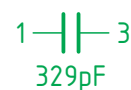
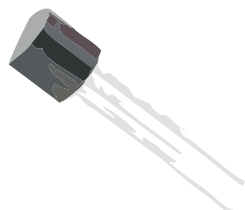


Popis Component tester Přístroj k určení a měření elektronických součástek, s mnoha dalšími funkcemi



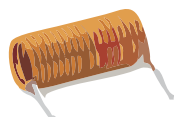
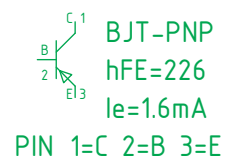
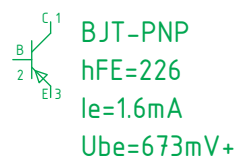
Verze 1.47-m



Markus Reschke

© 2012

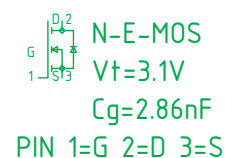
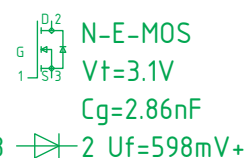
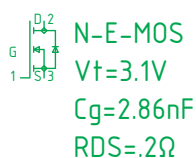
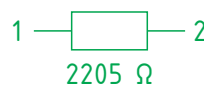
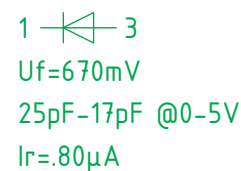
madires@theca-tabellaria.de



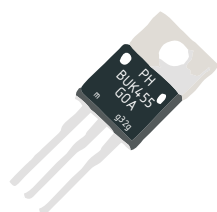
zkompileováno
od bm-magic



15. června 2022



13.05.2019/MOR



Obsah

1	O tomto Ctesteru	7
1.1	Úvod	7
1.2	Bezpečnost	7
1.3	Licence	7
1.3.1	Dostatečné licenční upozornění	7
1.4	Rozdíly ke k-verzi	8
1.5	Zdrojový kód	8
1.6	Podporovaná hardware	9
1.7	Podporování hardwareové možnosti	9
1.7.1	Obsluha	9
1.7.2	Vylepšení	9
1.7.3	Doplňkové zkoušky a měření	9
1.8	Upravení firmware	9
2	Hardware	10
2.1	Hardwareové možnosti	10
2.2	A samozřejmě softwarové možnosti	10
2.3	Dostupné jazyky	11
2.4	Konfigurace	11
2.5	Sběrnice & rozhraní	11
2.5.1	I2C/SPI	11
2.5.2	Serielní TTL Interface	11
2.5.3	One Wire	12
2.6	LCD moduly	12
2.6.1	Znakové sady a symboly	13
2.6.2	HD44780	13
2.6.3	ILI9163	13
2.6.4	ILI9341/ILI9342	14
2.6.5	ILI9481	15
2.6.6	ILI9486	15
2.6.7	ILI9488	15
2.6.8	PCD8544	16
2.6.9	PCF8814	16
2.6.10	SH1106	16
2.6.11	SSD1306	17
2.6.12	ST7036	17
2.6.13	ST7565R	18
2.6.14	ST7735	18
2.6.15	ST7920	18
2.6.16	STE2007/HX1230	18
2.6.17	VT100 Terminal	19
2.7	Tlačítka a ovládání	19
2.7.1	Testovací tlačítko	19

2.7.2	Rotační kodér (Hardwarová možnost)	19
2.7.3	(Vice/méně tlačítek)	19
2.7.4	Dotyková obrazovka (Hardwarová možnost)	19
2.7.5	Uživatelské rozhraní	20
2.7.6	Bzučák	20
2.7.7	Komunikace s PC	21
2.7.8	Seriový výstup	21
2.7.9	Automatisierung	21
2.7.10	Výstup přes VT100	21
3	Obsluha	22
3.0.1	Zapnutí	22
3.0.2	Hledání komponentů	22
3.0.3	Monitorování baterie	22
3.0.4	Vypínání,	22
3.1	Výběrové menu	22
3.1.1	PWM-Generátor	23
3.1.2	Jednoduchý PWM	23
3.1.3	Rozšířený PWM	23
3.1.4	Obdélníkový signální generátor	23
3.1.5	Zjištění Zenerového napětí (hardwarová úprava)	23
3.1.6	Logický tester (hardwarová úprava)	24
3.1.7	tester kontinuity	24
3.1.8	ESR-Tool	25
3.1.9	Unikající proud kondenzátoru	25
3.1.10	R/C/L Monitory	25
3.1.11	L/C-Měřič (hardwarová úprava)	25
3.1.12	Čítač kmitočtů (hardwarová úprava)	26
3.1.13	Jednoduchý čítač	26
3.1.14	Rozšířený čítač	26
3.1.15	Kroužkový tester (hardwarová úprava)	26
3.1.16	Počítadlo událostí (s hardwarovou úpravou)	27
3.1.17	Rotační kodér	27
3.1.18	Kontrast	27
3.1.19	Detektor/Dekodér pro IR dálkové ovládání	27
3.1.20	IR dálkové ovládání	28
3.1.21	Test optických spojek	29
3.1.22	Test servopohonů pro modely	30
3.1.23	OneWire skenování	30
3.1.24	Snímač teploty DS18B20	31
3.1.25	DHTxx senzory	31
3.1.26	Převodník termočlánků	31
3.1.27	Autotest	32
3.1.28	Samočinné nastavení	32
3.1.29	Ušchovat/Použít	33
3.1.30	Ukázat hodnoty	33
3.1.31	Znaková sada/symboly	33
3.1.32	Vypnout	33
3.1.33	Konec	33
4	Detaily měření	34
4.1	Odpory	34
4.2	Kondenzátory	34
4.3	tlumivky	35
4.4	Vybití součástek	35

4.5	ADC převzorkování	35
4.6	Zobrazení výsledků	35
4.7	Dodatečné pokyny	36
4.8	užitečné odkazy	38
4.9	Změny firmwaru	38
5	Dálkové ovládání	39
5.1	Příkazy dálkového ovládání	39
5.1.1	ERR	39
5.1.2	OK	39
5.1.3	N/A	39
5.2	Basiskommandos	39
5.2.1	VER	39
5.2.2	OFF	39
5.3	Testkommandos	39
5.3.1	PROBE	39
5.3.2	COMP	39
5.3.3	MSG	39
5.3.4	QTY	39
5.3.5	NEXT	39
5.3.6	TYPE	39
5.3.7	HINT	40
5.3.8	MHINT	40
5.3.9	PIN	40
5.3.10	R	40
5.3.11	C	40
5.3.12	L	40
5.3.13	ESR	40
5.3.14	I_l	41
5.3.15	V_F	41
5.3.16	V_F2	41
5.3.17	C_D	41
5.3.18	I_R	41
5.3.19	R_BE	41
5.3.20	h_FE	41
5.3.21	h_FE_r	41
5.3.22	I_C	41
5.3.23	I_E	41
5.3.24	V_BE	41
5.3.25	I_CEO	41
5.3.26	V_th	41
5.3.27	C_GS	41
5.3.28	R_DS	41
5.3.29	V_GS_off	42
5.3.30	I_DSS	42
5.3.31	C_GE	42
5.3.32	V_GT	42
5.3.33	V_T	42
5.3.34	R_BB	42
5.3.35	V_Z	42
6	Programový kód	43
6.1	Makefile	43
6.1.1	MCU typ	43
6.1.2	Taktová frekvence MCU	43

6.1.3	Typ oscilátoru	43
6.1.4	Avrdude typ MCU	44
6.1.5	Avrdude ISP programátor	44
6.2	config.h	45
6.2.1	Hardwarová obsluha	45
6.2.2	Možnosti softwaru	48
6.2.3	Řešení obcházení pro některé testery	50
6.2.4	Řešení pro některá IDE	50
6.2.5	uživatelské rozhraní	51
6.2.6	Správa napájení	53
6.2.7	Nastavení a kompenzace měření	54
6.2.8	R&D - určeno pro vývojáře firmwaru	55
6.2.9	Busse	56
6.3	Config_328.h	57
6.3.1	LCD moduly	57
6.3.2	Rozložení portů a pinů	57
6.3.3	Busse	59
6.4	Config_644.h	60
6.4.1	LCD moduly	60
6.4.2	Rozložení portů a pinů	60
6.4.3	Busse	63
6.5	Config_1280.h	64
6.5.1	LCD moduly	64
6.5.2	Rozložení portů a pinů	64
6.5.3	Busse	67
7	kolekce nastavení	68
7.1	Arduino Nano, Uno nebo Mega 2560	68
7.2	DIY Kit „AY-AT“ platí také pro GM_328A	68
7.3	BSide ESR02	70
7.4	Fish8840 TFT	71
7.5	GM328 !pozor! né GM328A	71
7.6	Hiland M644	72
7.7	M12864 DIY Transistor Tester	73
7.8	MK-328	74
7.9	T3/T4	74
7.10	Multifunktionstester TC-1 a rodina (T7) s ATmega324/644	76
7.11	Multifunktionstester T7 s ATmega328	78
7.12	Arduino MEGA zapojení	79
8	Programování testeru Componentů	80
8.1	Konfigurace testeru	80
8.2	Programování testeru	80
8.3	Operační system Linux	80
8.4	Použití s Linuxem	81
8.5	Instalace programových balíčků	81
8.6	Stáhnutí zdrojů	81
8.7	Používání rozhraní	81
8.8	Členství ve skupině	82
8.9	pracovní prostředí	82
8.10	Přeložení Firmware	83
9	Postup verzí	84
9.1	v1.47m 2022-07	84
9.2	v1.46m 2022-04	85

9.3	v1.45m	2021-12	85
9.4	v1.44m	2021-08	85
9.5	v1.43m	2021-03	86
9.6	v1.42m	2020-12	86
9.7	v1.41m	2020-09	86
9.8	v1.40m	2020-06	87
9.9	v1.39m	2020-03	88
9.10	v1.38m	2019-12	88
9.11	v1.37m	2019-09	89
9.12	v1.36m	2019-05	89
9.13	v1.35m	2019-02	89
9.14	v1.34m	2018-10	90
9.15	v1.33m	2018-05	90
9.16	v1.32m	2018-02	90
9.17	v1.31m	2017-12	90
9.18	v1.30m	2017-10	91
9.19	v1.29m	2017-07	91
9.20	v1.28m	2017-04	91
9.21	v1.27m	2017-02	91
9.22	v1.26m	2016-12	92
9.23	v1.25m	2016-09	92
9.24	v1.24m	2016-08	92
9.25	v1.23m	2016-07	92
9.26	v1.22m	2016-03	93
9.27	v1.21m	2016-01	93
9.28	v1.20m	2015-12	93
9.29	v1.19m	2015-11	93
9.30	v1.18m	2015-07	93
9.31	v1.17m	2015-02	93
9.32	v1.16m	2014-09	93
9.33	v1.15m	2014-09	94
9.34	v1.14m	2014-08	94
9.35	v1.13m	2014-07	94
9.36	v1.12m	2014-03	94
9.37	v1.11m	2014-03	94
9.38	v1.10m	2013-10	94
9.39	v1.09m	2013-07	95
9.40	v1.08m	2013-07	95
9.41	v1.07m	2013-06	95
9.42	v1.06m	2013-03	95
9.43	v1.05m	2012-11	95
9.44	v1.04m	2012-11	95
9.45	v1.03m	2012-11	96
9.46	v1.02m	2012-11	96
9.47	v1.01m	2012-10	96
9.48	v1.00m	2012-09	96
9.49	v0.99m	2012-09	96

1.1. Úvod

Transistor Tester je založen na projektu Markuse Frejka [1] a [2] s pokračováním Karl-Heinze Kübbelerem [3] a [4].

Componenten tester je alternativní firmware pro současný testovací obvod Karl-Heinze a nabízí některé změny uživatelského rozhraní a postupy měření a testování.

Zatímco firmware Karl-Heinze je oficiální verze a podporuje starší typy MCU ATmega, slouží tato verze k vyzkoušení a testování nových nápadů.

Je omezena na ATmegasy s minimálně 32kB Flash.

Primární jazyky pro tuto verzi jsou angličtina a němčina, ale lze snadno doplnit další jazyky.

Poznámka: Proveď vlastní nastavení u zcela nových testerů nebo firmwaru aktualizace.
Nebo při použití jiných měřicích kabelů.

1.2. Bezpečnost

Component Tester není multimetr!

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

Vstupy nejsou chráněny a budou poškozeny napětím nad 5V.

Nepoužívej tester pro obvody v provozu, ale pouze pro jednotlivé komponenty! 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 já jsme se dosud k autorovi nedostali.

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

Vzhledem k tomu že tato verze firmwaru je zcela nová verze, který zabírá 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 ke k-verzi

Karl-Heinz napsal k testeru opravdu dobrou dokumentaci. [17]
Určitě si ji přečti! Proto popisují pouze hlavní rozdíly k firmwaru k:

- uživatelské rozhraní

Nepropadej v panice! ;-)

+ Dotykový displej

+ Automatizace (příkazy pro dálkové ovládání)

+ dvě sady hodnot nastavení

- Adaptivní funkce vybíjení - Měření odporu

+ další metoda pro odpory $< 10 \Omega$ (místo měření ESR)

- Měření kapacity + od 5pF

+ další metoda pro kondenzátory mezi 4,7 μ F a 47 μ F

+ další metoda korekce / kompenzace

- Žádný SamplingADC () pro velmi nízké kapacity nebo indukčnosti

- diody

+ Logika rozpoznávání

- bipolární tranzistory

+ V_{ce} je interpolováno pro praktické zkušební proudy

+ Detekce germaniových tranzistorů s vyšším svodovým proudem

+ Detekce Schottkyho tranzistorů

JFET

+ Detekce JFET s velmi nízkými I_{DSS}

- TRIAC

+ Detekce MT1 a MT2

- detektor / dekodér pro IR dálkové ovladače

- IR dálkové ovládání

- Test optokoplerů

- Test serva na výrobu modelů

- OneWire (DS18B20)

- Řada sensorů (DHTxx)

- Počítadlo událostí

- Bzučákový tester

- Logický tester

- Zkoušečka kontinuity

- Převodník termočlánků MAX6675/MAX31855

- Strukturovaný zdrojový kód

- Jednoduché rámce pro displeje a datové sběrnice

- A další, na které nemůžu myslet.

Další podrobnosti najdeš v následujících sekcích.

1.5. Zdrojový kód

První m-firmware byla založena na zdrojovém kódu od Karla Heinze. Hodně bylo vymazáno, komentáře, proměnné přejmenovány, restrukturalizované funkce, velké funkce rozděleny na několik malých a více. Pak jsem pokračoval ve vývoji m-firmwaru jako nezávislou verzi. Existují jednoduché rámce pro displej a přišly k tomu i nové rozhraní.

Doufám, že zdrojový kód je snadno čitelný a srozumitelný.

Aktuální firmware najdeš na následujících webech:

[5] <https://github.com/madires/Transistortester-Warehouse>

[6] <https://github.com/kubi48/TransistorTester-source/tree/master/Markus>

[7] <https://github.com/Mikrocontroller-net/transistortester/tree/master/Software/Markus>

1.6. Podporovaná hardware

Firmware běží na všech testerech, které poskytují základní obvody jako u Karl-Heinze. Dokumentace je kompatibilní a používá jednu z následujících MCU:

- ATmega 328 config_328.h
- ATmega 324/644/1284 config_644.h
- ATmega 640/1280/2560 config_1280.h

Je možné upravit různá přiřazení pinů. Jako displej je možné používat textové nebo grafické displeje (monochromatické nebo barevné).

