rotačních snímačů**

- //# define SW\_ENCODER - odkomentuj k aktivaci

### **generátor obdélníkových vln**

- vyžaduje další tlačítka nebo otočný kodér
- #define SW\_SQUAREWAVE \* - komentuj k odvolbě

### **IR detektor/dekodér** (pomocí testovacích pinů)

- vyžaduje modul IR přijímače, např. série TSOP
- modul bude připojen k testovacím pinům
- #define SW\_IR\_RECEIVER \* - komentuj k odvolbě

### **rozložení vývodů sondy** pro modul IR přijímače

- užitečné zejména pro testery s konektorem ZIF
- vyber si jeden
- #define SW\_IR\_RX\_PINOUT\_G\_V\_D /\* 1-Gnd 2-Vcc 3-Data \*/ - standard
- //#define SW\_IR\_RX\_PINOUT\_D\_G\_V /\* 1-Data 2-Gnd 3-Vcc \*/
- //#define SW\_IR\_RX\_PINOUT\_D\_V\_G /\* 1-Data 2-Vcc 3-Gnd \*/

### **odpor omezující proud pro IR přijímací modul**

- pouze pro 5V moduly
- varování: jakýkoli zkrat může zničit MCU
- //#define SW\_IR\_DISABLE\_RESISTOR - odkomentuj k aktivaci



### **potvrzovací tón pro platný datový rámec/paket**

- vyžaduje bzučák (HW\_BUZZER)  
//#define SW\_IR\_RX\_BEEP

- pro aktivaci

### **další protokoly pro IR detektor/dekodér**

- vzácnější protokoly, které zvyšují využití flash paměti;  
//# define SW\_IR\_RX\_EXTRA

- odkomentuj k aktivaci

### **IR vysílač pro dálkové ovládání**

- vyžaduje další tlačítka a zobrazení s více než 4 řádky textu  
- vyžaduje také IR LED s jednoduchým ovladačem  
//# define SW\_IR\_TRANSMITTER

- odkomentuj k aktivaci

### **alternativní zpoždění pro IR dálkový vysílač**

- v případě, že C kompilátor zkazí výchozí smyčku zpoždění a způsobí nesprávné doby pulsu/pauzy  
//#define SW\_IR\_TX\_ALTDelay

- odkomentuj k aktivaci

### **dodatkové protokoly pro IR dálkový vysílač**

\* - vzácnější protokoly, které zvyšují využití paměti flash;  
//# define SW\_IR\_TX\_EXTRA

- odkomentuj k aktivaci

### **test optických spojek**

//# define SW\_OPTO\_COUPLER

- odkomentuj k aktivaci

### **test UJT tranzistorů**

#define SW\_UJT

\* - komentuj k odvolbě

### **test (Schottky-clamped BJT)**

#define SW\_SCHOTTKY\_BJT

\* - komentuj k odvolbě

### **servo test**

- vyžaduje další tlačítka a zobrazení s více než 2 řádky textu  
//# define SW\_SERVO

- odkomentuj k aktivaci

### **měření teploty s DS18B20**

- aktivuj také ONEWIRE\_PROBES nebo ONEWIRE\_IO\_PIN (viz část „bussy“)  
- podívej se prosím na UI\_ROUND\_DS18B20  
//#define SW\_DS18B20

- odkomentuj k aktivaci

### **měření teploty s DS18S20** jednodrátový snímač teploty

- DS18S20\_HIGHRES: zapnutí vysokého rozlišení (0,01 °C)  
normální rozlišení je 0,5°C  
- povol také ONEWIRE\_PROBES nebo ONEWIRE\_IO\_PIN (viz část "Sběrnice")  
//#define SW\_DS18S20 - odkomentuj k aktivaci  
//#define DS18S20\_HIGHRES /\* high resolution (0.01°C) \*/ - odkomentuj k aktivaci

### **OneWire RAM-kód** číst a zobrazit.

- vyžaduje více než 2 řádky textu  
- aktivuj také ONEWIRE\_PROBES nebo ONEWIRE\_IO\_PIN (viz část „bussy“)  
//#define SW\_ONEWIRE\_READ\_ROM - odkomentuj k aktivaci

### **OneWire skenování** zobrazuje ROM-kódy, všech připojených uživatelů.

- vyžaduje více než 2 řádky textu  
- aktivuj také ONEWIRE\_PROBES nebo ONEWIRE\_IO\_PIN (viz část „bussy“)  
//#define SW\_ONEWIRE\_SCAN - odkomentuj k aktivaci

### **test unikajících proudů kondenzátoru**

- vyžaduje LCD s více než dvěma řádky  
//# define SW\_CAP\_LEAKAGE - odkomentuj k aktivaci

### **reverzní zobrazení hFE pro BJT**

- výměna hFE za kolektor a emitor  
#define SW\_REVERSE\_HFE \* - komentuj k odvolbě

**zobrazení I<sub>C</sub>/I<sub>E</sub>** zkoušecího proudu při hFE-měření

- I<sub>C</sub> pro společný emitor
- I<sub>E</sub> pro společný kolektor

```
//#define SW_HFE_CURRENT
```

- odkomentuj k aktivaci

**R/C/L monitorování** na pinech #1 a #3

- monitorování indukci vyžaduje aktivaci SW\_INDUCTORS
- pro ESR musí být SW\_ESR nebo SW\_OLD\_ESR aktivován

```
//#define SW_MONITOR_R    jen R
```

```
//#define SW_MONITOR_C    jen C plus ESR
```

```
//#define SW_MONITOR_L    jen L
```

```
//#define SW_MONITOR_RCL  R plus L, nebo C plus ESR
```

```
//#define SW_MONITOR_RL   R plus L
```

- aktivuj(jeden||více)

**C/L monitor:** Auto-Hold (automatické podržení)

- vyžaduje displej s více než dvěma řádky textu
- odkomentuj pro povolení (jednoho nebo více)

```
//#define SW_MONITOR_HOLD_ESR auto-hold ESR (C monitor)
```

```
//#define SW_MONITOR_HOLD_L   auto-hold L   (L monitor)
```

**DHT11, DHT22 a kompatibilní senzory vlhkosti a teploty**

```
//# define SW_DHTXX
```

- odkomentuj k aktivaci

**kontrola Odporu, zda vyhovuje standardní hodnotě řady E**

- vyžaduje display s více než 2 řádky
- režim barevného kódu vyžaduje barevné grafické zobrazení

```
//#define SW_R_E24_5_T    E24 5% tolerance, text
```

```
//#define SW_R_E24_5_CC   E24 5% tolerance, color-code
```

```
//#define SW_R_E24_1_T    E24 1% tolerance, text
```

```
//#define SW_R_E24_1_CC   E24 1% tolerance, color-code
```

```
//#define SW_R_E96_T      E96 1% tolerance, text
```

```
//#define SW_R_E96_CC     E96 1% tolerance, color-code
```

```
//#define SW_R_E96_EIA96  E96 1% tolerance, EIA-96-code- aktivuj(jeden||více)
```

**kontrola kondenzátoru, zda vyhovuje standardní hodnotě řady E**

- vyžaduje display s více než 2 řádky

```
//#define SW_C_E6_T       E6 20% tolerance, text
```

```
//#define SW_C_E12_T      E12 10% tolerance, text
```

- aktivuj(jeden||více)

**kontrola cívky, zda vyhovuje standardní hodnotě řady E**

- vyžaduje display s více než 2 řádky

```
//#define SW_L_E6_T       E6 20% tolerance, text
```

```
//#define SW_L_E12_T      E12 10% tolerance, text
```

- aktivuj(jeden||více)

**test kontinuity**

- vyžaduje bzučák (HW\_BUZZER)

```
//#define SW_CONTINUITY_CHECK
```

- odkomentuj k aktivaci

**zobrazit další informace** o možném potenciometru/trimpotu

- zobrazuje součet obou odporů a poměrů v %

```
//#define SW_R_TRIMMER
```

- odkomentuj k aktivaci

**zobrazit úbytek samovybíjecího napětí (v %) u kondenzátoru > 50nF**

```
//#define SW_C_VLOSS
```

- odkomentuj k aktivaci

**diode/LED test fotodiod**

```
//#define SW_PHOTODIODE
```

- odkomentuj k aktivaci

**rychlý test diod/LED**

- vyžaduje displej s více než 2 řádky textu

```
//#define SW_DIODE_LED
```

- odkomentuj k aktivaci

### 5.2.3. Řešení obcházení pro některé testery

**deaktivace u některých testerů** měření hFE se společným kolektorovým obvodem a Rl jako  
bázový odpor  
- Problém:  
Hodnoty hFE jsou příliš vysoké, protože základní napětí je měřeno příliš nízko.  
- ovlivnění testerů:  
Hiland M664 (probíhá řešení)  
//#define NO\_HFE\_C\_RL - odkomentuj k aktivaci

### Alternativní řízení napájení pro klony s řídicí jednotkou MCU SCT15L104W.

- Problém:  
Tester se po prvním testovacím cyklu náhle vypne.  
- Dotčené testery:  
T7-H, pravděpodobně i další modely rodiny TC-1  
//#define PASSIVE\_POWER\_CTRL - aktivovat

### 5.2.4. Řešení pro některá IDE

**spouštěcí cykly oscilátoru** (po probuzení z úsporného režimu):

- typické hodnoty
- interní RC oscilátor:...6
- křemenný krystal:..... 16384 (také 256 nebo 1024 v závislosti na nastavení pojistky)
- rezonátor:..... 16384 (také 256 nebo 1024, v závislosti na nastavení pojistky)  
- změň hodnotu, pokud se nehodí k tvému testeru!

```
798 #ifndef OSC_STARTUP
799 #define OSC_STARTUP 16384
800 #endif
```

Výpis 5.1. změň hodnotu, pokud se neshoduje s testerem!

### 5.2.5. uživatelské rozhraní

**dostupné jazyky** standard je ISO 8859 -1.

Volba \_2 odpovídá ISO 8859 -2 (s háčky a čárkami.). - Ruština je vždy Windows -1251.

```
826 //define UI_ENGLISH
827 //define UI_BRAZILIAN
828 //define UI_CZECH
829 #define UI_CZECH_2
830 //define UI_DANISH
831 //define UI_FRENCH
832 //define UI_GERMAN
833 //define UI_ITALIAN
834 //define UI_POLISH
835 //define UI_POLISH_2
836 //define UI_ROMANIAN
837 //define UI_RUSSIAN
838 //define UI_RUSSIAN_2
839 //define UI_SPANISH
```

Výpis 5.2. Zde byla zvolena angličtina.

**čárka místo tečky** k označení desetinných zlomků.

//# define UI\_COMMA - odkomentuj k aktivaci

**teplota ve Fahrenheitu** místo Celsia.

//# define UI\_FAHRENHEIT - odkomentuj k aktivaci

**hexadecimální hodnoty** ukázat velkými písmeny

//# define UI\_HEX\_UPPERCASE - odkomentuj k aktivaci

**standart automatický režim** - místo nepřetržitého režimu

//# define UI\_AUTOHOLD - odkomentuj k aktivaci

**dočasné přepnutí do režimu automatického podržení** při detekci součásti

- pouze v nepřetržitém režimu  
//#define UI\_AUTOHOLD\_FOUND - odkomentuj k aktivaci

**volba menu zkratem** všech tří testovacích pinů.

- staré výchozí chování

```
//# define UI_SHORT_CIRCUIT_MENU
```

- odkomentuj k aktivaci

**pokyny místo kurzoru** , pokud jsou k dispozici.

- aktuálně pouze „Nabídka/Test“
- vyžaduje další tlačítka a displej s dostatečným počtem řádků textu (doporučeno: > = 8 řádků)

```
//#define UI_KEY_HINTS
```

- odkomentuj k aktivaci

**testovací profil** zobrazit při zapnutí testeru.

- \* - pro testery přídavnými testovacími kabely

```
//# define UI_SHOOSE_PROFILE
```

- odkomentuj k aktivaci

**přidání třetího profilu** pro hodnoty nastavení.

- pro testery přídavnými testovacími kabely

```
//# define UI_THREE_PROFILES
```

- odkomentuj k aktivaci

**výstup přes sériové rozhraní TTL**

- aktivuj také SERIAL\_BITBANG nebo SERIAL\_HARDWARE (viz část „bussy“)

```
//# define UI_SERIAL_COPY
```

- odkomentuj k aktivaci

**ovládání testeru přes sériové rozhraní TTL**

- aktivuj také SERIAL\_BITBANG nebo SERIAL\_HARDWARE a SERIAL\_RW

```
//#define UI_SERIAL_COMMANDS
```

- odkomentuj k aktivaci

**maximální čekací doba** po testování (v ms)

- platí pouze pro nepřetržitý režim
- čas mezi výstupem výsledku a začátkem nového zkušebního běhu.

```
#define CYCLE_DELAY ... 3000
```

- v případě potřeby změn čas

**maximální počet testovacích běhů** bez nalezených komponentů

- platí pouze pro nepřetržitý režim
- po dosažení tohoto počtu se tester vypne.

```
#define CYCLE_MAX... 5
```

- v případě potřeby změn číslo

**automatické vypnutí** , pokud není po nějakou dobu stisknuto žádné tlačítko.

- \* - vztahuje se pouze na režim automatického pozastavení

```
//#define POWER_OFF_TIMEOUT... 60
```

- odkomentuj k aktivaci

**efektní rozvržení vývodů** pro 3 pólové komponenty

- \* - vyžaduje grafické zobrazení a bitové mapy symbolů (config\_<MCU>.h)

```
#define SW_SYMBOLS
```

\* - komentuj k odvolbě

**efektní rozvržení vývodů** zobrazení čísel sond vpravo nad/pod symbolem

- vyžaduje zapnutí symbolů součástek (SW\_SYMBOLS) v kapitole 5.2.5 na stránce 36

```
//#define UI_PINOUT_ALT
```

- odkomentuj k aktivaci

**zobrazení symbolu otazníku** při neúspěšném provedení testu

- vyžaduje zapnutí symbolů součástek (SW\_SYMBOLS) v kapitole 5.2.5 na stránce 36

```
//#define UI_QUESTION_MARK
```

- odkomentuj k aktivaci

**zobrazení symbolu Zenerovy diody** při testu Zenerové diody

- vyžaduje zapnutí symbolů součástek (SW\_SYMBOLS) v kapitole 5.2.5 na stránce 36

```
//#define UI_ZENER_DIODE
```

- odkomentuj k aktivaci

**rozšířený čítač frekvence:** zobrazení symbolu křemenného krystalu pro režimy LF/HF

- vyžaduje zapnutí symbolů součástek (SW\_SYMBOLS) v kapitole 5.2.5 na stránce 36

```
//#define UI_QUARTZ_CRYSTAL
```

- odkomentuj k aktivaci

**DS18B20/DS18S20/DHTXX:** zobrazení symbolu senzoru

- vyžaduje zapnutí symbolů součástek (SW\_SYMBOLS) v kapitole 5.2.5 na stránce 36

```
//#define UI_ONEWIRE
```

- odkomentuj k aktivaci (zatím není podporováno)

**zakázat textový rozpis vývodů** pro 3pinové polovodiče

- vyžaduje zapnutí symbolů součástek (SW\_SYMBOLS) v kapitole 5.2.5 na stránce 36

```
//#define UI_NO_TEXTPINOUT
```

- odkomentuj k aktivaci

**vypnutí textového rozpisu vývodů** tělesa/vnitřní diody pro tranzistory MOSFET

- vyžaduje zapnutí symbolů součástek (SW\_SYMBOLS) v kapitole 5.2.5 na stránce 36
- //#define UI\_NO\_BODYDIODE\_TEXTPINOUT - odkomentuj k aktivaci

**stav baterie:** ikona na displeji

- vyžaduje písmo s dalšími znaky (zkontroluj písmo!)
- není k dispozici pro displeje založené na HD44780 a ST7036
- nelze použít s LCD\_VT100
- //#define UI\_BATTERY - odkomentuj k aktivaci

**stav baterie:** zobrazí se na posledním řádku po zobrazení výsledku sondování

- //#define UI\_BATTERY\_LASTLINE - odkomentuj k aktivaci

**zobrazení ID sond** pomocí obrácených barev

- vyžaduje písmo s dalšími znaky (zkontroluj písmo!)
- není k dispozici pro displeje založené na HD44780 a ST7036
- //#define UI\_PROBE\_REVERSED - odkomentuj k aktivaci

**barevné kódování zkušebních pinů**

- vyžaduje barevný grafický displej
- uprav colors.h pro výběr správných barev sond (COLOR\_PROBE\_1, COLOR\_PROBE\_2 a COLOR\_PROBE\_3)
- #define UI\_PROBE\_COLORS \* - komentuj k odvolbě

**barevné názvy**

- vyžaduje barevné grafické zobrazení
- uprav colors.h pro výběr preferované barvy (COLOR\_TITLE)
- //#define UI\_COLORED\_TITLES - odkomentuj k aktivaci

**barevný kurzor a nápovědy ke klávesám**

- vyžaduje barevný grafický displej
- úprava souboru colors.h pro výběr preferované barvy (COLOR\_CURSOR)
- //#define UI\_COLORED\_CURSOR - odkomentuj k aktivaci

**barevné hodnoty**

- pouze hodnota, nikoli jednotka
- vyžaduje barevné grafické zobrazení
- úprava souboru colors.h pro výběr preferované barvy (COLOR\_VALUE)
- //#define UI\_COLORED\_VALUES - odkomentuj k aktivaci