Podporované řadiče najdeš v kapitole 2.6 „LCD moduly“ od stránky 45.

1.7. Podporování hardwarové možnosti

1.7.1. Obsluha

- rotační kodér
- další klíče (více / méně)
- Dotyková obrazovka
- sériové rozhraní (TTL, RS232, sériový adaptér USB)

1.7.2. Vylepšení

- externí referenční napětí 2.5V
- pevný nastavovací kondenzátor
- ochranné relé pro vybíjení kondenzátorů

1.7.3. Doplnkové zkoušky a měření

- Zenerův test / měření externího napětí <50V
 - jednoduchý čítač kmitočtu
 - čítač rozšířené frekvence
- s předzesilovačem a křemenným oscilátorem pro nízké a vysoké frekvence
- pevný IR-RC přijímač
 - LC metr
 - Kroužkový tester (podle Boba Parkera)
 - Logický tester
 - Převodník termočlánků MAX6675/MAX31855

1.8. Upravení firmware

Nejprve bys měl upravit **Makefile** viz kapitola 6.1 na stránce 43 a zde upravit nastavení MCU model, frekvenci, typ oscilátoru a nastavení programovacího adaptéru.

V **config.h** viz kapitolu 6.2 od stránky 45 můžeš nastavit možnosti operace a nabídky, Zde si vybereš hardwarové a softwarové možnosti, jazyk provozu a v případě potřeby změníš výchozí hodnoty.

A nakonec v **config_<MCU>.h** globální konfiguraci MCU například přiřazení pinů a zobrazení které se liší v závislosti na vestavěném MCU. Detaily v kapitole 6.5 od stránky 64.

Soubory config<MCU>.h platí pro následující MCU:

- ATmega 328 config_328.h
- ATmega 324/644/1284 config_644.h
- ATmega 640/1280/2560 config_1280.h

V kapitole 7, “Kolekce nastavení” najdeš, od stránky 45, nastavení pro různé verze testerů.

Pokud tam tvůj tester není, pošli tvé nastavení e-mailem autorovi [8] pro pomoc ostatním uživatelům.

Všechna nastavení a hodnoty jsou vysvětleny v samotném souboru,

Jak Firmware přeložit se dozvíš v oddílu 8.10 na straně 83.

2.1. Hardwarové možnosti

- další klávesy nebo možnosti vstupu
- Rotační kodér
- Více / méně tlačítek
- Dotyková obrazovka
- Externí referenční napětí 2.5V
- Ochranné relé pro vybití kondenzátoru
- Měření zenerových diod (převodník DC-DC)
- Čítač kmitočtů (jednoduchá a rozšířená verze)
- Čítač událostí
- LC Měřič
- Test IR dálkového ovládání (pevný IR přijímací modul)
- Pevný kondenzátor pro samočinné nastavení kompenzace napětí
- Sběrnice SPI (bit-bang a hardware)
- Sběrnice I2C (bit-bang a hardware)
- TTL Serial (Bit-Bang and Hardware)
- OneWire Bus (Bit-Bang)

Externí referenční napětí 2,5 V by mělo být použito, pouze pokud je to faktor 10 přesnější než regulátor napětí. Jinak by to výsledky spíše zhoršilo, než by se zlepšilo. Pokud máš MCP1702 s typickou přesností 0,4% jako regulátor napětí opravdu nepotřebuješ další referenční napětí.

2.2. A samozřejmě softwarové možnosti

- PWM generátor (2 varianty)
- Měření indukčnosti
- ESR měření a ESR v obvodu
- Test rotačních kodérů
- Generátor čtvercové vlny (vyžaduje další tlačítka)
- Test dálkového ovládání IR (modul IR přijímače na testovacích pinech)
- IR dálkové ovládání (IR LED s tranzistorem ovladačem)
- Test optočlenů
- Test serva pro výrobu modelů (potřebuje další klávesy, displej > 2 řádky)
- OneWire-Detekce
- Detekce UJT
- Test těsnosti kondenzátoru
- Monitorování R/L/C
- Teplotní senzor DS18B20
- Řada senzorů DHTxx k měření teploty a vlhkosti
- Barevné kódování zkušebních pinů (vyžaduje barevné zobrazení)
- Výstup nalezených komponent paralelně přes TTL seriál, například na PC
- Příkazy pro dálkové ovládání pomocí TTL seriálu.
- Výstup reverzního hFE (C&E obráceny) pro bipolární tranzistory
- ...

z těch nabídek vyber podle svých představ a omezených zdrojů MCU, RAM, EEPROM a flash paměti.

2.3. Dostupné jazyky

- dánština (z adresy glennndk@mikrocontroller.net) potřebuje malé změny písma.
- němčina
- angličtina
- italština (od Gino_09@EEVblog)
- polština (od Szpila)
- rumunština (od von Dumidan@EEVblog)
- ruština (od indman@EEVblog)
 - Znaková sada s azbukou založená na systému Windows 1251
- ruština 2 (od hapless@EEVblog)
 - Znaková sada s azbukou založená na systému Windows 1251
- španělština (od pepe10000@EEVblog)
- čeština (od Kapa)
 - Znaková sada založená na systému ISO 8859-1
- čeština 2 (od Bohu)
 - Znaková sada založená na systému ISO 8859-2

Pokud dáváš přednost čárce místo desetinné tečky, existuje předvolba (Standart: Punkt).

2.4. Konfigurace

Tvoje MCU a programátor se nastavují v Makefile ... kapitola 6.1 od strany 43.

Tvé možnosti si můžeš vybrat v config.h ... kapitola 6.2 od strany 45.

Specifická nastavení, jako jsou přiřazení pinů a zobrazení, se liší v závislosti od MCU v souborech:

- ATmega 328 v config_328.h
- ATmega 324/644/1284 v config_644.h
- ATmega 640/1280/2560 v config_1280.h

Detaily si můžeš přečíst v ... kapitole 6.5 od strany 64.

Pokud bude firmware příliš velký, zkus znovu méně důležité možnosti vypnout.

2.5. Sběrnice & rozhraní

2.5.1. I2C/SPI Některé LCD moduly a další komponenty vyžadují na MCU I2C nebo SPI rozhraní. Firmware má proto ovladače pro oba systémy. Pro podporu různých obvodů mají ovladače sběrnic jeden Bit-Bang a jeden hardwarový režim. V režimu Bit-Bang mohou jakýkoli IO Piny používat stejný port, zatímco hardwarový režim používá pevně předepsané MCU Piny Busu. Nevýhodou Bit-Bang režimu je jeho pomalá rychlost. Hardwarový režim je hodně rychlejší. Ten rozdíl můžeš lehce pozorovat u barevných LCD modulů s vysokým rozlišením.

Pro testery s ATmega 328 je téměř vždy vyžadován Bit-Bang režim kvůli zapojení. ATmega 324/644/1284 má více I/O pinů, kromě toho dovoluje změněné zapojení používat pevné kolíky sběrnice pro hardwarový režim.

Protože je SPI nebo I2C primárně používán LCD modulem, najdeš oba v sekci pro moduly LCD v config<MCU>.h stránka 45 je můžeš přímo konfigurovat. Alternativně můžeš I2C nebo SPI v config.h na stránce 56 aktivovat a porty & piny v config<MCU>.h upravit. (Hledej I2C_PORT stránka 67 nebo SPI_PORT) na stránce 67.

Pokud je ke SPI sběrnici připojeno několik integrovaných obvodů, musí být každý z nich řízen svým vlastním /CS signálem. Pouze v případě jednoho IC na SPI sběrnici, lze tento /CS signál trvale spojit se zemí.

2.5.2. Sérielní TTL Interface Tester může mít také volitelné sériové rozhraní TTL. Používá se tato pro komunikaci s PC, mělo by to být kombinováno s USB na TTL převodníkem nebo s modulem ovladače RS-232. Firmware může použít UART MCU nebo softwarový UART (bit-bang). TTL rozhraní se aktivuje v config.h viz sekci "Buse" na straně 56 a Porty & Piny jsou definovány v config<MCU>.h (hledej SERIAL_PORT stránka 67).

Nevýhodou UART softwaru je, že TX signál není neustále "vysoký", když je rozhraní v klidu. To je způsobeno použitou metodou k ovládání Portových Pinů. Přepsání tohoto ovládání by výrazně zvýšilo obsah firmware. Zdá se, že tento problém nemá žádný účinek na většinu USB

na TTL převodníků. V případě, že bys měl přesto potíže, můžeš zkusit pull-up odpor (10-100k) na TX pinu k udržení “vysoké” úrovně signálu v klidovém režimu.

Výchozí nastavení sériového rozhraní je 9600 8N1:

- 9600 bps
- 8 datových bitů
- žádná parita
- 1 stop bit
- žádná kontrola toku

2.5.3. One Wire Další podporovaný bus je OneWire, který má buď zkušební kolíky (ONEWIRE_PROBES) nebo pevný pin MCU (ONEWIRE_IO_PIN), na staně 56. Ovladač je určen pro standardní rychlost sběrnice a klienty s externím napájením.

Zapojení zkušebních pinů:

sample # 1: Gnd

sample # 2: DQ (data)

sample # 3: Vcc (proud omezený odporem 680 Ω)

Je zapotřebí externí pull-up odpor 4,7 kΩ mezi DQ a Vcc!

Funkce, které vyžadují přesně jednoho klienta na sběrnici, mohou volitelně používat ROM Výstupní kód klienta (ONEWIRE_READ_ROM). V případě chyby CRC nebo pokud je na sběrnici připojeno více klientů, je výstup “-“. Je-li kompletní ROM kód nula, došlo k chybě čtení. Jinak první část ROM kódu je rodinu produktů a druhá část sériové číslo.

2.6. LCD moduly

V současné době jsou podporovány následující LCD moduly nebo ovladače:

- HD44780	(textové zobrazení, 2–4 řádky po 16–20 znakech)	s. 13
- ILI9163	(grafický barevný displej 128x160)	s. 13
- ILI9341/ILI9342	(grafický barevný displej 240x320 nebo 320x240)	s. 14
- ILI9481	(grafický barevný displej 320x480, částečně zkoušený)	s. 15
- ILI9486	(grafický barevný displej 320x480, částečně zkoušený)	s. 15
- ILI9488	(grafický barevný displej 320x480, částečně zkoušený)	s. 15
- PCD8544	(grafické zobrazení 84x48)	s. 16
- PCF8814	(grafické zobrazení 96x65)	s. 16
- SH1106	(grafické zobrazení 128x64)	s. 16
- SSD1306	(grafické zobrazení 128x64)	s. 17
- ST7036	(textové zobrazení, 3 řádky po 16 znacích)	s. 17
- ST7565R	(grafické zobrazení 128x64)	s. 18
- ST7735	(grafický barevný displej 128x160)	s. 18
- ST7920	(grafické zobrazení až 256x64)	s. 18
- STE2007 / HX1230	(grafické zobrazení 96x68)	s. 20
- Terminál VT100	(přes sériové rozhraní)	s. 19

Věnuj pozornost napájecímu napětí a logické úrovni modulu displeje! Použij Převaděč úrovně, pokud je to nutné. Jednoduchý řadič úrovně s odpory v sérii s využitím interních omezovacích diod displeje. Řadiče může fungovat, ale pouze pro pomalé busy, jako je Bit-Bang SPI.

Doporučuji tedy používat integrované obvody převodníku napětí.

Pokud displej, navzdory správnému zapojení nic nezobrazuje, zkus změnit kontrast v (config_<MCU>.h) viz 64.

Pokud chceš ušetřit IO piny, můžeš u většiny modulů LCD /CS a /RES signály pevně zapojit prostřednictvím pullup/down odporů a odpovídající IO piny vykomentovat, pokud na sběrnici visí pouze LCD modul.

Grafické displeje nabízejí obvykle nastavení orientace výstupu, např. otočení obrazu o 90 °. Lze je použít jak vodorovně, tak vertikálně. To umožňuje přizpůsobení obrazu příslušnému testerovi. Pro grafické barevné displeje jsou k dispozici další nastavení. V normálním barevném režimu lze použít několik barev, které lze změnit pomocí souboru color.h.

Dvoubarevný režim se aktivuje komentováním LCD_COLOR.

Barva písma je potom BARVA_PEN a barva pozadí je BARVA_BACKGROUND.

Pokud jsou primární barvy RGB červené a modré zaměněny, můžeš použít LCD_BGR k změně příslušných barevných kanálů. U některých displejů je pořadí dílčích pixelů RGB obrácené a ovladač o něm nic neví.

Poznámka k ATmega 328: Pokud je k PD2/PD3 připojen rotační kodér, připoj /CS od LCD modulu s PD5 a nastav LCD_CS v config_328.h (pouze pro grafické LCD moduly).

V opačném případě by rotační kodér pokazil datovou sběrnici a vedl k nesprávným výdajům.

2.6.1. Znakové sady a symboly

Konfigurace modulu displeje zahrnuje také znakovou sadu a volitelnou sadu symbolů.

Příslušné znakové sady a symboly jsou uvedeny v části nastavení pro každý modul v souboru config_<MCU>.h.

U grafických displejů si obvykle můžeš vybrat z několika znakových sad. Symboly se aktivují pomocí přepínače SW_SYMBOLS v souboru config.h.

Pokud se ti více líbí styl starých symbolů, můžeš je stále používat změnou nastavení symbolů na SYMBOLS_<size>_OLD_<format>.h.

Pro testovací účely můžeš aktivovat funkci nabídky, která pak vypíše všechny znaky v položce znaková sada (SW_FONT_TEST) nebo všechny symboly komponent (SW_SYMBOL_TEST).

2.6.2. HD44780 zapojený v 4-bytovém režimu.

Modul	config-<MCU>.h	328 75,6%	644 37,9%	1280 13,1%	odkaz
DB4	LCD_DB4	PD0	PB4	PB0	
DB5	LCD_DB5	PD1	PB5	PB1	
DB6	LCD_DB6	PD2	PB6	PB2	
DB7	LCD_DB7	PD3	PB7	PB3	
RS	LCD_RS	PD4	PB2	PB4	
R/W	LCD_RW	Gnd	Gnd	Gnd	
E	LCD_EN1	PD5	PB3	PB5	

Tabulka 2.1. Zapojení HD44780 v paralelním režimu.

Tento LCD modul lze také použít s I2C adaptérem PCF8574. Za tímto účelem musí být aktivován I2C a kromě toho je požadována adresa adaptéru.

Modul	config-<MCU>.h	328 76,5%	644 38,2%	1280 9,8%	odkaz
DB4	LCD_DB4		PCF8574_P4		
DB5	LCD_DB5		PCF8574_P5		
DB6	LCD_DB6		PCF8574_P6		
DB7	LCD_DB7		PCF8574_P7		
RS	LCD_RS		PCF8574_P0		
R/W	LCD_RW		PCF8574_P1		
E	LCD_EN1		PCF8574_P2		
LED	LCD_LED		PCF8574_P3		

Tabulka 2.2. Zapojení LCD-Modulu s PCF8574.

2.6.3. ILI9163 zapojený jako 4-drátový-SPI.

Modul	config-<MCU>.h	328 87,4%	644 43,8%	1280 11,2%	odkaz
/RESX	LCD_RES	PD4	PB2	PB4	možnost
/CSK	LCD_CS	PD5	PB4	PB5	možnost
D/CX	LCD_DC	PD3	PB3	PB7	
SCL	LCD_SCL	PD2	PB7	PB1	SPI Clock
SDIO	LCD_SDA	PD1	PB5	PB2	SPI MOSI

Tabulka 2.3. ILI9163 zapojení.