**menu:** posouvání menu po stránkách místo po položkách

- urychluje práci s nabídkou u grafických displejů, zejména barevné displeje s vysokým rozlišením
- //#define UI\_MENU\_PAGEMODE - odkomentuj k aktivaci

**automatické ukončení hlavní nabídky** po provedení funkce

- #define UI\_MAINMENU\_AUTOEXIT \* - komentuj pro deaktivaci

**volbu tester vypnout** ukázat v menu

- //# define SW\_POWER\_OFF - odkomentuj k aktivaci

**hlavní nabídka: zobrazení písma** pro testovací účely

- výchozí výstupní formát: indexové číslo (hex) a 8 znaků (včetně nedostupných).
- zabalený výstupní formát: žádný index, pouze dostupné znaky, kompletní textový řádek
- //#define SW\_FONT\_TEST - odkomentuj k aktivaci
- //#define FONT\_PACKED /\* zabalený výstupní formát \*/ - odkomentuj k aktivaci

**zobrazení symbolů komponentů** pro účely testování

- vyžaduje povolení symbolů (SW\_SYMBOLS) v kapitole 5.2.5 na stránce 36
- //#define SW\_SYMBOL\_TEST hfill - odkomentuj k aktivaci

### zaokrouhlení hodnot pro DS18B20

- DS18B20 (0.1 °C/F)

```
//#define UI_ROUND_DS18B20
```

- odkomentuj k aktivaci

### zarovnání informací a některých dalších textů na střed

- vyžaduje zobrazení s více než 3 řádky textu

```
//#define UI_CENTER_ALIGN hfill - odkomentuj k aktivaci
```

### potvrzovací pípnutí po dokončení sondování

- vyžaduje bzučák (HW\_BUZZER)

```
//#define UI_PROBING_DONE_BEEPhfill - odkomentuj k aktivaci
```

### ukládání dat firmwaru (texty, tabulky atd.)

- samoladící data se vždy ukládají do EEPROM

- písmena a symboly jsou vždy ukládány do Paměti flash

```
#define DATA_EEPROM store data in EEPROM
```

```
//#define DATA_FLASH store data in Flash
```

- zvol jedno

## 5.2.6. Správa napájení

### typ vypínače

- softwarové řešení, (standard), které dokáže vypnout Tester

- ruční vypínač. Zde se tester samostatně vypnout nemůže

```
//#define POWER_SWITCH_SOFT
```

```
#define POWER_SWITCH_MANUAL
```

- komentuj k odvolbě

### monitorovací režim baterie

- BAT\_NONE zcela deaktivuje monitorování baterie

- BAT\_DIRECT přímé měření napětí baterie (<5V)

- BAT\_DIVIDER měření pomocí děliče napětí

```
//#define BAT_NONE
```

```
//#define BAT_DIRECT
```

```
#define BAT_DIVIDER
```

- zvoleno

### volitelné externí napájení bez monitorování

- někteří testéři podporují další externí napájení, ale kvůli

zapojení, není dovoleno měřit napětí. To by způsobilo vypnutí, při

zjištění nízkého napětí. Volba níže zabraňuje vypnutí, když měřené

napětí klesne pod 0,9 V (způsobeno svodovým proudem diody).

```
//#define BAT_EXT_UNMONITORED
```

- odkomentuj k aktivaci

### děliče napětí pro monitorování baterie

- BAT\_R1: horní odpor v  $\Omega$

- BAT\_R2: dolní odpor v  $\Omega$

```
#define BAT_R1...10000
```

\*\* - v případě potřeby uprav hodnotu

```
#define BAT_R2...3300
```

\*\* - v případě potřeby uprav hodnotu

### úbytek napětí způsobený ochranou diodou proti zpětnému napětí

- nebo tranzistorem pro správu napájení (v mV)

- případně jinou částí obvodu v napájení

- vezmi si DMM a změř úbytek napětí!

- Schottkyho dioda asi 200 mV/PNP BJT asi 100 mV.

```
#define BAT_OFFSET...290
```

\*\* - v případě potřeby uprav hodnotu

### nízké napětí baterie (v mV)

- tester varuje, když je dosaženo BAT\_WEAK.

- úbytek napětí BAT\_OFFSET je zahrnut do výpočtu.

```
#define BAT_WEAK...7400
```

\*\* - v případě potřeby uprav hodnotu

### vypínací napětí baterie (v mV)

- tester se vypne, když je dosaženo BAT\_LOW.

- úbytek napětí BAT\_OFFSET je při výpočtu zohledněn

```
#define BAT_LOW...6400
```

\*\* - v případě potřeby uprav hodnotu

### režim spánku pro nižší spotřebu energie

```
#define SAVE_POWER
```

\* - komentuj k odvolbě

### 5.2.7. Nastavení a kompenzace měření

**ADC napětí reference** na základě Vcc (v mV)

```
#define UREF_VCC ... 5001
```

\*\* - v případě potřeby uprav hodnotu

**offset interního referenčního napětí** (v mV): -100 až 100

- kompenzuje rozdíly mezi skutečnou a změřenou hodnotou.

- ADC má rozlišení přibližně 4,88 mV pro V<sub>ref</sub> = 5V (Vcc) a 1,07 mV pro

V<sub>ref</sub> = 1,1V (bandgap).

- přidáno k měřenému napětí referenční hodnoty bandgapu.

```
#define UREF_OFFSET ... 0
```

\*\* - v případě potřeby uprav hodnotu

**přesné hodnoty** testovacích odporů

- výchozí hodnota pro R<sub>l</sub> je 680Ω

- výchozí hodnota pro R<sub>h</sub> je 470 kΩ

```
/ * Rl vΩ * /
```

```
#define R_LOW ... 680
```

\*\* - v případě potřeby uprav hodnotu

```
/ * Rh inΩ * /
```

```
#define R_HIGH ... 470000
```

\*\* - v případě potřeby uprav hodnotu

**offset pro systematické chyby** měření odporu pomocí R<sub>h</sub> (470k) vΩ

- pokud jsou odpory > 20k příliš vysoké nebo příliš nízké, uprav odpovídajícím způsobem offset.

- výchozí offset je 350Ω

```
#define RH_OFFSET ... 3500
```

\*\* - v případě potřeby uprav hodnotu

**odpor zkušebních pinů/kabelů** (v 0,01Ω) - standardní posun pro stopy a zkušební kabely

- odpor dvou testovacích kolíků připojených ke sérii

- za předpokladu, že všechny testovací piny mají stejný/podobný odpor je aktualizováno samoladěním

```
#define R_ZERO ... 20
```

\*\* - v případě potřeby uprav hodnotu

**odpor pro jednotlivé zkušební páry-kabelů** , v případě, že jsou velmi rozdílné

- je aktualizováno v samoladění

```
#define R_MULTIOFFSET
```

\*\* - v případě potřeby uprav hodnotu

**kapacita testovacích pinů/kabelů** (v pF) - standardní offset pro MCU, desku a testovací kabel

- je aktualizováno v samoladění

- Příklady kapacit pro různé délky kabelů:

3pF      asi 10cm

9pF      asi 30cm

15pF     asi 50cm

- maximální hodnota      ... 100

```
#define C_ZERO ... 43
```

- v případě potřeby uprav hodnotu

**specifická kapacita testovacích kabelů** místo průměrné hodnoty pro všechny testovací piny

```
// #define CAP_MULTIOFFSET
```

- odkomentuj k aktivaci

**maximální vybíjecí napětí** pro kondenzátory (v mV)

- pod kterým napětím vidíme kondenzátor jako vybitý

```
#define CAP_DISCHARGED ... 2
```

\*\* - v případě potřeby uprav hodnotu

**korekční faktory kondenzátorů** (v 0,1 %)

- pozitivní faktor zvyšuje hodnotu kapacity

- záporný faktor snižuje hodnotu kapacity

CAP\_FACTOR\_SMALL      pro kondenzátory      < 4,7 μF

CAP\_FACTOR\_MID        pro kondenzátory      4,7 - 47 μF

CAP\_FACTOR\_LARGE      pro kondenzátory      > 47 μF

```
#define CAP_FACTOR_SMALL ... 0
```

bez korekce      \*\* - uprav hodnotu

```
#define CAP_FACTOR_MID ... -40
```

-4,0 %      \*\* - uprav hodnotu

```
#define CAP_FACTOR_LARGE ... -90
```

- 9,0 %      \*\* - uprav hodnotu

**počet kol ADC pro každé měření**

- platné hodnoty se pohybují od 1 do 255.

```
#define ADC_SAMPLES ... 25
```

\*\* - v případě potřeby uprav hodnotu

### 100nF vyrovnávací kondenzátor AREF

- používají některé karty MCU
  - prodlouží dobu měření
  - doporučení: vyměň za 1nF kondenzátor
- ```
#define ADC_LARGE_BUFFER_CAP
```

- komentuj k odvolbě

### 5.2.8. R&D - určeno pro vývojáře firmwaru

#### aktivace funkce čtení dat modulu displeje

- ovladače displeje a nastavení rozhraní to musí podporovat
- ```
//#define LCD_READ
```

- odkomentuj k aktivaci

#### aktivace funkce čtení ID řadiče displeje

- ID je zobrazeno na uvítací obrazovce (podle verze firmwaru)
  - vyžaduje funkce čtení displeje (LCD\_READ)
  - doporučeno: sériový výstup (UI\_SERIAL)
- ```
//#define SW_DISPLAY_ID
```

- odkomentuj k aktivaci

**čte registry řadiče displeje** a vysílá je sériově přes TTL.

- vyžaduje funkce čtení displeje (LCD\_READ) a sériový výstup (UI\_SERIAL)
- ```
//#define SW_DISPLAY_REG
```

- odkomentuj k aktivaci

### 5.2.9. Busse

**I2C bus** může být vyžadován některým hardwarem

- již bylo možné aktivovat pomocí nastavení zobrazení (config<MCU>.h)
  - pro bit-bang-port a -ins viz I2C\_PORT (config<MCU>.h)
  - Hardware I2C (TWI) automaticky používá správné MCU piny
  - jeden z Taktů odkomentovat
- ```
//# define I2C _BITBANG          bit-bang I2C
//# define I2C _HARDWARE         Hardware MCU TWI
//# define I2C _STANDARD _MODE 100kHz sběrnice hodiny
//# define I2C _FAST _MODE      400kHz sběrnice hodiny
//# define I2C _RW              povolit podporu čtení
```

\* - chceš-li aktivovat, tak buď I2C\_BITBANG nebo I2C\_HARDWARE

**SPI bus** může být vyžadován některým hardwarem

- již bylo možné aktivovat pomocí nastavení zobrazení (config<MCU>.h)
  - pro bit-bang-port a -ins viz SPI\_PORT (config<MCU>.h)
  - hardware SPI automaticky používá správné piny MCU
- ```
//#define SPI _BITBANG          bit-bang SPI
//#define SPI _HARDWARE         Hardware SPI
//#define SPI _RW               Povolit podporu čtení
//#define SPI _SLOWDOWN         zpomalení SPI bit-bang
```

\* - chceš-li ho použít, tak buď SPI\_BITBANG nebo SPI\_HARDWARE

**sériové TTL rozhraní** může být již povoleno v LCD nastavení (config<MCU>.h)

- pro bit-bang-port a -ins viz SERIAL\_PORT (config<MCU>.h).
  - hardware Serial automaticky používá správné piny MCU
- ```
//# definuje SERIAL _BITBANG    bit-bang-serial
//# definuje SERIAL _HARDWARE   Hardware Serial
//# define SERIAL _RW           Povolit podporu čtení
```

\* - chceš-li povolit, odkomentuj buď SERIAL\_BITBANG nebo SERIAL\_HARDWARE

#### OneWire bus

- informace o vyhrazeném I/O MCU pinu najdeš pod ONEWIRE\_PORT v (config<MCU>.h).
- ```
// # define ONEWIRE _PROBES      přes testovací piny
// # define ONEWIRE _IO _PIN     pomocí vyhrazeného I/O pinu
```
- \* - povol buď ONEWIRE\_PROBES, nebo ONEWIRE\_IO\_PIN



Jak již bylo zmíněno, firmware lze přizpůsobit pro různé testery a další funkce.

### 6.1. V k-verzi

ovládáš nastavením v Makefile. Proto, že projektant pro tento tester tento soubor předkonfiguroval, musíš v praxi nastavit pouze tvůj programátor a jazyk obsluhy. Ten řádek 190 jsem změnil jen proto, že je ten výstup tak jako-tak aktivovaný. Jen ten text byl utlumený. Změny pro tento tester jsou minimální:

Software	originál KHK verze 1.13k
jméno podadresáře	mega328_color_kit
použito FLASH	98 %
použito EEPROM	87.8 %
řádek:	změna v Makefile:
75	UI_LANGUAGE = LANG_CZECH
190	CFLAGS += -DFREQUENCY_50HZ
375	PROGRAMMER=usbasp
376	BitClock=20
377	PORT=usb

Tabulka 6.1. Použitá SW a modifikace v Makefile u k-software

Kromě toho byly s úspěchem vyzkoušeny : POLOLU a USBtiny ISP

### 6.2. V m-verzi vypadá ta situace úplně jinak.

Vývojář se zcela spolehně na technické znalosti svých příznivců a jejich smysl k experimentování. Jako pomoc slouží soubor Clones.txt ve které jsou různé kloní stručně popsány, například tento tester je tam pod jménem AY AT Clone. To byl jeden z důvodů, napsat tuto příručku. Nastavení jsou zde rozmístněny v souboru [Makefile](#), [config.h](#) a [config\\_328.h](#).

- **Makefile** řídí překlad zdrojového kódu a obsahuje základní věci, jako jsou typy MCU a ISP programátory.

- **V souboru config.h** existují obecná nastavení pro provoz a funkce

**a soubor config\_328.h** je zodpovědný za věci na hardwarové úrovni, tedy za moduly LCD a přiřazení pinů.

### 6.3. Makefile

V Makefile se provádí nastavení nastavením určitých proměnných. K přizpůsobení změn prostě hodnotu nebo řetězec za proměnnou. Pro některé proměnné existuje několik návrhů, které jsou komentovány pomocí symbolu #. Tam, v případě potřeby komentář (# smazat) a nebo komentář k výchozímu nastavení (# vložit).

#### 6.3.1. MCU-Typ

25 MCU = atmega328

Výpis 6.1. Předvolba je již atmega328

#### 6.3.2. MCU-Taktfrequenz

32 FREQ = 8

Výpis 6.2. Předvolba je již 8MHz

#### 6.3.3. Oszillator-Typ

38 OSCILLATOR = Crystal

Výpis 6.3. Předvolba je již Crystal

### 6.3.4. Avrdude MCU-Typ

```
58 # avrdude: part number of MCU
59 # - ATmega 328 : m328
60 # - ATmega 328P : m328p
61 # - ATmega 328PB : m328pb
62 # - ATmega 324P : m324p
63 # - ATmega 324PA : m324pa
64 # - ATmega 640 : m640
65 # - ATmega 644 : m644
66 # - ATmega 644P : m644p
67 # - ATmega 644PA : m644p
68 # - ATmega 1280 : m1280
69 # - ATmega 1284 : m1284
70 # - ATmega 1284P : m1284p
71 # - ATmega 2560 : m2560
72 PARTNO = m328p
```

Výpis 6.4. Předvolba je již m328p

### 6.3.5. Avrdude ISP-Programmierer

Avrdude potřebuje:

- jméno programátora
- bitový takt
- port.

```
76 # Arduino as ISP
77 #PROGRAMMER = stk500v1
78 #PORT = /dev/ttyACM0
79 #OPTIONS = -b 19200
80
81 # Bus Pirate
82 #PROGRAMMER = buspirate
83 #PORT = /dev/bus_pirate
84 #OPTIONS = -B 10.0
85
86 # Diamex ALL-AVR/AVR-Prog
87 PROGRAMMER = avrispmkII
88 PORT = usb
89 OPTIONS = -B 1.0
90
91 # Pololu USB AVR Programmer
92 #PROGRAMMER = stk500v2
93 #PORT = /dev/ttyACM0
94 #OPTIONS = -B 1.0
95
96 # USBasp
97 #PROGRAMMER = usbasp
98 #PORT = usb
99 #OPTIONS = -B 20
100
101 # USBtinyISP
102 #PROGRAMMER = usbtiny
103 #PORT = usb
104 #OPTIONS = -B 5.0
105
106 # Arduino Uno bootloader via serial/USB
107 #PROGRAMMER = arduino
```

Výpis 6.5. Předvolba je Diamex

Zde byla souprava programátorů již editovaná a známé a vyzkoušené nastavení přidáno. Pokud není tvůj programátor uvedený, přidej ho do Makefile ručně.

Další informace najdeš v příručce Avrdude nebo v online dokumentaci [5].

## 6.4. config.h

Tento soubor slouží k nastavení provozu a funkcí. Protože se zde jedná o normální soubor se záhlavím C, používají se zde, na rozdíl od „Makefile“ známá pravidla komentování v C. Chceš-li něco aktivovat, odstraň znaky „//“ na začátku řádku a na deaktivování je zase na začátek řádku vlož. Některá nastavení vyžadují číselnou hodnotu, kterou můžeš případně upravit.