Pro správné zobrazení budeš možná muset použít nastavení X/Y otočení. V případě potřeby můžeš také posunout směr X. Je-li LCD_LATE_ON aktivován, začíná tester s vymazaným displejem, což vede ke krátkému zpoždění při zapnutí. Jinak se při zapnutí zobrazí krátce náhodné pixely.

2.6.4. ILI9341/ILI9342 je možné použít přes SPI nebo 8-bit-Paralel.

Modul	config-<MCU>.h	328 87,4%	644 51,7%	1280 13,2%	odkaz
RES	LCD_RES	PD4	PB2	PB4	možnost
CS	LCD_CS	PD5	PB4	PB5	možnost
DC	LCD_DC	PD3	PB3	PB7	
SCK	LCD_SCK	PD2	PB7	PB1	SPI Clock
SDI	LCD_SDI	PD1	PB5	PB2	SPI MOSI
SDO	LCD_SDO	-	PB6	PB3	nur ILI9341*
* zatím nepoužité					

Tabulka 2.4. 4-Line-SPI zapojení ILI9341/ILI9342.

Modul	config-<MCU>.h	328	644	1280 13,1%	odkaz
	LCD_PORT			PORTB	
/RES	LCD_RES			PB4	možnost
/CSX	LCD_CS			PB5	možnost
D/CX	LCD_DC			PB7	
WRX	LCD_WR			PB0	
RDX	LCD_RD			PB6	
	LCD_PORT2			PORTL	
D0	LCD_DB0			PL0	LCD_PORT2 Pin #0
D1	LCD_DB1			PL1	LCD_PORT2 Pin #1
D2	LCD_DB2			PL2	LCD_PORT2 Pin #2
D3	LCD_DB3			PL3	LCD_PORT2 Pin #3
D4	LCD_DB4			PL4	LCD_PORT2 Pin #4
D5	LCD_DB5			PL5	LCD_PORT2 Pin #5
D6	LCD_DB6			PL6	LCD_PORT2 Pin #6
D7	LCD_DB7			PL7	LCD_PORT2 Pin #7

Tabulka 2.5. 8-bit-Paralelní zapojení ILI9341/ILI9342.

Pro správné zobrazení budeš možná muset použít nastavení X/Y otočení. A nezapomeň na správné rozlišení X a Y. Nastavení (ILI9341 je 240x320 a ILI9342 je 320x240). Některé moduly displeje mají deaktivovanou rozšířenou sadu instrukcí ILI9341 (pin EXTC je zapnutý Masse), a výstup je pak často jen vágní. Toho lze opravit s LCD_EXT_CMD_OFF.

Vzhledem k vysokému počtu pixelů je výstup přes SPI trochu pomalý. Vymazání celého displeje trvá s Bit-Bang SPI asi 3 sekundy při 8MHz taktu MCU.

Lepší je použití hardwarového SPI nebo paralelní sběrnice.

2.6.5. ILI9481 lze použít jako 8-Bit-Paralel, 16-Bit-Paralel nebo 4-vodičový SPI.

Modul	config-<MCU>.h	328	644 51,6%	1280 13,2%	odkaz
	LCD_PORT			PORTB	
/RESX	LCD_RES		PC4	PB4	možnost
CSX	LCD_CS		PC3	PB5	možnost
D/CX	LCD_DC		PC2	PB7	
WRX	LCD_WR		PC1	PB0	
RDX	LCD_RD		PC0	PB6	možnost
	LCD_PORT2			PORTL	
D0	LCD_DB0		PB0	PL0	LCD_PORT2 Pin #0
D1	LCD_DB1		PB1	PL1	LCD_PORT2 Pin #1
D2	LCD_DB2		PB2	PL2	LCD_PORT2 Pin #2
D3	LCD_DB3		PB3	PL3	LCD_PORT2 Pin #3
D4	LCD_DB4		PB4	PL4	LCD_PORT2 Pin #4
D5	LCD_DB5		PB5	PL5	LCD_PORT2 Pin #5
D6	LCD_DB6		PB6	PL6	LCD_PORT2 Pin #6
D7	LCD_DB7		PB7	PL7	LCD_PORT2 Pin #7

Tabulka 2.6. ILI9481/ILI9486 8-Bit-Paralel

Modul	config-<MCU>.h	328	644	1280	odkaz
LCD_D8	LCD_DB8			PC0	LCD_PORT3 Pin #0
LCD_D9	LCD_DB9			PC1	LCD_PORT3 Pin #1
LCD_D10	LCD_DB10			PC2	LCD_PORT3 Pin #2
LCD_D11	LCD_DB11			PC3	LCD_PORT3 Pin #3
LCD_D12	LCD_DB12			PC4	LCD_PORT3 Pin #4
LCD_D13	LCD_DB13			PC5	LCD_PORT3 Pin #5
LCD_D14	LCD_DB14			PC6	LCD_PORT3 Pin #6
LCD_D15	LCD_DB15			PC7	LCD_PORT3 Pin #7

Tabulka 2.7. dodatečné zapojení pro 16-Bit-Paralel.

Modul	config-<MCU>.h	328 82,5%	644 39,1%	1280 10,0%	odkaz
RES	LCD_RES	PD4	PB2	PB4	možnost
CS	LCD_CS	PD5	PB4	PB5	možnost
D/C	LCD_DC	PD3	PB3	PB7	
SCL	LCD_SCL	PD2	PB7	PB1	SPI Clock
SDA	LCD_SDA	PD1	PB5	PB2	SPI MOSI

Tabulka 2.8. Přirazení pinů pro ILI9481 4-řádkové-SPI.

Vzhledem k vysokému rozlišení displeje a barevnému schématu RGB666 (3 bajty na pixel). SPI je poměrně pomalé, a to i v případě hardwarového SPI a 16 MHz taktu MCU. Proto nedoporučuji rozhraní SPI nepoužívat. Pro správné zobrazení je obvykle nutné displej otočit (LCD_ROTATE). V případě potřeby jej můžete také otočit o X a/nebo Y.

2.6.6. ILI9486 je řízena 8-bitová paralelní sběrnice, 16-bitová paralelní sběrnice nebo 4-řádkový SPI a používá stejné vývody jako ILI9481.

2.6.7. ILI9488 je řízena 8-bitová paralelní sběrnice, 16-bitová paralelní sběrnice nebo 4-řádkový SPI a používá stejné vývody jako ILI9481.

2.6.8. PCD8544 je řízen pomocí SPI.

Modul	config-<MCU>.h	328 82,5%	644 39,1%	1280 10,0%	odkaz
/RES	LCD_RES	PD4	PB2	PB4	možnost
/SCE	LCD_SCE	PD5	PB4	PB5	možnost
D/C	LCD_DC	PD3	PB3	PB7	
SCLK	LCD_SCLK	PD2	PB7	PB1	SPI Clock
SDIN	LCD_SDIN	PD1	PB5	PB2	SPI MOSI

Tabulka 2.9. Přiřazení pinů pro PCD8544.

Protože displej má pouze 84 bodů ve směru X, max. 14 znaků na řádek pro znakovou sadu 6x8. Takže mohou být až dva znaky “neviditelné“. Pokud to vadí, můžeš zkrátit texty ve variables.h.

2.6.9. PCF8814 je obvykle řízen 3-vodičovým SPI.

Modul	config-<MCU>.h	328 82,6%	644 41,3%	1280 10,6%	odkaz
/RES	LCD_RES	PD4	PB2	PB4	
/CS	LCD_CS	PD5	PB4	PB5	možnost
SCLK	LCD_SCLK	PD2	PB7	PB1	SPI Clock
SDIN	LCD_SDIN	PD1	PB5	PB2	SPI MOSI

Tabulka 2.10. Přiřazení pinů pro PCD8814.

V případě potřeby lze výstup upravit pomocí nastavení Y-Flip a ten MX-Pin (X-Flip) od PCF8814 pomocí (Pull-Up/Down) otočit.

2.6.10. SH1106 je řízen 3-řádkovým SPI, 4-řádkovým SPI nebo I2C. 3-řádkové SPI vyžaduje Bit-Bang režim a SPI_9 musí být povolen.

Modul	config-<MCU>.h	328 83,0%	644 41,6%	1280 10,7%	odkaz
/RES	LCD_RESET	PD4	PB2	PB4	možnost
/CS	LCD_CS	PD5	PB4	PB5	možnost
A0	LCD_A0	PD3	PB3	PB7	
SCL (D0)	LCD_SCL	PD2	PB7	PB1	SPI Clock
SI (D1)	LCD_SI	PD1	PB5	PB2	SPI MOSI

Tabulka 2.11. Přiřazení pinů pro SH1106 4-řádkové-SPI.

Modul	config-<MCU>.h	328 83,2%	644 41,6%	1280 10,7%	odkaz
/RES	LCD_RESET	PD4	PB2	PB4	možnost
/CS	LCD_CS	PD5	PB4	PB5	možnost
A0		Gnd			
SCL (D0)	LCD_SCL	PD2	PB7	PB1	SPI Clock
SI (D1)	LCD_SI	PD1	PB5	PB2	SPI MOSI

Tabulka 2.12. Přiřazení pinů pro SH1106 3-řádkové-SPI (jen Bit-Bang).

Modul	config-<MCU>.h	328 84,0%	644 41,9%	1280 10,7%	odkaz
/RES	LCD_RESET	PD4	PB2	PB0	možnost
/CS		Gnd			
SCL (D0)	I2C_SCL	PD1	PC0	PD0	
SDA (D1)	I2C_SDA	PD0	PC1	PD1	
SA0	Gnd (0x3c)				3.3V (0x3d)

Tabulka 2.13. Přiřazení pinů pro SH1106 v I2C.

Pomocí X/Y-Flip nastavení můžeš změnit orientaci displeje.

U mnoha zobrazovacích modulů SH1106 musí být X-posunutí nastaveno na 2.

2.6.11. SSD1306 je řízen 3-řádkovým SPI, 4-řádkovým SPI nebo I2C.

3-řádkové SPI vyžaduje Bit-Bang režim a SPI_9 musí být povolen.

Modul	config-<MCU>.h	328 83,1%	644 41,6%	1280 10,7%	odkaz
/RES	LCD_RES	PD4	PB2	PB4	možnost
/CS	LCD_CS	PD5	PB4	PB5	možnost
DC	LCD_DC	PD3	PB3	PB7	
SCLK (D0)	LCD_SCLK	PD2	PB7	PB1	SPI Clock
SDIN (D1)	LCD_SDIN	PD1	PB5	PB2	SPI MOSI

Tabulka 2.14. Přirazení pinů pro SSD1306 4-řádkové-SPI.

Modul	config-<MCU>.h	328 83,2%	644 41,6%	1280 10,7%	odkaz
/RES	LCD_RES	PD4	PB2	PB4	možnost
/CS	LCD_CS	PD5	PB4	PB5	možnost
SCLK (D0)	LCD_SCLK	PD2	PB7	PB1	SPI Clock
SDIN (D1)	LCD_SDIN	PD1	PB5	PB2	SPI MOSI

Tabulka 2.15. Přirazení pinů pro SSD1306 3-řádkové-SPI (jen Bit-Bang).

Modul	config-<MCU>.h	328 84,1%	644 42,0%	1280 10,7%	odkaz
/RES	LCD_RES	PD4	PB2	PB0	možnost
/SCL (D0)	I2C_SCL	PD1	PC0	PD0	možnost
SDA (D1&2)	I2C_SDA	PD1	PC1	PB0	
SA0 (D/C)	Gnd (0x3c)				3.3V (0x3d)

Tabulka 2.16. Přirazení pinů pro SSD1306 I2C.

Pomocí X/Y-Flip nastavení můžeš změnit orientaci displeje.

2.6.12. ST7036 (nevyzkoušen) je adresován přes 4-bitové-Paralelní nebo 4-bit SPI.

Modul	config-<MCU>.h	328 76,2%	644 38,1%	1280 9,8%	odkaz
DB4	LCD_DB4	PD0	PB4	PB0	
DB5	LCD_DB5	PD1	PB5	PB1	
DB6	LCD_DB6	PD2	PB6	PB2	
DB7	LCD_DB7	PD3	PB7	PB3	
RS	LCD_RS	PD4	PB2	PB4	
R/W	LCD_RW	Gnd	???	???	možnost
E	LCD_EN	PD5	PB3	PB5	
XRESET		Vcc	???	???	možnost

Tabulka 2.17. Přirazení pinů pro 4-bitový-paralelní ST7036.

Modul	config-<MCU>.h	328 76,1%	644 38,1%	1280 9,8%	odkaz
XRESET	LCD_RESET	PD4	PB2	PB4	možnost
CSB	LCD_CS	PD5	PB4	PB5	možnost
RS	LCD_RS	PD3	PB3	PB7	
SCL (DB6)	LCD_SCL	PD2	PB7	PB1	SPI Clock
SI (DB7)	LCD_SI	PD1	PB5	PB2	SPI MOSI

Tabulka 2.18. Přirazení pinů pro 4-drátový-SPI od ST7036.

ST7036i hovoří I2C, ale (zatím) není podporován. Zvláštní Funkce ST7036 je pin pro aktivaci rozšířené sady instrukcí (Pin EXT), který je zapnut pro většinu modulů. Měl by být vypnut, musíš nastavení LCD_EXTENDED_CMD a LCD_CONTRAST vykomentovat.

2.6.13. ST7565R je řízen 4/5-řádkovým-SPI.

Modul	config-<MCU>.h	328 83,2%	644 41,7%	1280 10,7%	odkaz
/RES	LCD_RESET	PD0	PB2	PB4	možnost
/CS1	LCD_CS	PD5	PB4	PB5	možnost
A0	LCD_A0	PD1	PB3	PB7	
SCL (DB6)	LCD_SCL	PD2	PB7	PB1	SPI Clock
SI (DB7)	LCD_SI	PD3	PB5	PB2	SPI MOSI

Tabulka 2.19. Přiřazení pinů pro 4/5-řádkový-SPI k ST7565R.

Pro správné zobrazení musíš možná použít nastavení X/Y-Flip a experimentovat s X-Offset.

2.6.14. ST7735 ovládaný 4-drátovým-SPI.

Modul	config-<MCU>.h	328 95,7%	644 47,9%	1280 12,2%	odkaz
/RESX	LCD_RES	PD4	PB2	PB4	možnost
/CSK	LCD_CS	PD5	PB4	PB5	možnost
D/CX	LCD_DC	PD3	PB3	PB7	
SCL	LCD_SCL	PD2	PB7	PB1	SPI Clock
SDIO	LCD_SDA	PD1	PB5	PB2	SPI MOSI

Tabulka 2.20. Přiřazení pinů od ST7735.

Pro správné zobrazení budeš možná muset s X/Y-Flip nastavením experimentovat. Když je aktivována LCD_LATE_ON, startuje tester s vymazaným displejem, což má za následek krátké zpoždění při zapnutí. V opačném případě je po zapnutí krátce vidět náhodné pixely.

2.6.15. ST7920 lze ovládat ve 4-bitovém-paralelním-Modu nebo s SPI.

Modul	config-<MCU>.h	328 83,5%	644 xx,x%	1280 xx,x%	odkaz
/XRESET	LCD_RES	PD4	PB2	PB4	možnost
/CS	LCD_CS	PD5	PB4	PB5	možnost
SCLK	LCD_SCLK	PD2	PB7	PB1	SPI Clock
SDIN	LCD_SDIN	PD1	PB5	PB2	SPI MOSI

Tabulka 2.21. Přiřazení pinů pro SPI s ST7920.