### 6.4.1. Pro tento tester musí být změněno

```
39 #define HW_ENCODER
```

Výpis 6.6. nastavení kodéru

```
61 #define ENCODER_STEPS 20
```

Výpis 6.7. počet kroků kodéru

```
83 #define HW_REF25
```

Výpis 6.8. externí reference

```
117 #define HW_ZENER
```

Výpis 6.9. aktivace voltmetru

```
140 #define ZENER_UNSWITCHED
```

Výpis 6.10. periodické měření bez nutnosti mačkání tlačítka

```
167 #define HW_PROBE_ZENER
```

Výpis 6.11. po měření kontrolovat také svorky napětí

```
191 #define HW_FREQ_COUNTER_BASIC
```

Výpis 6.12. aktivace čítače frekvence

```
207 //#define FREQ_COUNTER_PRESCALER 16 /* 16:1 */
```

Výpis 6.13. deaktivace děliče, který není v sadě

### 6.4.2. Nyní začíná utrpení nutného výběru

z velkého množství možných variant souboru config.h, protože paměť nastavuje limity. Naštěstí si můžeš vytvořit několik sad, které můžeš podle potřeby měnit.

```
235 #define HW_EVENT_COUNTER
```

Výpis 6.14. aktivace čítače událostí

```
245 #define EVENT_COUNTER_TRIGGER_OUT
```

Výpis 6.15. aktivace výstupu

```
388 //#define SW_PWM_SIMPLE
```

Výpis 6.16. deaktivace jednoduchého PWM

```
398 #define SW_PWM_PLUS
```

Výpis 6.17. aktivace rozšířeného PWM

```
407 #define PWM_SHOW_DURATION
```

Výpis 6.18. aktivace zobrazení délky trvání pulzů

```
425 //#define SW_ESR
```

Výpis 6.19. deaktivace měření ESR v obvodu

453 **#define** SW\_SQUAREWAVE

Výpis 6.20. aktivace generátoru obdélníkových vln

463 *//#define SW\_IR\_RECEIVER*

Výpis 6.21. deaktivace IR přijímače (modul schází)

541 *//#define SW\_OPTO\_COUPLER*

Výpis 6.22. deaktivace opto kopler

549 *//#define SW\_UJT*

Výpis 6.23. deaktivace UJT

557 *//#define SW\_SCHOTTKY\_BJT*

Výpis 6.24. deaktivace Schotky UJT

567 *//#define SW\_SERVO*

Výpis 6.25. aktivace testování modelových servos

577 **#define** SW\_DS18B20

Výpis 6.26. aktivace měření teploty s DS18B20

1168 **#define** UI\_ROUND\_DS18B20

Výpis 6.27. přesné měření na 0,1 °C

589 **#define** DS18S20\_HIGHRES */\* high resolution (0.01 C) \*/*

Výpis 6.28. aktivace měření teploty s DS18S20 s přesností na 0,01°C

670 **#define** SW\_DHTXX

Výpis 6.29. aktivace měření vlhkosti a teploty

724 **#define** SW\_R\_TRIMMER

Výpis 6.30. aktivace součtů obou odporů a poměrů u potenciometru

740 **#define** SW\_PHOTODIODE

Výpis 6.31. aktivace měření fotodiod

829 **#define** UI\_CZECH\_2

Výpis 6.32. deaktivace angličtiny a aktivace češtiny

847 **#define** UI\_COMMA

Výpis 6.33. používání čárky místo tečky

872 **#define** UI\_AUTOHOLD

Výpis 6.34. standart automatický režim

890 **#define** UI\_SHORT\_CIRCUIT\_MENU

Výpis 6.35. volba menu zkratem všech 3 pinů

901 **#define** UI\_KEY\_HINTS

Výpis 6.36. aktivace pomoci pro menu

967 **#define** POWER\_OFF\_TIMEOUT 30

Výpis 6.37. automatické vypínání za 30 vteřin

995 **#define** UI\_QUESTION\_MARK

Výpis 6.38. při nepodařeném měření zobrazit otazník

1013 **#define** UI\_QUARTZ\_CRYSTAL

Výpis 6.39. symbol krystalu pro režimy LF/HF

1050 **#define** UI\_BATTERY

Výpis 6.40. ikona baterie na displeji

1058 **#define** UI\_BATTERY\_LASTLINE

Výpis 6.41. ukázat stav baterie po měření

1079 *//#define UI\_PROBE\_COLORS*

Výpis 6.42. deaktivace barevného označení pinů

1099 **#define** UI\_COLORED\_CURSOR

Výpis 6.43. aktivace barevného kurzoru

1110 **#define** UI\_COLORED\_VALUES

Výpis 6.44. aktivace barevných hodnot

1136 **#define** SW\_POWER\_OFF

Výpis 6.45. aktivace nabídky vypnutí v menu

1177 **#define** UI\_CENTER\_ALIGN

Výpis 6.46. aktivace vyrovnávání textů

1196 *//#define DATA\_EEPROM /\* store data in EEPROM \*/*

1197 **#define** DATA\_FLASH */\* store data in Flash \*/*

Výpis 6.47. Aktivace zápisu do paměti Flash

1563 **#define** ONEWIRE\_PROBES */\* via probes \*/*

Výpis 6.48. Aktivace ONEWIRES Probes

## 6.5. Config\_328.h

obsahuje nastavení na hardwarově blízké úrovni pro displeje, ovládání a tak dále. Protože přiřazení pinů závisí na MCU typu, existují pro ATmega328 a rodinu kolem ATmega644 vlastní soubory s příslušným standardním přiřazením. Při překladu firmware je podle v Makefile zvolený MCU, automaticky integrován vhodný soubor. Zde se také jedná o soubor se záhlavím C, tzn. platí zde stejná pravidla komentování jako v config.h. Kromě “//” pro jednotlivé řádky, se pro blokové komentáře používají “#if 0 ... #endif”. tzn. na začátek vložit “#if 0” a na konec “#endif”. K použití kódu jednoduše řádky s “#if 0” a “#endif” odstranit. Místo odstranění stačí vložit před “#if 0” a před “#endif” “//”.

### 6.5.1. Nutné změny

```
629 #if 0
630 #define LCD_ST7565R          /* display controller ST7565R */
```

Výpis 6.49. Aktivní LCD ST756R modul deaktivován

```
672 #endif
```

Výpis 6.50. Aktivní LCD ST756R modul deaktivován

```
681 // #if 0
682 #define LCD_ST7735          /* display controller ST7735 */
```

Výpis 6.51. LCD\_ST7735 aktivovaný

```
689 #define LCD_RES    PD0      /* port pin used for /RESX (optional) */
690 #define LCD_CS     PD5      /* port pin used for /CSX (optional) */
691 #define LCD_DC     PD1      /* port pin used for D/CX */
692 #define LCD_SCL    PD2      /* port pin used for SCL */
693 #define LCD_SDA    PD3      /* port pin used for SDA */
```

Výpis 6.52. upravené nastavení pinů

```
699 #define LCD_FLIP_X          /* enable horizontal flip */
```

Výpis 6.53. horizontální točení

```
703 // #define LCD_LATE_ON      /* turn on LCD after clearing it */
```

Výpis 6.54. Pokud by měl tester začít s prázdným displejem, odstraň komentář

```
705 // #define FONT_8x16_ALT_HF    /* 8x16 alternative font */
706 // #define FONT_10x16_HF       /* 10x16 font */
707 // #define FONT_6X8_IS08859_2_HF /* 6x8 Central European font */
708 // #define FONT_8X8_IS08859_2_HF /* 8x8 Central European font */
709 #define FONT_8X12T_IS08859_2_HF /* thin 8x12 Central European font */
710 // #define FONT_8X16_IS08859_2_HF /* 8x16 Central European font */
711 // #define FONT_10X16_IS08859_2_HF /* 10x16 Central European font */
```

Výpis 6.55. vyber si písmo ... můžeš také přidat z volby v adresáři bitmaps.

Možné jsou všechny fondy, které mají na konci HF. (Horizontální točení).

```
714 #define SYMBOLS_24X24_HF      /* 24x24 symbols */
715 // #define SYMBOLS_30X32_HF    /* 30x32 symbols */
716 // #define SYMBOLS_30X32_ALT1_HF /* 30x32 alternative symbols #1 */
717 // #define SYMBOLS_30X32_ALT2_HF /* 30x32 alternative symbols #2 */
718 // #define SYMBOLS_32X32_HF    /* 32x32 symbols */
719 // #define SYMBOLS_32X32_ALT1_HF /* 32x32 alternative symbols #1 */
```

Výpis 6.56. zde můžeš z adresáře bitmaps přidat symboly.

V řádku 714 je přidána řada symbolů ve velikosti 24x24 pixelu.

**Poznámka:** čím větší znaky, tím stoupá spotřeba paměti a tím máš méně místa pro volby.

```
726 // #endif
```

Výpis 6.57. nezapomeň na komentování

```

952 #define ENCODER_A PD1 /* rotary encoder A signal */
953 #define ENCODER_B PD3 /* rotary encoder B signal */

```

Výpis 6.58. nastavení rotačního kodéru

```

963 #define KEY_INC PD1 /* increase push button (low active) */
964 #define KEY_DEC PD3 /* decrease push button (low active) */

```

Výpis 6.59. nastavení rotačního kodéru

```

993 #define IR_PORT PORTC /* port data register */
994 #define IR_DDR DDRC /* port data direction register */
995 #define IR_PIN PINC /* port input pins register */
996 #define IR_DATA PC6 /* data signal */

```

Výpis 6.60. nastavení IR detektor/dekodér

**6.5.2. informace** Toto nastavení bylo tak zvolené, aby bylo možné srovnání s k-verzí. Jak je z tabulky vidět, více se do ATmega 328 nevejde. K aktivování jiných funkcí m-verse musíš některé funkce, které nepotřebuješ v config.h deaktivovat.

Software	m-verse 1.52m
<a href="#">jméno clone</a>	<a href="#">AY AT Clone</a>
Program	97,2 %
Data	12,1 %
EEPROM	2,1 %

Tabulka 6.2. Údaje pro aktivovanou m-software

**Poznámka:** Při nastavení, **které se nevejde** do ATmega 328, na příklad při pokusu aktivovat v config\_328 u LCD Modulu na stránce 46 v řádce 711 font 10x16\_ISO8859 zobrazí se po **příkazu** `make` následující obraz:

```

/usr/lib/gcc/avr/5.4.0/../../../../avr/bin/ld: ComponentTester section `.text' will not fit in region `text'
/usr/lib/gcc/avr/5.4.0/../../../../avr/bin/ld: region `text' overflowed by 2234 bytes
collect2: Fehler: ld gab 1 als Ende-Status zurück
make: *** [Makefile:198: ComponentTester] Fehler 1
bohu@fuji-w:/media/bohu/Programme/Elektronik/TTester/GM328A/GM328a_52m$

```

Obrázek 6.1. zde trestá kompilátor chamtivost ;-)

po změně na font 8x12\_ISO8859 to vypadá daleko lépe:

```

AVR Memory Usage
-----
Device: atmega328

Program:  32166 bytes (98.2% Full)
(.text + .data + .bootloader)

Data:      248 bytes (12.1% Full)
(.data + .bss + .noinit)

EEPROM:    22 bytes (2.1% Full)
(.eeprom)

bohu@fuji-w:/media/bohu/Programme/Elektronik/TTester/GM328A/GM328a_52m$

```

Obrázek 6.2. všechno v cajku, ještě: `make upload`

a ... operace se zdařila. Ctester je nyní osobní a „nezaplatitelný”.

**Nezapomeň na kalibraci, samočinné nastavení a poté na ... uschování výsledků v paměti!**

Podle kapitoly 5.1.31

Aby zůstalo všem ostatním kolegům zoufalství a „bezesné noci“, kterými trpěl autor této kapitoly poté, co získal klone tester a bez jakékoli zkušenosti s AVR se rozhodl, ho „naučit česky“, ušetřeno, vznikla tato kapitola. Zde získané zkušenosti by měly pomoci všem ostatním naivním, „ochotným ... lehkomyšlným a nezkušeným“..., **ÚSPĚŠNĚ** naprogramovat jejich tester.

Tato příležitost je zároveň využita, poděkovat autorovi a vývojáři tranzistorového testeru Karlovi-Heinzy Kübbelerovi viz [2] za jeho obětavost a trpělivost, protože bez jeho pomoci, by následující stránky nebyly napsány.

Aby překlad firmwaru a vypálení do MCU uspělo a současně ... „nemuselo být „kolo“ znovu objeveno“, je část následujících stránek převzatá z popisu testeru tranzistoru od Karl-Heinze Kübbelera viz [2].

Tak ještě jednou ... **MOC VELKÝ VDĚK.**

### 7.1. Konfigurace testeru

K tomu si přečti kapitolu 6 od stránky 41.

### 7.2. Programování testeru

Programování testeru je řízeno souborem Makefile. Makefile zajišťuje, že přeložená software odpovídá předem zvoleným možnostem.

Výsledkem překladu má příponu souboru .hex a .eep.

Soubory se obvykle nazývají ComponentTester.hex a ComponentTester.eep.

Soubor .hex obsahuje data pro programovou paměť (Flash) procesoru ATmega.

Soubor .eep obsahuje data pro EEPROM ATmega. Oba soubory musí být načteny do správného úložiště.

Navíc musí být u ATmega nakonfigurovány správně pojistky. Pokud používáš Makefile spolu s programem avrdude [5], nepotřebuješ mít žádnou přesnou znalost detailů pojistek.

Pokud si nejsi s nastavením pojistek jistý, nech je na poprvé nastavit standartě a nech tester běžet v tomto režimu. Když používáš 8MHz operační takt je možné, že program běží příliš pomalu, to ale můžeš to opravit později!

Nesprávně nastavené pojistek však mohou zabránit pozdějšímu ISP programování.

### 7.3. Operační systém Linux

Programování pod Linuxem přináší mnoho výhod, protože tento OS byl vyvinut odborníky, kteří se orientují přáním uživatelů.

Prostředí je navíc k dispozici zdarma a je dokonale udržováno. Další výhodou je zabezpečení samotného operačního systému, hlavně při používání internetu. Jak používání, tak i instalace dnešních vydání je mnohem jednodušší než u konkurenčních operačních systémů.

Tento tutoriál je tak navržen, aby povzbudil všechny „ne“ uživatele Linuxu, aby se o tom, naprogramování svého testeru v Linuxu, přesvědčili.

Jako příklad, je zde použitý Linux Mint v aktuální verzi, která je bezplatně k dispozici na internetu. Instalace je možná na různé způsoby, Linux přinese svého spouštěcího asistenta, který se samostatně předchozí OS respektuje a nakonfiguruje.



## 7.4. Použití s Linuxem

jako nově instalovaný operační systém.

Pro ty, kteří neradi píší, nabízí Linux snadný způsob, jak si to ulehčit.

Zkopíruj tuto příručku na USB klíčenku a otevři ji v tvém Linuxu.

Poté přesuň myš na název dokumentu, stiskni levé myší tlačítko a táhni dokument k levému okraji obrazovky, až se zobrazí možný rámeček. Nyní myš uvolni.

Příručka nyní zabere levou polovinu obrazovky.

V dalším kroku se současně stiskni **Strg** + **Alt** + **t** k otevření příkazového okna, které již známým způsobem přesuneš, nyní na pravou polovinu obrazovky.

## 7.5. Instalace programových balíčků

s připojeným a aktivním internetem,

musíš nejprve stáhnout s internetu a nainstalovat programové balíčky:

'binutils-avr', 'avrdude', 'avr-libc' a 'gcc-avr'. Dále správu verzí 'git'.

Toho dosáhneš jednoduše, když přejdeš na této stránce k následujícímu textu:

```
sudo apt-get install avrdude avr-libc binutils-avr gcc-avr git
```

Označ levým myším tlačítkem ten výše jmenovaný text v levém okně,

Přesuň myš na kurzor v pravém příkazovém okně a stiskni prostřední tlačítko myši (rolovací kolečko) **dále zkráceně ST**. Tím kopíruješ text mezi okny.

Po potvrzení pomocí **Enter**, vyžaduje 'sudo' tvé uživatelské heslo.

Tím se automaticky nainstalují všechny potřebné softwarové balíčky.

Eventuálně musíš mezitím potvrdit možnou otázku pomocí **J**.

Zapamatuj si, že Linux vždy rozlišuje mezi malými a velkými písmeny.

Takže neodpovídej s **j**, ale s **J**!

Zda byl úspěšně nainstalován systém pro správu verzí git, lze ověřit příkazem:

```
git version
```

Program by měl odpovědět výstupem s číslem své verze.

## 7.6. Stáhnutí zdrojů

a dokumentace

z archivu Git se dosáhne pomocí příkazu:

```
git clone https://github.com/kubi48/TransistorTester-source
```

Soubory jsou nyní v Linuxu [Osobní složka] na (/home/ „user“ / „TransistorTester-source“.

Kontrola přítomnosti. Otevři okno terminálu, zadej

```
ls
```

a potvrď s **Enter** nebo **↵**.

Pro stažení nových aktualizací stačí v budoucnu zadat:

```
cd ~/Tra
```

a následovně **tab** a **↵**, a nyní v tomto adresáři

```
git pull
```

a následovně potvrdit s **↵**.

**Poznámka:** V případě problémů nebo chybových hlášení je nejjednodušší, stávající složku "TransistorTester-source" odstranit nebo přejmenovat, a poté celý archiv znovu stáhnout podle výše uvedených pokynů.

## 7.7. Používání rozhraní

... připravit uživatele (user).

USB zařízení lze zjistit zadáním 'lsusb' v příkazovém okně. Zadej 'lsusb' nejprve bez a potom s připojeným USB programátorem.