Modul	config-<MCU>.h	328 83,8%	644 41,8%	1280 10,7%	odkaz
/XRESET	LCD_RESET	Vcc	Vcc	Vcc	
E	LCD_EN	PD5	PB3	PB5	
RS	LCD_RS	PD4	PB2	PB4	
RW	LCD_RW	Gnd	Gnd	Gnd	
D4	LCD_DB4	PD0	PB4	PB0	
D5	LCD_DB5	PD1	PB5	PB1	
D6	LCD_DB6	PD2	PB6	PB2	
D7	LCD_DB67	PD3	PB7	PB3	

Tabulka 2.22. Přiřazení pinů pro 4-bitové-paralelní rozhraní ST7920.

Z důvodu špatného designu modelu ST7920 lze používat pouze písma se šířkou 8 pixelů. K zpracování vodorovného adresování v 16 ti bitových krocích jsem musel nastavit vyrovnávací paměť obrazovky pro znaky.

2.6.16. STE2007/HX1230 Typické 3-drátové-SPI ovládání pro STE2007.

Modul	config-<MCU>.h	328 82,5%	644 41,3%	1280 10,6%	odkaz
/RES	LCD_RES	PD4	PB2	PB4	možnost
/CS	LCD_CS	PD5	PB4	PB5	možnost
SCLK	LCD_SCLK	PD2	PB7	PB1	
SDIN	LCD_SDIN	PD1	PB5	PB2	

Tabulka 2.23. Přiřazení pinů pro SPI s ST7920.

V případě potřeby můžeš výstup otočit pomocí X/Y-Flip nastavení.

2.6.17. VT100 Terminal Ovladač VT100 nahrazuje LCD modul a kompletní výstup je realizován přes sériový terminál VT100. Konfigurační sekce v VT100 aktivuje automaticky také sériové rozhraní. Vezmi prosím na vědomí, že VT100 ovladač ostatní možnosti, které by mohly ovlivnit výstup, deaktivuje.

2.7. Tlačítka a ovládání

Tester je primárně ovládán testovacím tlačítkem, ale umožňuje další možnosti ovládání, které usnadňují provoz nebo které jsou pro některé funkce nutné.

2.7.1. Testovací tlačítko **zapíná** tester a slouží k **obsluze**.

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 komponentů pokračuje.

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

2.7.2. Rotační kodér (Hardwarová možnost) S rotačním kodérem operace dostává další funkce, závislé na kontextu.

Podrobnosti jsou vysvětleny v následujících částech.

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

Čtecí algoritmus bere v úvahu počet pulzů Gray kódu na krok (ENCODER_PULSES) a také počet kroků k plné 360° otáče.

S hodnotou (ENCODER_steps) můžeš provést jemné nastavení detekce rychlosti otáčení. Obě hodnoty najdeš na straně 45. Vyšší hodnota zpomaluje rychlost otáčení, nízká hodnota ji zvyšuje. Pokud je směr otáčení obráceně, změň definici pinů pro A a B v config<MCU>.h.

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.

2.7.3. (Vice/méně tlačítek). Pokud dáváš přednost klávesám před rotačním kodérem, můžeš alternativně také použít dvě tlačítka viz na straně 45. Tlačítka jsou stejně připojené jako otočný kodér (pull-up odpory, logická úroveň nízká při aktivaci).

Pro zrychlení rychlosti, podobné jako při otáčení rotačního kodéru, jednoduše stiskni dlouze tlačítko. Čím déle, tím je vyšší zrychlení.

2.7.4. Dotyková obrazovka (Hardwarová možnost) Radiče ST7565 nebo SSD1306 se obvykle připojují s 4-vodičovým SPI rozhraním. Další možnost vstupu je dotyková obrazovka. K tomuto účelu by měl být LCD modul dostatečně velký a podporovat alespoň 8 řádků textu s 16 znaky nebo více. Abychom ušetřili cenný prostor na LCD, nepoužíváme k dotyku ikony. Místo toho jsou vlevo a vpravo neviditelné pruhy (každý 3 znaky široký), nahoře a dole (2 řádky vysoké) a jedna oblast uprostřed.

Pruhy nalevo a nahoře jsou pro „méně“ nebo „výběr nahoru“, pruhy vpravo a dole pro „více“ nebo „výběr dolů“. Takže stejné funkce jako u otočného kodéru.

Dlouhý dotyk urychluje změny některých funkcí, podobně rychlosti otáčení rotačního kodéru.

Oblast uprostřed je softwarová verze testovacího tlačítka, **nelze** jí ale použít např. v testu Zenerových diod pro napájení externím proudem.

Pro použití dotykové obrazovky je nutná kalibrace.

Pokud ještě nejsou v paměti EEPROM uloženy hodnoty nastavení, spustí se automaticky po zapnutí testeru. Kromě toho ji můžeš také spustit z hlavní nabídky. Postup je celkem jednoduchý.

Pokud vidíš (na barevných LCD žlutou *) hvězdičku, stiskni ji. Tester pak odstraní hvězdičku a zobrazí nativní x/y polohu po každém dotyku. První nastavovací bod je vpravo nahoře, druhý vlevo dole. V závislosti na výsledku tester opakuje srovnání až třikrát. Pokud máš problémy s kalibrací, například podivné x/y pozice, zkontroluj orientaci dotykové obrazovky.

Levá horní část je považována za nulový bod.
Kalibraci můžeš kdykoliv přerušit testovacím tlačítkem.
Řadič má spínače pro otočení a zaměnění orientace:

* Vpravo nahoře je první nulový bod		
x	y	upravit hodnotu nastavení s:
nízká	nízká	TOUCH_FLIP_X
nízká	vysoká	TOUCH_FLIP_X & TOUCH_FLIP_Y
vysoká	nízká	v pořádku
vysoká	vysoká	TOUCH_FLIP_Y
Po úspěšné úpravě kalibrace nezapomeň, uložit hodnoty offsetu... (Hlavní nabídka: Uložit).		

Tabulka 2.24. Kalibrace x/y-hodnot.

Podporované ovladače dotykové obrazovky: - ADS7843/XPT2046.
Konfiguraci najdeš pod oddílem pro LCD moduly v config<MCU>.h
(aktuálně pouze v config_644.h, protože 328 nemá dostatečně množství IO-pinů).

2.7.5. Uživatelské rozhraní Existuje několik možností, jak zobrazit výstup pro displeje, aby byl více efektní nebo více srozumitelnější.

U displejů s více než dvěma řádky textu:

- Zobrazení nápovědy ke klávesám místo kurzorů, pokud jsou k dispozici (UI_KEY_HINTS).

Pro grafické displeje:

- všechny předchozí možnosti
- Symboly s vývody pro třívodičové polovodiče (SW_SYMBOLS, ne pro displeje s nízkým rozlišením)
- žádný výstup textového vývodu pro třívodičové polovodiče (UI_NO_TEXTPINOUT, pokud je aktivována funkce SW_SYMBOLS)
- žádný výstup z textového vývodu tělesa diody MOSFET (UI_NO_BODYDIODE_TEXTPINOUT)
- zobrazení testovacích vývodů v invertovaném zobrazení (UI_PROBE_REVERSED)
- Zobrazení symbolu otazníku v případě neúspěšného vyhledávání (UI_QUESTION_MARK)
- Zobrazení symbolu zenerovy diody pro všechny zenerovy testy (UI_ZENER_DIODE)
- Zobrazení symbolu oscilujícího křemene pro režimy LF/HF v rozšířeném režimu čítače frekvence (UI_QUARTZ_CRYSTAL)

Pro barevné grafické displeje:

- všechny předchozí možnosti
- Barevné kódování testovacích pinů (UI_PROBE_COLORS)
- Nadpisy s vlastní barvou (UI_COLORED_TITLES)
- Kurzory a nápovědy kláves s vlastní barvou (UI_COLORED_CURSOR)
- Barevné označení rezistorů (SW_R_E24_5_CC, SW_R_E24_1_CC a SW_R_E96_CC)




2.7.6. Bzučák Pokud si přeješ také akustický signál, můžeš použít aktivní bzučák (s integrovaným oscilátorem). Bzučák se ovládá pomocí (viz BUZZER_CTRL v config_<MCU>.h) a může být použit přímo (< 20mA) nebo prostřednictvím tranzistoru (NPN nebo n-kanálový MOSFET).

V současné době se bzučák používá pouze při testu spojitosti. Další aplikace budou následovat v budoucnu.

2.7.7. Komunikace s PC Tester používá ke komunikaci s PC sériové TTL rozhraní. Toto může být jednosměrné připojení (pouze odeslat) k sériovému připojení vydání nalezených součástí, nebo dokonce obousměrné k automatizaci. V obou případech musí být aktivováno sériové rozhraní. (Viz „Busses“ v config_MCU.h strana 56).

Zvláštní znaky jsou nahrazeny standardními znaky, např. je z Ω (Ohm) R.

Konverzní tabulka:

	> <
	
	□
Ω	R
μ	u

Poznámky:

- 9600 8N1

- nový řádek je <CR><LF>

2.7.8. Seriový výstup Tester také poskytuje detekované komponenty prostřednictvím sériového rozhraní pokud je povoleno (viz UI_SERIAL_COPY v sekci „různá nastavení“ na straně 51 v config.h).

K tomu postačuje jednoduchý terminálový program na PC.

Výstup sleduje výstup na LCD displeji, ale pouze pro nalezené komponenty.

Neexistuje žádný výstup z nabídek a funkcí přes sériové rozhraní, s výjimkou výsledků testu optočlenů.

2.7.9. Automatisierung Automatizace umožňuje dálkové ovládání testeru pomocí příkazů přes obousměrné sériové připojení.

Povolíš-li tuto funkci, viz UI_SERIAL_COMMANDS v sekci „různá nastavení“, stránka 51 v config.h. tak se mírně mění chování testeru.

Automatizace vynucuje automatický režim, a tester po zapnutí **nehledá** komponentu. Příkazové rozhraní je velmi jednoduché. Pošleš příkaz a tester odpoví.

Komunikace je založena na textových řádcích ASCII a příkazy rozlišují velká a malá písmena.

Každý příkazový řádek je zakončen (novým řádkem) <CR> <LF> nebo <LF>.

Tester přijímá příkazy pouze během čekání po zapnutí uživatelem, po výstupu komponenty nebo provedení funkce menu.

Řádky odpovědi končí <CR> <LF> (novým řádkem).

Seznam příkazů a jejich popis je v oddílu „Příkazy pro dálkové ovládání“ od stránky 39.

2.7.10. Výstup přes VT100 Namísto LCD displeje můžeš dosáhnout kompletní výstup přes VT100 terminál (viz 2.6 v sekci LCD moduly na stránce 19). Aby nedocházelo k nejasnostem v rozvržení výstupu, jsou další možnosti sériového rozhraní deaktivované.

3.0.1. 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.c. V tomto případě to nemůže udělat tester sám.

3.0.2. 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 (strana 48) a CYCLE_MAX (strana 51) v config.h.

Režim automatického přidržování POWER_OFF_TIMEOUT (strana 52) má volitelné automatické vypnutí, které je aktivní pouze při vyhledávání komponent a výstupu.

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

3.0.3. Monitorování baterie je možné nastavit podle tvých představ na stránce 53. 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.

3.0.4. 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.

3.1. Výběrové menu

docílíš krátkým dvojitém 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 51), 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.

3.1.1. 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

3.1.2. 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 kodérem 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 %.

3.1.3. 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 %).

Dvojím stisknutím tlačítka PWM generátor ukončíš.

3.1.4. 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 dvojité stisknutí ukončí generátor.

Rozložení testovacích pinů:

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

Pin # 1 a # 3: uzemněny

Poznámka: Nutný Rotační enkodér nebo jiná volba ovládání!

3.1.5. 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.

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, spustit zenerův test automaticky během běžného vyhledávání součástek (HW_PROBE_ZENER). Pokud není na normálních testovacích vývodech nalezena žádná součástka, kontroluje tester napětí na zenerových testovacích vývodech.

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

3.1.6. 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.

3.1.7. tester continuity 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 R1 (680 Ω). Prahové hodnoty pro bzučák jsou:

< 100	mV	nepřetržitý zvuk
100-700	mV	opakované krátké pípnutí
> 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 zkoušených vývodů:

Pin #1:	Vcc (680Ω odpor pro omezení proudu)
Pin #3:	Zem

3.1.8. 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 přesvědčte, 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: -

3.1.9. Unikající proud kondenzátoru Test svodového proudu nabíjí kondenzátor a zobrazuje proud a napětí na měřicím odporu. Načítání začíná s R_L (680Ω) a přepne na R_H ($470k\Omega$), 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 vybijí 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: -

3.1.10. 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 přeskocit.

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.

3.1.11. L/C-Měřič (hardwarová úprava) je založen na jednoduchém obvodu LC oscilátoru, jaký se používá u více levných PIC L/C měřičů. Obvyklý design ($82\mu H$ a $1nF$) 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 vidíte „-“ bez připojené součástky, proveďte 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 měřičem LCR a podle toho změň LC_METER_C_REF. - Pokud tě zajímá frekvence LC oscilátoru a jeho drift, pak aktivuj LC_METER_SHOW_FREQ.

3.1.12. 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 dělič.

Oba okruhy jsou popsány v dokumentaci Karla-Heinze [16].

3.1.13. Jednoduchý čítač pokud je nainstalován přídavný 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.

3.1.14. 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.

3.1.15. 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 (nejasné)
>= 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 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 680Ω odporem pro omezení proudu)
Pin #3:	Zem
T0:	Vstup čítače

3.1.16. 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 událostí 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 680Ω odpor k omezení proudu)
- Pin #3: uzemění

3.1.17. 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.

3.1.18. 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.

3.1.19. 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.

Druhou variantou je pevný modul, který je připojen ke konkrétnímu MCU pinu.

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.

Mění 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 μ PD1986C <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.

- IR přijímací modul na testovacích pinech

Nejprve připojte IR přijímací modul k IR detektoru dálkového ovládání!

Administrace pro modul TSOP:

Ukázka # 1: uzemnění / Gnd

Sonda # 2: Vs (680 Ω omezovač proudu)

Sonda # 3: Data/Out

Poznámka: 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.

- Pevný IR přijímací modul

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

3.1.20. 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>

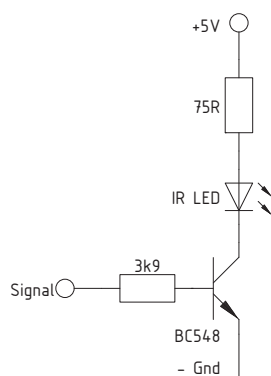
Rozložení testovacích pinů:

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



Obrázek 3.1. Příklad na 50mA IR ovladač s (V_f 1.5V, I_f 100mA,)

Poznámka: Pokud se načasování pulsů/pauzy nehodí, použij alternativu metody čekající smyčky SW_IR_TX_ALTDELAY na straně 48.

To je nutné, pokud tvůj C kompilátor, přesto že je nastaven na zachování kódu vloženého v assembleru, optimalizujete.

3.1.21. 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.

3.1.22. 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 kódem 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 3.1. Rozložení pinů pro typické 3kolíkové servopohony

3.1.23. OneWire skenování zobrazuje ROM-kódy, všech připojených uživatelů.

Informace o nastavení sběrnice OneWire naleznete v části „Sběrnice a rozhraní“ na straně 56.

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 12.

Funkci ukončí, jako obvykle, dvě krátké stisknutí.