Porovnáním výsledků najdeš tvůj USB programátor. Výsledek lsusb může vypadat takto:

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 003: ID 046d:c050 Logitech, Inc. RX 250 Optical Mouse
Bus 002 Device 058: ID 03eb:2104 Atmel Corp. AVR ISP mkII
Bus 002 Device 059: ID 2341:0042 Arduino SA Mega 2560 R3 (CDC ACM)
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub}
```

Zde byl detekován jako zařízení 58 AVR ISP mkII (DIAMEX ALL-AVR). ID 03eb je ID výrobce a ID 2104 je ID produktu.

Tyto dva identifikátory jsou potřebné na zapsání v souboru /etc/udev/rules.d/90-atmel.rules zadáním:

```
sudo xed /etc/udev/rules.d/90-atmel.rules
```

V tomto příkladu se soubor 90-atmel.rules skládá z jednoho řádku:

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="2104", MODE="0660", GROUP="plugdev"
```

Tato položka umožňuje přístup k zařízení pro členy skupiny 'plugdev'.

Chceš-li použít většinu programátorů, doporučuje se v 90-atmel.rules následující text:

```
# Copy this file to /etc/udev/rules.d/90-atmel.rules
# AVR ISP mkII - DIAMEX ALL-AVR
SUBSYSTEM=="usb", ATTRS {idVendor}=="03eb", ATTS {idProduct}=="2104", MODE="0660",
GROUP = "plugdev",
# USB ISP-programmer für Atmel AVR
SUBSYSTEM=="usb", ENV {DEVTYPE}=="usb_device", SYSFS {idVendor}=="16c0", MODE="0666",
SYSFS {idProduct} == "05dc",
# USB asp programmer
ATTRS {idVendor}=="16c0", ATTRS {idProduct}=="05dc", GROUP="plugdev", MODE="0660"
# USBtiny programmer
ATTRS {idVendor}=="1781", ATTRS {idProduct}=="0c9f", GROUP="plugdev", MODE="0660"
# Pololu programmer
SUBSYSTEM=="usb", ATTRS {idVendor}=="1fffb", MODE="0666"
```

Po vytvoření souboru lze tvorbu a obsah kontrolovat pomocí:

```
less /etc/udev/rules.d/90-atmel.rules
```

Systém USB zařízení Arduino SA Mega 2560, s 'lsusb' známý jako Device 59, generuje přístup k sériovému zařízení „/dev/ttyACM0“ pro členy skupiny „dialout“.


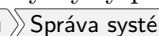
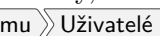
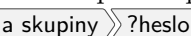
## 7.8. Členství ve skupině

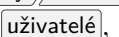

pro tvé vlastní uživatelské jméno, ve skupinách 'plugdev' i 'dialout' dosáhneš příkazem:

```
sudo usermod -a -G dialout,plugdev $USER
```

Nyní by měl být možný přístup s avrdude k objem zařízení. Můžeš to kontrolovat příkazem: 'id'.

Pokud by se vyskytly problémy, můžeš také přistoupit k členství prostřednictvím:


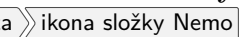
    se zobrazí okno se dvěma záložkami.

Pokud nyní klikneš na své jméno na kartě , uvidíš svůj profil a skupinové přidružení na pravé straně. Pomocí tlačítka  je nyní možné, přidat nové skupiny.

## 7.9. pracovní prostředí

příprava.

Aby se zachoval originál a protože se terminálové okno vždy otevírá v ../home/"user", nabízí se tam přesunout svůj pracovní adresář s názvem **Mytester**.


Nejdříve naviguj pomocí   do /TransistorTester-source/Markus.

Jako druhé klikni pravým tlačítkem na ComponentTester-1.(nejvyšší číslo)m.tgz a ve výběru <rozbalte zde> složku dekomprimuj. Nemo zase zavři.

Za třetí označ následující adresář, již známou metodou, a vlož do okna terminálu s :




```
cd TransistorTester-source/Markus/
```

Po potvrzení a zadání 'ls' se zobrazí všechny složky s příponou.tgz, pouze u jedné složky tato přípona chybí -> naše (právě rozbalená) složka.

Pro následující dva příkazy nejprve **JEN** vlož do terminálového okna **bez** stisknutí !:

```
cp -r 'MyT' Mytester/
```

Označ myši ten nahoře právě rozbalený adresář.

Nyní umísti, pomocí  klávesnice, blikající kurzor za poslední znak textu „MyT“ a tyto znaky vymaž. Po odstranění posledního znaku stiskni  na myši. Teprve nyní použij . Tím jsi vytvořil pracovní prostředí. Kontrola existence a obsahu je možná pomocí:

diff 'MyT' Mytester/  
také zde musí být „MyT“ nahrazeno jménem „požadovaného modelu testeru“. S posledním výrokem:

```
ln -s ~/TransistorTester-source/Markus/Mytester ~/Mytester
```

vytvoříš odkaz na pracovní adresář.

Od této chvíle se dostaneš lehce do tohoto adresáře pomocí:

```
[Strg]+[Alt]+[t], cd [mezerník] My [Tab] [Enter]
```

a jseš v požadovaném adresáři. S 'ls' můžeš vidět jeho obsah.

Nyní pokračuj v úpravách Makefile pomocí již známého příkazu:

```
xed Ma [Tab] [Enter]
```

**Zde je nejdůležitější přihlásit svůj EXISTUJÍCÍ USB Programátor.**

## 7.10. Přeložení Firmware

Po úpravě makefile, config.h nebo config-<MCU>.h udělej “make” nebo cokoli, co chce tvoje IDE k přeložení firmware.

Výsledkem je vytvoření dvou souborů:

- ComponentTester.hex firmware ve formátu Intel Hex
- ComponentTester.eep EEPROM data ve formátu Intel Hex

Firmware je zapsán do FLASH a EEPROM-data do EEPROM.

Data obsahují dvě sady standardních hodnot nastavení, texty a tabulky.

Pokud chceš jen software aktualizovat a své staré hodnoty nastavení v paměti EEPROM ponechat, můžeš použít přepínač DATA\_FLASH v config.h k přesunutí textů a tabulek do firmwaru.

V tomto případě bude **jen** firmware zapsána do FLASH a EEPROM zůstává nezměněn.

Makefile nabízí následující cíle:

clean	smazání všech souborů objektů Set
make	zkompileovat program
make fuses	nastavit bitové pojistky (přes avrdude)
make upload	vypálit firmware a EEPROM data (přes avrdude)
make prog_fw	vypalovat pouze firmware (přes avrdude)
make prog_ee	vypálit pouze EEPROM data (přes avrdude)

**Poznámky** k optimalizacím překladače/linkovače:

*V Makefile je několik řádků CFLAGS a LDFLAGS s volbami kompilátoru a linkeru. Jsou zde také zakomentované řádky s dalšími optimalizačními volbami, které mohou snížit velikost firmwaru, ale nejsou používány všemi kompilátory. Zde je nutné experimentovat ;-)*

**Poznámky** ke speciálním nastavení v souboru Makefile:

*- V prostředí Linux/Unix lze aktivovat funkci OPTIMIZE\_VECTORS, která optimalizuje tabulku vektorů přerušení a tím snížit velikost firmwaru.*

Nyní zbývá jen radost po dosaženém úspěchu.

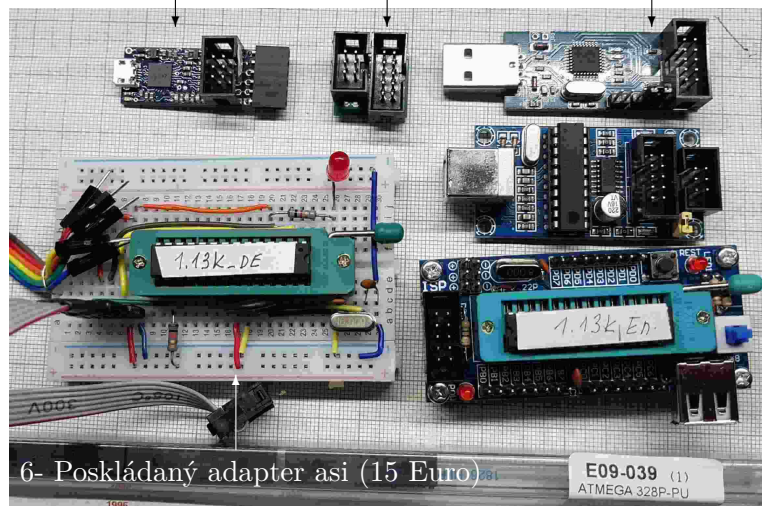
## 7.11. Hardware k programování

Pro úplné začátečníky, kteří ještě nic nemají následující informace.

**7.11.1. Programátor** nepotřebuje v Linuxu ovladač. Jen ho musíš podle části 7.7 přihlásit a v Makefile 6.3.5 správně nastavit.

Výhodou 'avrdude' je, že se spokojí i s lacinými programátory, které koupíš již za 2 Euro.