3.1.24. Snímač teploty DS18B20 v tomto výběru lze použít tento teplotní OneWire senzor ke čtení teplot.

K nastavení OneWire sběrnice na stránce 56, 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, 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))

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

3.1.25. DHTxx senzory 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 být poškodit MCU.

3.1.26. Převodník termočlánků MAX6675/MAX31855

Oba MAX jsou termočlánkové převodníky s kompenzací studeného spoje a jsou řízeny prostřednictvím sběrnice SPI. MAX6675 je určen pouze pro termočlánky 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.

3.1.27. 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:

- interní referenční napětí T1 (v mV)
- T2 Srovnání odporů R1 (offset v mV)
- T3 Srovnání R_H odporů (offset v mV)
- T4 Odstraňte zkrat zkušebních pinů/kabelu
- Test těsnosti T5 pro zkušební piny s úrovní Gnd (napětí v mV)
- Test těsnosti T6 pro zkušební piny s úrovní Vcc (napětí v mV)

3.1.28. 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 54.

Samo-ladění je do značné míry autotestu provozem a obsluhou podobné. Kroky seřízení jsou:

- kompenzace A1 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
- vnitřní odpor A4 portových pinů pro Gnd (napětí přes RiL)
- A5 vnitřní odpor pinů portu pro Vcc (napětí přes RiH)
- kapacita testovacích pinů/kabelů A6 (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 |

Poznámka: Pokud se hodnoty odporu zkušebních kolíků příliš liší, je možný kontaktní problém.

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.

3.1.29. Ušchovat/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.

3.1.30. Ukázat hodnoty Tato funkce zobrazuje aktuální hodnoty nastavení.

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

3.1.31. 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.

3.1.32. Vypnout Za předpokladu, že jsi tuto funkci přes SW_POWER_OFF strana 52 aktivoval, můžeš zde tester vypnout.

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

4.1. Odpor

se měří dvakrát. tj. v obou směrech a hodnoty pak porovnány. Pokud jsou hodnoty příliš odlišné, předpokládá tester, že se jedná o dva odpory a ne o jeden.

V tomto případě se zobrazí dva odpory na stejných pinech ve tvaru „1 - 2 - 1“ a dvě hodnoty.

Pro odpory menší než 10Ω se provádí další měření s vyšším rozlišením.

Ve výjimečných případech nemůže tester velmi malé odpory detekovat. Nejlépe pak jednoduše opakovat měření.

Pokud je aktivována volitelná kontrola standardních hodnot E (SW_R_E *) na stránce 50, porovná tester další menší a následující větší standardní hodnotu s změřeným odporem při zohlednění tolerance komponenty.

Existují dva výstupní režimy. V textovém režimu dává tester řadu E plus tolerance a vhodné standardní hodnoty. „-“ znamená, že žádná hodnota normy nepasuje. V režimu barevného kódu zobrazuje tester série E a barevný kód standardní hodnoty včetně inkoustové pásky pro toleranci. Barvy se mohou, podle použitého displeje lišit Pokud některá barva nesouhlasí, změň jednoduše hodnotu barvy (COLOR_CODE_*) v souboru color.h. Internetové vyhledávání „nástroje RGB565“ uvádí mnoho online nástrojů pro výběr hodnot barev RGB565.

4.2. Kondenzátory

se měří třemi metodami. Velké kondenzátory $> 47\mu F$ používají metodu cyklického nabíjení s 10ms dlouhými pulzy. Středně velké mezi $4,7\mu F$ a $47\mu F$ používají stejnou metodu ale nabíjecí pulsy jsou 1ms dlouhé.

Malé kondenzátory běží přes analogový komparátor. Tímto způsobem je optimalizována přesnost měření kapacit.

Změřené hodnoty velkých kondenzátorů vyžadují korekci. Bez korekce by byly hodnoty příliš vysoké. Domnívám se, že to způsobuje měřicí metoda, protože analogově-digitální převod po nabití impulzem potřebuje jistý čas ve kterém kondenzátor část nabití, kvůli svodovým proudům, ztratí. Ztráty jsou samozřejmě také, samotným převodem z analogového na digitální. Nabíjení kondenzátoru tedy trvá déle a kapacita se zdá být vyšší. Pozdější měření samovybíjení se to snaží vyrovnat což se jen částečně povede. Korekční faktory CAP_FACTOR_SMALL, CAP_FACTOR_MID a CAP_FACTOR_LARGE jsou v config.h na stránce 54 tak zvolené, aby pasovaly pro většinu testerů. V některých případech však může být změna nutná. Přidána byla logika zabráňující detekci velkých kondenzátorů jako odporů. Odporů menší než 10Ω budou dodatečně zkontrolovány na kapacitu. U kondenzátorů vyšších než 18nF se snaží tester měřit i ESR. Alternativně můžeš ale aktivovat starou metodu měření, která měří až od 180nF.

Protože měření nepoužívá AC s určitou frekvencí, nečekej super přesné výsledky.

Použitá metoda může možná odpovídat měření při 1 kHz.

Pro zkoušku elektrolytických kondenzátorů je měření více než dostatečné.

U filmových kondenzátorů s malými hodnotami mohou nastat různé hodnoty v závislosti od frekvence taktu MCU.

Myslím, že by-to pan Fourier uměl vysvětlit.

Naměřená hodnota kapacity větší než 5 pF (včetně nulového posunu) se považuje za platnou.

Nižší hodnoty jsou příliš nepřesné a mohou být způsobeny třeba jiným umístěním měřících kabelů.

Dalším měřením je svodová proudní ztráta pro kondenzátory vyšší než $4,7\mu F$.

Hodnota vydá poznámku k označení stavu Elko.

Z mých testů jsou typické hodnoty pro elkos:

- 10-220 μF 1-3 μA
- 330-470 μF 4-5 μA
- 470-820 μF 4-7 μA
- > 1000 μF 5-7 μA na 1000 μF

Volitelný test pro standardní hodnoty E je také k dispozici pro kondenzátory (SW_C_E_*) stránka 50, ale pouze v textovém režimu, protože existuje příliš mnoho různých barevných kódů pro kondenzátory.

4.3. tlumivky

Měření indukčnosti není příliš přesné a věci jako např. takt MCU a deska ovlivňují výsledek. Samotné měření je založeno na stanovení času mezi zapnutím elektřiny a dosažením určitého proudu. U velkých tlumivek se provádí měření nízkého proudu, a pro malé tlumivky vysoko proudové měření, které dočasně překračuje proudový limit MCU pinů po dobu až asi 25 mikrosekund. Při vyšetřování efektů MCU taktu jsem našel vzory v odchylkách, které lze použít pro kompenzaci. V závislosti na testeru může být nutné provést úpravu. V souboru inductance.c ve funkci MeasureInductor() je proměnná „Offset“ kompenzace.

To je offset pro referenční napětí. Kladná hodnota snižuje indukčnost, a záporná ji zvyšuje. Kompenzace pro měření vysokého proudu je založena na taktu MCU a je rozdělena do tří časových pásem. Pro měření nízkého proudu je aktuálně pouze jednoduchá kompenzace, protože zde by-byla nutné další testy.

Pokud vidíš, ve srovnání se „správným“ LCR měřičem, velké rozdíly hodnoty offsetu, můžeš podle něj svůj tester upravit. Pokud chceš zkontrolovat hodnoty E-norm, aktivuj přepínač SW_L_E_* v config.h na stránce 50 (pouze textový režim).

Poznámka:

- V případě neočekávaných naměřených hodnot opakuj měření.
- Měření indukce se provádí, jen když je její odpor menší než $2\text{k}\Omega$.

4.4. Vybití součástek

Tester zkouší připojenou součástku před a během měření vybití. Pokud nelze nastavit na předem určenou nulovou hodnotu (CAP_DISCHARGED), zobrazí chybu označující testovací pin a zbytkové napětí.

Funkce vybíjení není založena na pevném časovém limitu, ale přizpůsobí se automaticky průběhu vybíjení. Tímto způsobem je baterie rychleji detekovaná a velké kondenzátory získávají více času k vybití. Pokud je velký elektrolytický kondenzátor rozpoznán jako baterie, zkus to znovu. V prostředí se spoustou elektrického rušení, může být nulová hodnota CAP_DISCHARGED s 2mV příliš nízká (v případě potřeby uprav na stránce 55). Zobrazené zbytkové napětí pomáhá najít optimální nulovou hodnotu.

4.5. ADC převzorkování

podporuje variabilní převzorkování (1 - 255).

Výchozí hodnota je 25 vzorků.

Můžeš se pokusit zvýšit přesnost testeru zvýšením hodnoty převzorkování.

Avšak s vyšší hodnotou je nutný delší čas, tzn. měření se zpomalí.

4.6. Zobrazení výsledků

Některá jména a zkratky byly změněny. Výstup některých komponent byl rozdělen na více stránek, pokud nemá LCD modul dostatek řádků.

U jediné diody je výsledek Vf měření s nízkým zkušebním proudem ($10\mu\text{A}$) zobrazen v závorkách, pokud je hodnota měření pod 250 mV. To ukazuje na germaniovou diodu.

Většina datových listů germaniových diod udává pro Vf měřicí proud 0,1 mA.

Bohužel tester tento měřicí proud nepodporuje. Při vyšších proudech je Vf kolem 0,7 V, což ztěžuje rozlišení od křemíkových diod.

Tester ukáže svodový proud I_R pro diodu nebo I_{CEO} pro bipolární tranzistor pokud je vyšší než 50nA.

Germaniové tranzistory mají svodový proud v rozmezí několika μA až asi 500 μA . Germaniové diody mají obvykle jen málo μA .

U některých součástí se zobrazí hodnota kapacity. Je-li kapacita pod 5pF nebo měření selhalo, je na výstupu 0pF.

Pokud je nalezen (ochuzovací typ) FET se symetrickým odtokem a zdrojem, ukazuje přiřazení pinů „x“ místo „D“ nebo „S“, protože jsou funkčně totožné a nejdou v testeru rozlišit.

V takovém případě vyhledej podrobnosti o přiřazení pinů v datovém listu.

Triakové přiřazení pinů je označeno:

„G“, „1“ a „2“. „1“ je MT1 a „2“ je MT2.

Pro UJT, pokud je detekce aktivovaná, je:

„1“ pro B1, „2“ pro B2 a „E“ pro emitor.

Pokud byla aktivována funkce „Fancy Pinout“ (SW_SYMBOLS), zobrazí se symbol součástky a jeho položení na zkušebních pinech pro 3pinové polovodiče.

Není-li na displeji dostatek místa, bude výstup ikony přeskočen.

4.7. Dodatečné pokyny

bipolární tranzistory

Malé písmeno za hFE-hodnotou zobrazuje typ zkoušecího prostředí, použitého k měření:

- e: emitorové
- c: kolektorové

Pokud jsi aktivoval výstup z testovacího proudu hFE (SW_HFE_CURRENT), pak tester ukazuje I_C pro obvod emitoru a I_E pro kolektorový obvod.

Při testu na diody je V_f měřen za prvé s R_h (vysokým zkoušecím proudem) a za druhé s R_l (nizkým zkoušecím proudem). Výchozí funkce pro bipolární tranzistory interpoluje s obou V_f hodnot V_{BE} závisle na hFE pro virtuální zkoušecí proud. Tím lze získat praxi blízké hodnoty, protože V_{BE} malého signálového tranzistoru je měřen jiným proudem nežli u výkonového tranzistoru.

U bipolárního tranzistoru s bázevým emitorovým odporem je odpor zobrazen.

Nezapomeň, že má B-E odpor vliv na V_{BE} i na hFE.

Pokud má tranzistor navíc ochrannou diodu, může být rozpoznán jako tranzistor nebo jako dvě diody, v závislosti na hodnotě odporu báze-emitor.

v druhém případě jsou zobrazeny dvě diody plus odpor s uvedením možného tranzistoru.

Bohužel nízký odpor báze-emitor brání jednoznačné detekci jako tranzistor.

Dalším zvláštním případem je bipolární tranzistor s ochrannou diodou umístěnou na stejném substrátu.

Integrovaný PN přechod vytváří parazitní druhý tranzistor. NPN dostane parazitní PNP a naopak.

Pokud je takový tranzistor detekován, je po specifikaci typu označen znakem „+“.

u šotky-tranzistoru (Schottky-clamped BJT) je schotky-dioda mezi bází a kolektorem incl. V_f ukázána, pokud je to aktivováno s (SW_SCHOTTKY_BJT) na straně 49.

Nezapomeň, že je I_{CEO} u šotky-tranzistoru zvýšen.

TRIACKY s jsou provozovány ve třech nebo čtyřech režimech, známých také jako kvadranty.

Obvykle se některé parametry liší v závislosti na kvadrantu, např. spouštěcí proud I_{GT} . V některých případech se stává, že zkušební proud Tester je dostatečný pro spuštění TRIACu v jednom kvadrantu, ale ne v jiném. Protože tester potřebuje dva testovací běhy, aby získal rozložení pinů pro MT1 a MT2, nemůže tester v takových případech připojení rozlišit, tím je možné, že jsou označeny obráceně.

Některé TRIACKY mohou být testerem spuštěny, ale mají vysoký udržující proud (I_H), čímž je nelze správně rozpoznat. Pokud je spouštěcí proud je pro tester příliš vysoký, je označen jako odpor.

CLD Test diod detekuje CLD (Current Limiting Diode) jako normální diodu a vydává její aktuální I_F jako svodový proud.

- Všimni si, že jsou u CLD anoda a katoda obráceny oproti normální diodě.
- Samostatná CLD detekce je obtížná, protože svodový proud germania nebo výkonné Šottkyho diody leží v rozsahu I_F (od přibližně $33\mu A$).
- Pokud má dioda neobvyklý V_f , neobvykle nízký V_f při nízkém proudovém testu (druhá hodnota v závorce) a nelze měřit žádnou kapacitu, pak je to pravděpodobně CLD.

Nepodporované komponenty všechny polovodiče, které vyžadují vysoký řídicí proud, nelze rozpoznat, protože má tester max. asi 7 mA spínací proud. Tester také dodává pouze 5V, což např. nestačí na DIAC s V_{BO} od 20-200 V.

Zjednáání nápravy u některých testerů.

Jestli tvůj tester ukazuje následovný problém, můžeš ho takto zkusit opravit.

-hFE moc vysoký

Problem:

U kolektorového okruhu s R_l jako bázevým odporem je z neznámého důvodu měřeno moc nízké bázevé napětí. Tímto vypadá bázevý proud také nižší a způsobí vysokou hFE hodnotu.

Postižený tester:

Hiland M644

Náprava:

Aktivuj `NO_HFE_C_RL` v `config.h`! Viz na stránce 50.

Znamé problémy Úložiště nebo super-kondenzátor, např. Řada Panasonic NF je detekován jako dioda nebo dvě antiparalelní diody.

Měření kapacity nemůže určit použitelnou hodnotu.

- Při použití spínacího zdroje nebo DC-DC převodníku pro napájení, vydá někdy tester omylem Elko kolem $50\mu F$, ačkoli není žádná součástka připojena.
- ESR může pracovat s kondenzátory s 180 - 220nF v závislosti na MCU taktu variovat.
- Problém s měřením ESR elektrolytických kondenzátorů s nízkým ESR a pevným elektrolýtem.

4.8. užitečné odkazy

[10] Slovenské fórum

<https://svetelektro.com/phpbb/?w3=dmlld3RvcGljLnBocD9mPTE4JnQ9MzAyODU=>
-Všetko zo sveta elektroniky

[11] Německé fórum

<https://www.mikrocontroller.net/topic/248078>
Hlavní řeč je němčina, anglicky je ale také ok.

[12] Anglické fórum

[https://www.eevblog.com/forum/testgear/\\$20-lcr-esr-transistor-checker-project/](https://www.eevblog.com/forum/testgear/$20-lcr-esr-transistor-checker-project/)
Jen anglicky.

[13] Ruské fórum

<https://vrtp.ru/index.php?showtopic=16451>
- Infos zu verschiedenen Clonen, wie Bilder, Schaltpläne und Firmware

[14] Ruské webové stránky (od indman@EEVblog)

<https://disk.yandex.ru/d/yW8xa5NJgUo5z>
- Informace o různých klonech, jako jsou obrázky, schémata a firmware

[15] Manuál plus div. soubory pro WinAVR (od indman@EEVblog)

https://drive.google.com/file/d/1-IJA8uTcsCA_6SYHEuMydjfS2vNgmwdH/edit

[16] Aktuální k-Firmware

<https://github.com/kubi48/TransistorTester-source>

[17] Aktuální dokumentace pro k-Firmware

<https://github.com/kubi48/TransistorTester-documentation>

4.9. Změny firmwaru

Přečti si Kapitulu 9 od stránky 84.

5.1. Příkazy dálkového ovládání

Když tester příkaz přijme, odpoví konkrétními údaji nebo jeden z následujících standardních textů:

5.1.1. ERR - neznámý příkaz

- není podporován v aktuálním kontextu komponenty
- Přetečení vyrovnávací paměti

5.1.2. OK - příkaz byl proveden

(provedení některých příkazů nějakou dobu trvá)

5.1.3. N/A - Informace/hodnota nejsou k dispozici

Odpovědi s daty nikdy nezačínají žádným z nadstandardních textů aby se vyhnulo možné dvojznačnosti

5.2. Basiskommandos

5.2.1. VER - vrací verzi firmwaru

- ověření připojení a určení sady příkazů na základě verze
- příklad odpovědi: „1,33m“

5.2.2. OFF - vypne tester

- Tester před vypnutím odpoví „OK“

příklad odpovědi: „OK“ <tester se vypne>

5.3. Testkommandos

5.3.1. PROBE - hledá část a přeskočí všechny pauzy pro zadání uživatele

- Tester po ukončení vyhledávání odpoví „OK“

vzorová odpověď: <uplyne nějaký čas pro sondování> „OK“

5.3.2. COMP - vrací ID typu součásti

- ID viz COMP_* in common.h

- Příklad odpovědi pro BJT: „30“

5.3.3. MSG - vrací chybovou zprávu

- pouze v případě, že došlo k chybě (COMP: 1)
- Odpověď může záviset na jazyce uživatelského rozhraní

příklad odpověď: „Baterie? 1:1500 mV“

5.3.4. QTY - vrací počet nalezených komponent

příklad odpověď pro BJT: „1“

5.3.5. NEXT - vyber druhou komponentu

- pouze pokud byly nalezeny dvě složky (POČET:2)

příklad odpověď: „OK“

5.3.6. TYPE -dává zpátky součástky

- jen pro BJT, FET a IGBT

- možné typy:

- | | |
|----------|---|
| - NPN | NPN (BJT) |
| - PNP | PNP (BJT) |
| - JFET | JFET (FET) |
| - MOSFET | MOSFET (FET) |
| - N-ch | n-Kanal (FET, IGBT) |
| - P-ch | p-Kanal (FET, IGBT)- enh. typ obohacení (FET, IGBT) |
| - dep. | typ vyčerpání (FET, IGBT) |

Příklad odpovědi pro BJT: „NPN“

- Ukázková odpověď pro FET (MOSFET): „MOSFET n-ch boost.“

5.3.7. HINT - vrací poznámky o zvláštních funkcích

- pouze pro diody, BJT, FET a IGBT

- možné poznámky:

- NPN možná NPN BJT (dioda)
- PNP možná PNP BJT (dioda)
- R_BE báze-emitor-odpor (dioda, BJT)
- BJT+ integrovaná volnoběžná dioda na stejném substrátu
generuje parazitický další BJT (BJT)
- D_FB Body/volnoběžná dioda (BJT, FET, IGBT)
- D_CLAMP Šotky-tranzistor (BJT)
potřebuje aktivované Šotky-tranzistor rozpoznání
- SYM symetrický odtok a zdroj (FET)

Příklad odpovědi pro BJT: „D_FB R_BE“

- Ukázková odpověď: pro FET (MOSFET): „D_FB“

5.3.8. MHINT - vrací poznámky o měření

- pouze pro BJT

- možné poznámky:

- h_FE_e měření h_FE_e v emitrovém prostředí
- h_FE_c měření h_FE_c v kolektorovém prostředí

Příklad odpovědi pro BJT: „h_FE_e“

5.3.9. PIN - vrací přiřazení pinů

-použité identifikátory:

- | | | | |
|---------------|------------------------------------|---------------|-----------------|
| - odpor | x = verbunden, | - = nespojený | |
| - kondenzátor | x = spojený, | - = nespojený | |
| - dioda | A = anoda, C = katoda, | - = nespojený | |
| - BJT | B = báze, C = kolektor, E = emitor | | |
| - FET | G = gate, S = zdroj, D = odtok, | | x = odtok/zdroj |
| - IGBT | G = gate, C = kolektor, E = emitor | | |
| - SCR | G = gate, A = anoda, C = katoda | | |
| - TRIAC | G = gate, 2 = MT2, 1 = MT1 | | |
| - PUT | G = gate, A = anoda, C = katoda | | |
| - UJT | E = emitor, 2 = B2, 1 = B1 | | |

- Format der Antwort:

<Test-Pin #1 Kennung><Test-Pin #2 Kennung><Test-Pin #3 Kennung>

- Ukázková odpověď pro odpor: „xx-“

- Ukázková odpověď pro diodu: „C-A“

- Ukázková odpověď pro BJT: „EBC“

5.3.10. R - vrací hodnotu odporu

- jen pro odpor (obsahuje také indukivitu)

- ukázková odpověď: „122R“

5.3.11. C - vrací hodnotu kapacity

- jen pro kondenzátor

- ukázková odpověď: „98nF“ „462uF“

5.3.12. L - vrací hodnotu indukivity

- jen pro odpor (obsahuje také indukivitu)

- ukázková odpověď: „115uH“

5.3.13. ESR - vrací hodnotu ESR-Wert zurück (Equivalent Series Resistance)

-vyžaduje povolení měření ESR

- jen pro kondenzátor

- ukázková odpověď: „0.21R“

- 5.3.14. I_l** - vrací I_{leak} zpět (Svodový proud odpovídá samovybíjení)
 - jen pro kondenzátor
 - ukázková odpověď: „3.25uA“
- 5.3.15. V_F** - vrací V_F zpět (dopředné napětí)
 - jen pro diody a PUT
 - také pro tělo
 - dioda od MOSFET a svodová dioda od BJT nebo IGBT
 - ukázková odpověď: „654mV“
- 5.3.16. V_{F2}** - vrací V_F měření nízkého proudu (forward voltage)
 - jen pro diody
 - ukázková odpověď: „387mV“
- 5.3.17. C_D** - vrací C_D (kapacitu diody)
 - jen pro diody
 - ukázková odpověď: „8pF“
- 5.3.18. I_R** - vrací I_R zurück (reverse current)
 - nur pro diody
 - ukázková odpověď: „4.89uA“
- 5.3.19. R_{BE}** - vrací R_{BE} zurück (Basis-emitor-odpor)
 - jen pro diody a BJT
 - ukázkové odpovědi: „38.2R“ „5171R“
- 5.3.20. h_{FE}** - vrací h_{FE} zurück (DC-Stromverstärkungsfaktor)
 - jen pro BJT
 - ukázková odpověď: „234“
- 5.3.21. h_{FE_r}** - vrací obrácený h_{FE} (kolektor a emitor překroucený)
 - jen pro BJT
 - ukázková odpověď: „23“ - vrací h_{FE} (DC-Aktuální činitel zesílení)
 - jen pro BJT
 - ukázková odpověď: „234“
- 5.3.22. I_C** - vrátí testovací proud I_C měření hFE
 - vyžaduje aktivovaný výstup zkušebního proudu měření hFE
 - pro měření hFE s obvodem emitoru
 - pouze pro BJT
 - ukázková odpověď: "3245uA"
- 5.3.23. I_E** - vrací testovací proud I_E měření hFE
 - vyžaduje aktivovaný výstup zkušebního proudu měření hFE
 - pro měření hFE s kolektorovým obvodem
 - pouze pro BJT
 - ukázková odpověď: "3245uA"
- 5.3.24. V_{BE}** - vrací V_{BE} (Basis-emitor-napětí)
 - jen pro BJT
 - ukázková odpověď: „657mV“
- 5.3.25. I_{CEO}** - vrací I_{CEO} (kolektor-emitor-proud, otevřená báse)
 - jen pro BJT
 - ukázková odpověď: „460.0uA“
- 5.3.26. V_{th}** - vrací V_{th} (threshold voltage) prahové napětí
 - jen pro FET (MOSFET) und IGBT
 - ukázková odpověď: „2959mV“
- 5.3.27. C_{GS}** - vrací C_{GS} (gate-zdroj-kapacita)
 - jen pro FET (MOSFET)
 - ukázková odpověď: „3200pF“
- 5.3.28. R_{DS}** - vrací R_{DS_on} (odtok-zdroj-odpor zapnutý)
 - jen pro FET (MOSFET)
 - ukázková odpověď: „1.20R“

- 5.3.29. V_GS_off** - vrací V_GS_off (Cutoffovo napětí)
- jen pro zchudlý FET
- ukázková odpověď: „-3072mV“
- 5.3.30. I_DSS** - vrací I_DSS (odtok-zdroj-Strom, zkracovaný gate)
- jen pro zchudlý-FET
- ukázková odpověď: „6430μA“
- 5.3.31. C_GE** - vrací C_GE (gate-emitor-kapacita)
- jen pro IGBT
- ukázková odpověď: „724pF“
- 5.3.32. V_GT** - vrací V_GT (gate-Trigger-napětí)
- jen pro Thyristor und TRIAC
- ukázková odpověď: „865mV“
- 5.3.33. V_T** - vrací V_T zurück (Offset-Spannung)
- jen pro PUT
- ukázková odpověď: „699mV“
- 5.3.34. R_BB** - vrací R_BB (mezibázová rezistence)
- vyžaduje povolení detekce UJT
- jen pro UJT
ukázková odpověď: „4758R“
- 5.3.35. V_Z** - vrací V_Z (Zenerovo nebo externí napětí)
- potřebuje aktivovaný Zenerův test při běžném vyhledávání dílů
- pouze pro Zenerovu diodu nebo externí napětí (hardwarová volba)
- Příklad odpovědi: „6750mV“

Jak již bylo zmíněno, firmware lze přizpůsobit pro různé testery a další funkce. Existují některá nastavení v souboru Makefile, v config.h a config<MCU>.h. Tato kapitola vysvětluje nastavení. 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<MCU>.h je zodpovědný za věci na hardwarové úrovni, tedy za moduly LCD a přiřazení pinů.

6.1. 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).

6.1.1. MCU typ

```
# avr-gcc: MCU model
# - ATmega 328/328P : atmega328
# - ATmega 328PB    : atmega328pb
# - ATmega 324P/324PA : atmega324p
# - ATmega 640      : atmega640
# - ATmega 644/644P/644PA : atmega644
# - ATmega 1280     : atmega1280
# - ATmega 1284/1284P : atmega1284
# - ATmega 2560     : atmega2560
MCU = atmega328
```

Výpis 6.1. Předvolba je atmega328

6.1.2. Taktová frekvence MCU

```
# MCU frequency:
# - 1MHz : 1
# - 8MHz : 8
# - 16MHz : 16
# - 20MHz : 20
FREQ = 8
```

Výpis 6.2. Předvolba je 8MHz

6.1.3. Typ oscilátoru

```
# oscillator type
# - internal RC oscillator : RC
# - external full swing crystal : Crystal
# - external low power crystal : LowPower
OSCILLATOR = Crystal
```

Výpis 6.3. Předvolba je Crystal

6.1.4. Avrdude typ MCU

```
# avrdude: part number of MCU
# - ATmega 328 : m328
# - ATmega 328P : m328p
# - ATmega 328PB : m328pb
# - ATmega 324P : m324p
# - ATmega 324PA : m324pa
# - ATmega 640 : m640
# - ATmega 644 : m644
# - ATmega 644P : m644p
# - ATmega 644PA : m644p
# - ATmega 1280 : m1280
# - ATmega 1284 : m1284
# - ATmega 1284P : m1284p
# - ATmega 2560 : m2560
PARTNO = m328p
```

Výpis 6.4. Předvolba je m328p

6.1.5. Avrdude ISP programátor

Avrdude potřebuje:

Avrdude jméno programátora, bitový takt a port.

```
# Arduino as ISP
#PROGRAMMER = stk500v1
#PORT = /dev/ttyACM0
#OPTIONS = -b 19200

# Bus Pirate
#PROGRAMMER = buspirate
#PORT = /dev/bus_pirate
#OPTIONS = -B 10.0

# Diamex ALL-AVR/AVR-Prog
PROGRAMMER = avrispmkII
PORT = usb
OPTIONS = -B 1.0

# Pololu USB AVR Programmer
#PROGRAMMER = stk500v2
#PORT = /dev/ttyACM0
#OPTIONS = -B 1.0

# USBasp
#PROGRAMMER = usbasp
#PORT = usb
#OPTIONS = -B 20

# USBtinyISP
#PROGRAMMER = usbtiny
#PORT = usb
#OPTIONS = -B 5.0

# Arduino Uno bootloader via serial/USB
#PROGRAMMER = arduino
#PORT = /dev/ttyACM0
#OPTIONS = -D -b 115200

# Arduino Mega2560 bootloader via serial/USB
#PROGRAMMER = wiring
#PORT = /dev/ttyACM0
#OPTIONS = -D -b 115200
```

Výpis 6.5. Předvolba je Diamex

Pokud není tvůj programátor uvedený, přidej ho do Makefile ručně.

Další informace najdeš v příručce Avrdude nebo v online dokumentaci [9].

6.2. 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, odstraň znaky „//“ na začátku řádku a na deaktivování je zase na začátek řádku vlož. Některá nastavení vyžadují číselnou hodnotu, kterou můžeš případně upravit.

6.2.1. 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 zarážek 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 $\geq 5,5$ číslic DMM
- `#define UREF_25...2495` ** - Pokud je třeba změnit hodnotu

Ochranná relé pro vybíjení kondenzátorů

- Standardní pin:PC4 (ATmega 328)
- Nízký signál: kratší 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)
- děliče 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 Ω
 - ZENER_R2: spodní odpor v Ω
- ```
//#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
```

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 */ ** - změnit hodnotu
#define ZENER_VOLTAGE_MAX 30000 /* max. napětí v mV */ ** - změnit hodnotu
```

### Výstup pevného signálu

- pokud je pin OC1B MCU zapojen jako vyhrazený výstup signálu namísto odporu R1 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 nalezneš 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
 - vyber 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
- ```
//# define 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
- ```
//# define 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_p (v 0,1 pF)

- měl by být 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
- ```
//#define HW_BUZZER
```
- odkomentuj k aktivaci

### MAX6675 převodník termočládku

- 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 termočládků

- viz MAX31855_CS v config-<MCU>.h pro vyhrazený pin portu
 - vyžaduje sběrnici SPI a podporu čtení SPI
- ```
//#define HW_MAX31855
```
- odkomentuj k aktivaci

## 6.2.2. 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

### Generátor PWM: 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
```

- Vyber SW\_OLD\_ESR pro starou metodu měření od 180nF

```
//# define SW_OLD_ESR
```

\* - komentuj k odvolbě

- 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 Detector/Decoder (pomocí testovacích pinů)

- vyžaduje modul IR přijímače, např. série TSOP

- Modul bude připojen k testovacím pinům

```
#define SW_IR_RECEIVER
```

\* - komentuj k odvolbě

### 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

### 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

### **Test servo**

- vyžaduje další tlačítka a zobrazení s více než 2 řádky textu

```
//# define SW_SERVO
```

- odkomentuj k aktivaci

### **Měření teploty s DS18B20**

- aktivujte také ONEWIRE\_PROBES nebo ONEWIRE\_IO\_PIN (viz část „bussy“)

```
//#define SW_DS18B20
```

- 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 unikající 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)

### **Monitory C/L:** 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

//# definire 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
```

## 6.2.3. Řešení obcházení pro některé testery

**Deaktivace u některých testerů** měření hFE se společným kolektorovým obvodem a Rl jako bázevý 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
```

## 6.2.4. 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!

```
#ifndef OSC_STARTUP
#define OSC_STARTUP 16384
#endif
```

Výpis 6.6. změň hodnotu, pokud se neshoduje s testerem!