- 1- Pololu asi (13 Euro)      2- Souprava, kabel 10 pólů (není na obrázku),  
Adapter 6x10 pinů a USBasp dohromady pod (2 Euro)



Obrázek 7.1. Různé hardwarové možnosti. Pro každou kapsu ...;-)

Jak vidíš s obrázkem 7.1 tak potřebuješ jen

- (2)- programátor
  - (4)- malou vývojovou desku
  - (5)- a pro jistotu náhradní ATmega 328 P.
- To dostaneš dohromady pod 10 Euro.

- (6)- Je poskládaná deska, ale kdybys měl ty použité součástky koupit, tak se to nevyplatí.

### 7.11.2. Možnosti nákupu:

- **Programátor:** například na [9].
- **Vývojová deska:** například na [10].
- **A jestli ještě žádný tester nevlastníš, tak kup raději:** například na [11].

Tak nyní s chutí do toho.

Model	GM 328 A
Velikost	78 x 63 x 28 mm
Druh součástek	SMD
AVR	ATmega 328P
Krystal	8 MHz
Displej	ST7735 (128 x 160 pixelů)
IDE možné	ne
Ovládání	Rotační snímač s integrovaným tlačítkem
Napájení	9V blok
Spotřeba v provozu	
Spotřeba standby	20 nA
Měřicí napětí	5V
Měřicí proud	6 mA
Určení a měření	Tranzistory, MOSFET, JFET, P-IGBT, diody, Tyristory a triaky
Určení a měření	Odpory, kondenzátory, cívky
Určení a měření	Podle instalované SW různé další možnosti
Měření frekvence	1 Hz - 2 MHz
Generování frekvence	1 Hz - 2 MHz
Generování impulsů	Při 8 MHz taktu = frekvence 7,8 kHz. Impuls 1% - 99%
Měření napětí	0V - 50 V
Rozsah odpory	0,01 -
Rozsah kondenzátory	1pF - 100mF
Rozsah cívky	0,01mH -

Tabulka 8.1. Technické údaje

### 8.1. Pomoc a otázky

Při potížích se můžeš obrátit na německý veb [6].

V angličtině to můžeš skusit na [7]

a nebo na slovenskou pomoc ve Svetelektro [8].

### 8.2. A pro chvílku oddechu

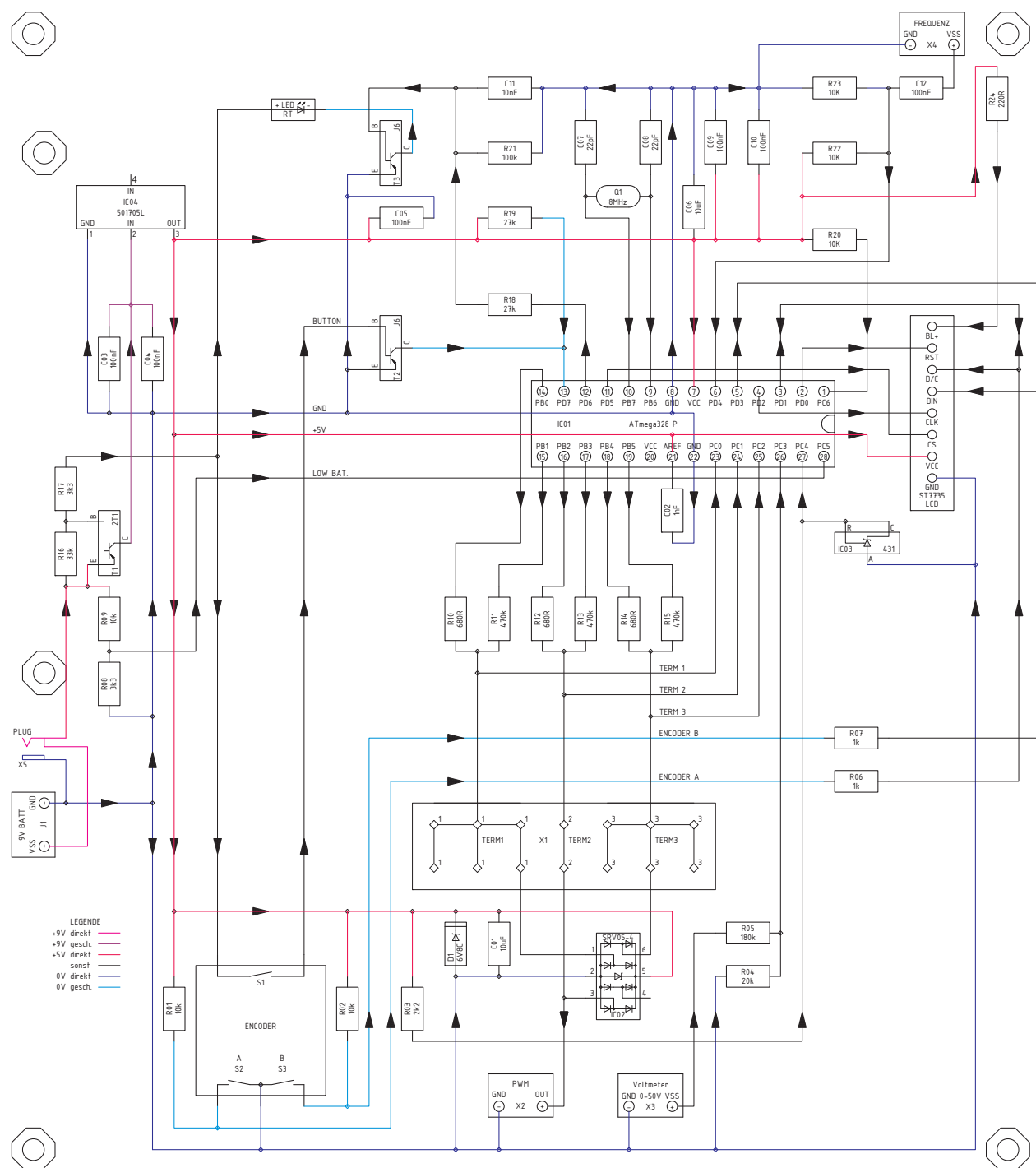
a nebo pro zabavení potomstva: [12].

### 8.3. Schema GM 328 A

**Pozor!** Následující schema zobrazuje propojení součástek bez záruky na úplnost.

Schema nemá žádné měřítko.

Vestavěné součástky byly měřeny tímto testerem, při čemž nebylo možné, zjistit kapacitu většiny kondenzátorů.



Obrázek 8.1. Schema propojení součástek

jak je viditelné, je zde velmi jednoduché, vyměnit PWM konektor (X2) za třípolový k zapojení pevných testovacích kabelů.



---

## *Literatura*

---

- [1] <http://www.mikrocontroller.net/articles/AVR-Transistortester>  
*Online Dokumentace Transistortesteru*, Online Article, 2009-2011
- [2] <https://github.com/kubi48/TransistorTester-documentation>  
*Aktuální dokumentace k Transistor testeru*
- [3] <https://github.com/madires/Transistortester-Warehouse>  
*Kompletní dokumentace i zdroj*
- [4] <https://github.com/kubi48/TransistorTester-source>  
*Kompletní zdroj softwaru*
- [5] <http://www.mikrocontroller.net/articles/AVRDUDE>  
*Online Dokumentace avrdude IDE*
- [6] <https://www.mikrocontroller.net/topic/248078>  
*Hlavní řeč je němčina, anglicky je ale také ok.*
- [7] [https://www.eevblog.com/forum/testgear/\(dolarováznačka\)  
20-lcr-esr-transistor-checker-project/](https://www.eevblog.com/forum/testgear/(dolarováznačka)20-lcr-esr-transistor-checker-project/)  
*Jen anglicky.*
- [8] <https://svetelektro.com/> *Všetko zo sveta elektroniky  
Sme najnavštevovanejší portál zo zameraním na elektroniku na Slovensku! od 2006*
- [9] [https://www.ebay.de/itm/usbasp-avrisp-usbisp-Programmer-usb-10Pin-  
-Convert-to-6P-Adapter-Board-STK50-AHS/302923364644](https://www.ebay.de/itm/usbasp-avrisp-usbisp-Programmer-usb-10Pin-Convert-to-6P-Adapter-Board-STK50-AHS/302923364644)  
*Programmer.*
- [10] [https://https://www.ebay.de/itm/335327569081?itmmeta=  
01HXF2CAJ87N4E5YAMYEMGXK8B&hash=item](https://https://www.ebay.de/itm/335327569081?itmmeta=01HXF2CAJ87N4E5YAMYEMGXK8B&hash=item)  
*Vývojová deska*
- [11] [www.aliexpress.com/item/4000069589587.html](http://www.aliexpress.com/item/4000069589587.html)  
*Hiland644 Tester*
- [12] <https://dragaosemchama.com/en/2017/01/rex/>  
*Hra tetrís pro tester a další*