### 6.2.5. uživatelské rozhraní

**Dostupné jazyky** Standard je ISO 8859 -1.

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

```
#define UI_ENGLISH
//#define UI_CZECH
//#define UI_CZECH_2
//#define UI_DANISH
//#define UI_GERMAN
//#define UI_ITALIAN
//#define UI_POLISH
//#define UI_POLISH_2
//#define UI_ROMANIAN
//#define UI_RUSSIAN
//#define UI_RUSSIAN_2
//#define UI_SPANISH
```

Výpis 6.7. Zde je 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

**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

Řádky textu (doporučeno: > = 8 řádků)

//#define UI\_KEY\_HINTS

- odkomentuj k aktivaci

**Testovací profil** zobrazit při zapnutí testeru.

\* - pro testery přídatnými testovacími kabely

//# define UI\_SHOOOSE\_PROFILE

- odkomentuj k aktivaci

**Přidání třetího profilu** pro hodnoty nastavení.

- pro testery přídatnými 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 změň 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 změň 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`)

`//#define UI_PINOUT_ALT`

- odkomentuj k aktivaci

**zobrazení symbolu otazníku** při neúspěšném provedení testu

- vyžaduje zapnutí symbolů součástek (`SW_SYMBOLS`)

`//#define UI_QUESTION_MARK`

- odkomentuj k aktivaci

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

- vyžaduje zapnutí symbolů součástek (`SW_SYMBOLS`)

`//#define UI_ZENER_DIODE`

- odkomentuj k aktivaci

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

- vyžaduje zapnutí symbolů součástek (`SW_SYMBOLS`)

`//#define UI_QUARTZ_CRYSTAL`

- odkomentuj k aktivaci

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

- vyžaduje zapnutí symbolů součástek (`SW_SYMBOLS`)

`//#define UI_NO_TEXTPINOUT`

- odkomentuj k aktivaci

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

- vyžaduje zapnutí symbolů součástek (`SW_SYMBOLS`)

`//#define UI_NO_BODYDIODE_TEXTPINOUT`

- odkomentuj k aktivaci

**zobrazuje ID sondy pomocí obrácených barev**

- vyžaduje písmo s dalšími znaky

- dočasná volba `UI_PROBE_REVERSED_X` pro obrácené 'x', pokud je to možné protože ne všechna písmena tento znak obsahují (zkontrolujte písmo!)

`//#define UI_PROBE_REVERSED`

- odkomentuj k aktivaci

`//#define UI_PROBE_REVERSED_X`

- 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

**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

**Display Font Test**

```
//# definiere SW_FONT_TEST
```

- odkomentuj k aktivaci

**zobrazení symbolů komponentů** pro účely testování

- vyžaduje povolení symbolů (SW\_SYMBOLS) v kapitole 6.2.5 na stránce 52

```
//#define SW_SYMBOL_TEST
```

- odkomentuj k aktivaci

**Zaokrouhlení hodnot pro DS18B20**

- DS18B20 (0.1 °C/F)

```
//#define UI_ROUND_DS18B20
```

- odkomentuj k aktivaci

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

- Samoladící data se vždy ukládají do EEPROM

- Písma a symboly jsou vždy ukládány do Flash

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

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

- zvol jedno

## 6.2.6. 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

### **6.2.7. Nastavení a kompenzace měření**

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

- #define UREF\_VCC ... 5001 \*\* - v případě potřeby uprav hodnotu

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

- Kompenzuje rozdíly mezi skutečnou a změřenou hodnotou.
- ADC má rozlišení přibližně 4,88 mV pro V\_ref = 5V (Vcc) a 1,07 mV pro V\_ref = 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 Rl je 680Ω
- Výchozí hodnota pro Rh 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 Rh (470k) vΩ

- Pokud jsou odpory > 20k příliš vysoké nebo příliš nízké, uprav odpovídajícím způsobem ofset.
- 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
- Maximalní 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 $\mu$ F    |                    |
| CAP_FACTOR_MID                   | pro kondenzátory | 4,7 - 47 $\mu$ F |                    |
| CAP_FACTOR_LARGE                 | pro kondenzátory | > 47 $\mu$ F     |                    |
| #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

## 6.2.8. 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

## 6.2.9. 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
  - K aktivaci buď I2C\_BITBANG nebo I2C\_HARDWARE odkomentovat
  - jeden z Taktů odkomentovat
- ```
//# define I2C _BITBANG          bit-bang I2C
//# define I2C _HARDWARE         Hardware MCU TWI
//# define I2C _STANDARD _MODE 100kHz sběrníkové hodiny
//# define I2C _FAST _MODE      400kHz sběrníkové hodiny
//# define I2C _RW              povolit podporu čtení (netestováno)
```

* - 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
```

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

**Serial TTL Interface** může být již povolena 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



### 6.3. Config\_328.h

obsahuje nastavení na hardwarově blízké úrovni pro displeje, ovládání a tak dále. Při překladu firmware je podle v Makefile zvoleným MCU, automaticky integrován vhodný soubor. Pravidla komentování jsou stejné 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 “ “// “.

#### 6.3.1. LCD moduly

Displeje a jejich nastavení jsou v kapitole 2.6. na stránce 12. Zde je na příklad display ST765R, u kterého byly #if0 a #endif komentovány pomocí “// “.

##### Modul ST765R předvolba

```
//#if 0
#define LCD_ST765R /* display controller ST765R */
```

Výpis 6.8. Pomocí //if 0 a #define ST765R ... je správný modul aktivován

```
//#endif
```

Výpis 6.9. Konec smyčky je také vykomentován “// “.

#### 6.3.2. Rozložení portů a pinů

##### testovací piny/sondy

- první 3 piny analogového portu musí být použity pro zkušební piny.
- Neměň definici pro TP1, TP2 a TP3!

```
#define TP1 PC0 /* test pin / probe #1 */
#define TP2 PC1 /* test pin / probe #2 */
#define TP3 PC2 /* test pin / probe #3 */
```

Výpis 6.10. Piny neměnit.

##### Pevné předvolby

```
#define TP_ZENER PC3 /* test pin for Zener check (10:1 voltage divider) */
#define TP_REF PC4 /* test pin for 2.5V reference and relay */
#define TP_BAT PC5 /* test pin for battery (4:1 voltage divider) */
#define TP_LOGIC PC3 /* test pin for Logic Probe (4:1 voltage divider) */
```

Výpis 6.11. Neměnit!

##### Testovací odpory

- Pro výstup PWM/čtvercových vln přes zkušební pin 2 musí být R\_RL\_2 PB2/OC1B.
- Nesdílejte tento port s POWER\_CTRL nebo TEST\_BUTTON!

```
#define R_RL_1 PB0 /* Rl (680R) for test pin #1 */
#define R_RH_1 PB1 /* Rh (470k) for test pin #1 */
#define R_RL_2 PB2 /* Rl (680R) for test pin #2 */
#define R_RH_2 PB3 /* Rh (470k) for test pin #2 */
#define R_RL_3 PB4 /* Rl (680R) for test pin #3 */
#define R_RH_3 PB5 /* Rh (470k) for test pin #3 */
```

Výpis 6.12. Při potřebě upravit

##### Výdej signálu přes OC1B - Prosím neměň!

```
#define SIGNAL_OUT PB2 /* MCU's OC1B pin */
```

Výpis 6.13. Neměnit!

##### Hlavní vypínač - Nemůže být stejný port jako ADC\_PORT nebo R\_PORT.

```
#define POWER_CTRL PD6 /* control pin (1: on / 0: off) */
```

Výpis 6.14. Možné změnit

##### Testovací tlačítko - Nemůže být stejný port jako ADC\_PORT nebo R\_PORT.

```
#define TEST_BUTTON PD7 /* test/start push button (low active) */
```

Výpis 6.15. Možné změnit

## Rotační kodér

```
#define ENCODER_A PD3 /* rotary encoder A signal */
#define ENCODER_B PD2 /* rotary encoder B signal */
```

Výpis 6.16. Možné změnit

## Tlačítka Vice/méně

```
#define KEY_INC PD2 /* increase push button (low active) */
#define KEY_DEC PD3 /* decrease push button (low active) */
```

Výpis 6.17. Možné změnit

## Čítač kmitočtů

- Jednoduchá a rozšířená verze
- Vstup musí být pin PD4/T0

```
#define COUNTER_IN PD4 /* signal input T0 */
```

Výpis 6.18. neměnit!

## Kroužkový tester

- Vstup musí být pin PD4/T0

```
#define RINGTESTER_OUT PD5 /* pulse output */
```

Výpis 6.19. neměnit!

## IR detektor/dekodér

- pevný modul připojený k vyhrazenému I/O pinu

```
#define IR_DATA PD5 /* data signal */
```

Výpis 6.20. Možné změnit

## Bzučák

```
#define BUZZER_CTRL PD5 /* control pin (low: off / high: on) */
```

Výpis 6.21. Možné změnit

## MAX6675

```
#define MAX6675_CS PD5 /* port pin used for /CS */
```

Výpis 6.22. Možné změnit

## MAX31855

```
#define MAX31855_CS PD5 /* port pin used for /CS */
```

Výpis 6.23. Možné změnit

### 6.3.3. Busse

- SPI** - Hardware-SPI používá SCK PB5, MOSI PB3, MISO PB4 a /SS PB2  
- Bit-Bang-SPI může být již přes LCD zapnutý

```
/* for bit-bang and hardware SPI */
#ifndef SPI_PORT
#define SPI_PORT PORTB /* port data register */
#define SPI_DDR DDRB /* port data direction register */
#define SPI_PIN PINB /* port input pins register */
#define SPI_SCK PB5 /* pin for SCK */
#define SPI_MOSI PB3 /* pin for MOSI */
#define SPI_MISO PB4 /* pin for MISO */
#define SPI_SS PB2 /* pin for /SS */
#endif
```

Výpis 6.24. Možné změnit

- I2C** Hardware-I2C (TWI) používá SDA PC4 a SCL PC5  
- Bit-Bang-I2C může být již přes LCD zapnutý

```
/* for bit-bang I2C */
#ifndef I2C_PORT
#define I2C_PORT PORTC /* port data register */
#define I2C_DDR DDRC /* port data direction register */
#define I2C_PIN PINC /* port input pins register */
#define I2C_SDA PC4 /* pin for SDA */
#define I2C_SCL PC5 /* pin for SCL */
#endif
```

Výpis 6.25. Možné změnit

### seriální TTL interface

- Hardware-USART0 používá Rx PD0 & Tx PD1

```
/* for hardware RS232 */
#define SERIAL_USART 0 /* use USART0 */
/* for bit-bang RS232 */
#define SERIAL_PORT PORTD /* port data register */
#define SERIAL_DDR DDRD /* port data direction register */
#define SERIAL_PIN PIND /* port input pins register */
#define SERIAL_TX PD1 /* pin for Tx (transmit) */
#define SERIAL_RX PD0 /* pin for Rx (receive) */
#define SERIAL_PCINT 16 /* PCINT# for Rx pin */
```

Výpis 6.26. Možné změnit

### OneWire - vyhrazený I/O-Pin

```
#define ONEWIRE_PORT PORTD /* port data register */
#define ONEWIRE_DDR DDRD /* port data direction register */
#define ONEWIRE_PIN PIND /* port input pins register */
#define ONEWIRE_DQ PD5 /* DQ (data line) */
```

Výpis 6.27. Možné změnit

## 6.4. Config\_644.h

obsahuje nastavení na hardwarově blízké úrovni pro displeje, ovládání a tak dále. Při překladu firmware je podle v Makefile zvoleným MCU, automaticky integrován vhodný soubor. Pravidla komentování jsou stejné 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 “ “// “.

### 6.4.1. LCD moduly

Displeje a jejich nastavení jsou v kapitole 2.6. na stránce 12. Zde je na příklad display ILS9341, u kterého byly #if0 a #endif komentovány pomocí “// “.

#### Modul ILI9341 předvolba

```
//#if 0
#define LCD_ILI9341 /* display controller ILI9341/ILI9342 */
```

Výpis 6.28. Pomocí //if 0 a #define ILI9341 ... je správný modul aktivován

```
//#endif
```

Výpis 6.29. Konec smyčky je také vykomentován “// “.

### 6.4.2. Rozložení portů a pinů

#### testovací piny/sondy

- první 3 piny analogového portu musí být použity pro zkušební piny.
- Neměň definici pro TP1, TP2 a TP3!

```
#define TP1 PA0 /* test pin / probe #1 */
#define TP2 PA1 /* test pin / probe #2 */
#define TP3 PA2 /* test pin / probe #3 */
```

Výpis 6.30. Piny neměnit.

#### Pevné předvolby

```
#define TP_ZENER PA3 /* test pin for Zener check (10:1 voltage divider) */
#define TP_REF PA4 /* test pin for 2.5V reference and relay */
#define TP_BAT PA5 /* test pin for battery (4:1 voltage divider) */
#define TP_CAP PA7 /* test pin for self-adjustment cap */
#define TP_LOGIC PA6 /* test pin for Logic Probe (4:1 voltage divider) */
```

Výpis 6.31. Neměnit!

#### Testovací odpory

- Pro výstup PWM/čtvercových vln přes zkušební pin 2 musí být R\_RL\_2 PD4/OC1B.
- Nesdílejte tento port s POWER\_CTRL nebo TEST\_BUTTON!

```
#define R_RL_1 PD2 /* Rl (680R) for test pin #1 */
#define R_RH_1 PD3 /* Rh (470k) for test pin #1 */
#define R_RL_2 PD4 /* Rl (680R) for test pin #2 */
#define R_RH_2 PD5 /* Rh (470k) for test pin #2 */
#define R_RL_3 PD6 /* Rl (680R) for test pin #3 */
#define R_RH_3 PD7 /* Rh (470k) for test pin #3 */
```

Výpis 6.32. Při potřebě upravit

#### Výdej signálu přes OC1B - Prosím neměň!

```
#define SIGNAL_OUT PD4 /* MCU's OC1B pin */
```

Výpis 6.33. Neměnit!

#### Hlavní vypínač - Nemůže být stejný port jako ADC\_PORT nebo R\_PORT.

```
#define POWER_CTRL PC6 /* control pin (1: on / 0: off) */
```

Výpis 6.34. Možné změnit

**Testovací tlačítko** - Nemůže být stejný port jako ADC\_PORT nebo R\_PORT.

```
#define TEST_BUTTON PC7 /* test/start push button (low active) */
```

### Rotační kodér

Výpis 6.35. Možné změnit

```
#define ENCODER_A PC3 /* rotary encoder A signal */
#define ENCODER_B PC4 /* rotary encoder B signal */
```

Výpis 6.36. Možné změnit

### Tlačítka Vice/méně

```
#define KEY_INC PC4 /* increase push button (low active) */
#define KEY_DEC PC3 /* decrease push button (low active) */
```

Výpis 6.37. Možné změnit

### Čítač kmitočtů

- Jednoduchá a rozšířená verze
- Vstup musí být pin PB0/T0

```
#define COUNTER_IN PB0 /* signal input T0 */
```

Výpis 6.38. nicht ändern!

### Ovládání pro rozšířený čítač kmitočtů

```
#define COUNTER_CTRL_DIV PC0 /* prescaler (low 1:1, high x:1) */
#define COUNTER_CTRL_CH0 PC1 /* channel addr #0 */
#define COUNTER_CTRL_CH1 PC2 /* channel addr #1 */
```

Výpis 6.39. Možné změnit

### Kroužkový tester

- Frekvenční vchod má být zapojený na pinu PB0/T0

```
#define RINGTESTER_OUT PC0 /* pulse output */
```

Výpis 6.40. neměnit!

### L/C měřič

- Frekvenční vchod má být zapojený na pinu PB0/T0

```
#define LC_CTRL_CP PC0 /* reference cap (low: on / high: off) */
#define LC_CTRL_LC PC1 /* L/C selection (low: C / high: L) */
```

Výpis 6.41. neměnit!

### IR detektor/dekodér

- pevný modul připojený k vyhrazenému I/O pinu

```
#define IR_DATA PC2 /* data signal */
```

Výpis 6.42. Možné změnit

### Pevný kondenzátor pro Samočinné nastavení platí pro 470k odpor

- ADC-Pin je TP\_CAP shora
- měl by být filmový kondenzátor mezi 100nF und 1000nF

```
#define ADJUST_RH PC5 /* Rh (470k) for fixed cap */
```

Výpis 6.43. Možné změnit

### Bzučák

```
#define BUZZER_CTRL PC2 /* control pin (low: off / high: on) */
```

Výpis 6.44. Možné změnit

## MAX6675

```
#define MAX6675_CS PC2 /* port pin used for /CS */
```

Výpis 6.45. Možné změnit

## MAX31855

```
#define MAX31855_CS PC2 /* port pin used for /CS */
```

Výpis 6.46. Možné změnit

## Paralelní kondenzátorové relé (vzorkovací ADC)

- TP1 a TP3
- Kondenzátor by měl mít mezi 10nF a 27nF

```
#define CAP_RELAY_CTRL PC2 /* control pin */
```

Výpis 6.47. Možné změnit

### 6.4.3. Busse

**SPI** - Hardware-SPI používá SCK PB7, MOSI PB5, MISO PB6 a /SS PB4

- Bit-Bang-SPI může být již přes LCD zapnutý

```
/* for bit-bang and hardware SPI */
#ifndef SPI_PORT
#define SPI_PORT PORTB /* port data register */
#define SPI_DDR DDRB /* port data direction register */
#define SPI_PIN PINB /* port input pins register */
#define SPI_SCK PB7 /* pin for SCK */
#define SPI_MOSI PB5 /* pin for MOSI */
#define SPI_MISO PB6 /* pin for MISO */
#define SPI_SS PB4 /* pin for /SS */
#endif
```

Výpis 6.48. Možné změnit

**I2C** Hardware-I2C (TWI) používá PC1 a PC0

- Bit-Bang-I2C může být již přes LCD zapnutý

```
/* for bit-bang I2C */
#ifndef I2C_PORT
#define I2C_PORT PORTC /* port data register */
#define I2C_DDR DDRC /* port data direction register */
#define I2C_PIN PINC /* port input pins register */
#define I2C_SDA PC1 /* pin for SDA */
#define I2C_SCL PC0 /* pin for SCL */
#endif
```

Výpis 6.49. Možné změnit

### seriální TTL interface

- Hardware-USART0 používá PD0 & PD1, USART1 používá PD2 & PD3

```
/* for hardware TTL serial */
#define SERIAL_USART 0 /* use USART0 */
/* for bit-bang TTL serial */
#define SERIAL_PORT PORTD /* port data register */
#define SERIAL_DDR DDRD /* port data direction register */
#define SERIAL_PIN PIND /* port input pins register */
#define SERIAL_TX PD1 /* pin for Tx (transmit) */
#define SERIAL_RX PD0 /* pin for Rx (receive) */
#define SERIAL_PCINT 24 /* PCINT# for Rx pin */
```

Výpis 6.50. Možné změnit

**OneWire** - vyhrazený I/O-Pin

```
#define ONEWIRE_PORT PORTC /* port data register */
#define ONEWIRE_DDR DDRC /* port data direction register */
#define ONEWIRE_PIN PINC /* port input pins register */
#define ONEWIRE_DQ PC2 /* DQ (data line) */
```

Výpis 6.51. Možné změnit

## 6.5. Config\_1280.h

obsahuje nastavení na hardwarově blízké úrovni pro displeje, ovládání a tak dále. Protože přiřazení pinů závisí na MCU typu, existují zde různé soubory s příslušným přiřazením. Při překladač firmwaru je podle v Makefile zvoleným MCU, automaticky integrován vhodný soubor. Pravidla komentování jsou stejné 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. LCD moduly

Displeje a jejich nastavení jsou v kapitole 2.6. na stránce 12.

Zde je na příklad display ILS9486, u kterého byly #if0 a #endif komentovány pomocí “// “.

#### Modul ILI9486 předvolba

```
//#if 0
//#define LCD_ILI9481 /* display controller ILI9481 */
#define LCD_ILI9486 /* display controller ILI9486 */
//#define LCD_ILI9488 /* display controller ILI9488 */
```

Výpis 6.52. Pomocí // #if 0 a #define ILI9486 ... je správný modul aktivován

```
//#endif
```

Výpis 6.53. Konec smyčky je také vykomentován “// “.

### 6.5.2. Rozložení portů a pinů

#### testovací piny/sondy

- první 3 piny analogového portu musí být použity pro zkušební piny.
- Neměň definici pro TP1, TP2 a TP3!

```
#define TP1 PF0 /* test pin / probe #1 */
#define TP2 PF1 /* test pin / probe #2 */
#define TP3 PF2 /* test pin / probe #3 */
```

Výpis 6.54. Piny neměnit.

#### Pevné možnosti

```
#define TP_ZENER PF3 /* test pin for Zener check (10:1 voltage divider) */
#define TP_REF PF4 /* test pin for 2.5V reference and relay */
#define TP_BAT PF5 /* test pin for battery (4:1 voltage divider) */
#define TP_CAP PF7 /* test pin for self-adjustment cap */
#define TP_LOGIC PF6 /* test pin for Logic Probe (4:1 voltage divider) */
```

Výpis 6.55. Neměnit!

#### Testovací odpory

- Pro výstup PWM/čtvercových vln přes zkušební pin 2 musí být R\_RL\_2 PB6/OC1B.
- Nesdílejte tento port s POWER\_CTRL nebo TEST\_BUTTON!

```
#define R_RL_1 PK0 /* Rl (680R) for test pin #1 */
#define R_RH_1 PK1 /* Rh (470k) for test pin #1 */
#define R_RL_2 PK2 /* Rl (680R) for test pin #2 */
#define R_RH_2 PK3 /* Rh (470k) for test pin #2 */
#define R_RL_3 PK4 /* Rl (680R) for test pin #3 */
#define R_RH_3 PK5 /* Rh (470k) for test pin #3 */
```

Výpis 6.56. Při potřebě upravit

#### Výdej signálu přes OC1B - Prosím neměň!

```
#define SIGNAL_OUT PB6 /* MCU's OC1B pin */
```

Výpis 6.57. Neměnit!



**Hlavní vypínač** - Nemůže být stejný port jako ADC\_PORT nebo R\_PORT.

```
#define POWER_CTRL PA6 /* control pin (1: on / 0: off) */
```

Výpis 6.58. Možné změnit

**Testovací tlačítko** - Nemůže být stejný port jako ADC\_PORT nebo R\_PORT.

```
#define TEST_BUTTON PA7 /* test/start push button (low active) */
```

Výpis 6.59. Možné změnit

**Rotační kodér**

```
#define ENCODER_A PA3 /* rotary encoder A signal */
#define ENCODER_B PA1 /* rotary encoder B signal */
```

Výpis 6.60. Možné změnit

**Tlačítka Vice/méně**

```
#define KEY_INC PA3 /* increase push button (low active) */
#define KEY_DEC PA1 /* decrease push button (low active) */
```

Výpis 6.61. Možné změnit

**Čítač kmitočtů**

- Jednoduchá a rozšířená verze
- Vstup musí být pin PD7/T0

```
#define COUNTER_IN PD7 /* signal input T0 */
```

Výpis 6.62. nicht ändern!

**Ovládání pro rozšířený čítač kmitočtů**

```
#define COUNTER_CTRL_DIV PD4 /* prescaler (low 1:1, high x:1) */
#define COUNTER_CTRL_CH0 PD5 /* channel addr #0 */
#define COUNTER_CTRL_CH1 PD6 /* channel addr #1 */
```

Výpis 6.63. Možné změnit

**Kroužkový tester**

- Frekvenční vchod má být zapojený na pinu PD7/T0 sein (použij COUNTER\_IN)

```
#define RINGTESTER_OUT PD4 /* pulse output */
```

Výpis 6.64. neměnit!

**L/C měřič**

- Frekvenční vchod má být zapojený na pinu PB0/T0

```
#define LC_CTRL_CP PC4 /* reference cap (low: on / high: off) */
#define LC_CTRL_LC PC5 /* L/C selection (low: C / high: L) */
```

Výpis 6.65. neměnit!

**IR detektor/dekodér**

- pevný modul připojený k vyhrazenému I/O pinu

```
#define IR_DATA PA0 /* data signal */
```

Výpis 6.66. Možné změnit

**Pevný kondenzátor pro Samočinné nastavení** platí pro 470k odpor

- ADC-Pin je TP\_CAP shora
- měl by být filmový kondenzátor mezi 100nF und 1000nF

```
#define ADJUST_RH PA5 /* Rh (470k) for fixed cap */
```

Výpis 6.67. Možné změnit

**Bzučák**

```
#define BUZZER_CTRL PA2 /* control pin (low: off / high: on) */
```

Výpis 6.68. Možné změnit

## MAX6675

```
#define MAX6675_CS PA2 /* port pin used for /CS */
```

Výpis 6.69. Možné změnit

## MAX31855

```
#define MAX31855_CS PA2 /* port pin used for /CS */
```

Výpis 6.70. Možné změnit

## Paralelní kondenzátorové relé (vzorkovací ADC)

- TP1 a TP3
- Kondenzátor by měl mít mezi 10nF a 27nF

```
#define CAP_RELAY_CTRL PA2 /* control pin */
```

Výpis 6.71. Možné změnit

### 6.5.3. Busse

- SPI** - Hardware-SPI používá SCK PB1, MOSI PB2, MISO PB3 a /SS PB0  
- Bit-Bang-SPI může být již přes LCD zapnutý

```
/* for bit-bang and hardware SPI */
#ifndef SPI_PORT
#define SPI_PORT PORTB /* port data register */
#define SPI_DDR DDRB /* port data direction register */
#define SPI_PIN PINB /* port input pins register */
#define SPI_SCK PB1 /* pin for SCK */
#define SPI_MOSI PB2 /* pin for MOSI */
#define SPI_MISO PB3 /* pin for MISO */
#define SPI_SS PB0 /* pin for /SS */
#endif
```

Výpis 6.72. Možné změnit

- I2C** Hardware-I2C (TWI) používá SDA PD1 a SCL PD0  
- Bit-Bang-I2C může být již přes LCD zapnutý

```
/* for bit-bang I2C */
#ifndef I2C_PORT
#define I2C_PORT PORTD /* port data register */
#define I2C_DDR DDRD /* port data direction register */
#define I2C_PIN PIND /* port input pins register */
#define I2C_SDA PD1 /* pin for SDA */
#define I2C_SCL PD0 /* pin for SCL */
#endif
```

Výpis 6.73. Možné změnit

#### seriální TTL interface

- hardware USART používá
- |                         |                         |
|-------------------------|-------------------------|
| USART0: Rx PE0 a Tx PE1 | USART2: Rx PH0 a Tx PH1 |
| USART1: Rx PD2 a Tx PD3 | USART3: Rx PJ0 a Tx PJ1 |

```
/* for hardware TTL serial */
#define SERIAL_USART 0 /* use USART0 */
/* for bit-bang TTL serial */
#define SERIAL_PORT PORTE /* port data register */
#define SERIAL_DDR DDRE /* port data direction register */
#define SERIAL_PIN PINE /* port input pins register */
#define SERIAL_TX PE1 /* pin for Tx (transmit) */
#define SERIAL_RX PE0 /* pin for Rx (receive) */
#define SERIAL_PCINT 8 /* PCINT# for Rx pin */
```

Výpis 6.74. Možné změnit

#### OneWire - vyhrazený I/O-Pin

```
#define ONEWIRE_PORT PORTA /* port data register */
#define ONEWIRE_DDR DDRA /* port data direction register */
#define ONEWIRE_PIN PINA /* port input pins register */
#define ONEWIRE_DQ PA4 /* DQ (data line) */
```

Výpis 6.75. Možné změnit

pro různé modely testerů.

Pokud jsi dal zde neuvedený tester do provozu, zašli prosím krátký popis testeru a odpovídající nastavení na e-mail autora [8], aby to pomohlo ostatním uživatelům.

### 7.1. Arduino Nano, Uno nebo Mega 2560

- Nano/Uno: ATmega 328, 16MHz takt                      Mega 2560: ATmega 2560, 16MHz takt
- Stáhni si Arduino šema a pinout, abys viděl mapování jeho pinů. Příklad na stránce 79.
- Integrovaný USB2serial adaptér usnadňuje sériovou komunikaci.  
(aktivuj SERIAL\_HARDWARE a odpovídající možnosti).
- **NEPOUŽÍVEJ** piny TXD/RXD k ničemu jinému.
- Pokud chceš používat pin D13 Arduina, odstraň LED obvod, který by jinak zasahoval.
- Zde jsou uvedena pouze základní nastavení

```
#define HW_FIXED_SIGNAL_OUTPUT
```

Výpis 7.1. Hardwarové nastavení

```
#define UI_AUTOHOLD
```

Výpis 7.2. Uživatelské rozhraní

```
#define POWER_SWITCH_MANUAL
#define BAT_NONE
```

Výpis 7.3. Rozložení pinů k regulaci výkonu

```
#define ADC_LARGE_BUFFER_CAP /* 100nF cap at AREF */
```

Výpis 7.4. Nastavení měření a kompenzace

### 7.2. DIY Kit „AY-AT“ platí také pro GM328A

- ATmega328; Barevný LCD modul ST7735 (Bit-Bang-SPI)
- Rotační kodér (PD1 & PD3, paralelně k displeji)
- Externí referenční napětí 2,5 V (TL431)
- Jednoduchý čítač frekvence s vyhrazeným vstupem (PD4)
- Měření vnějšího napětí do 45 V (PC3)
- Nastavení jsou od flywheelz@EEVBlog
- nastavení pro polo-kompatibilní displej ST7735 poskytované b0hoon4@gmail.com

#### Upozornění:

- Některé modely GM328A mají polokompatibilní displej ST7735, který nebude fungovat s nastavením standardního ovladače ST7735. U těchto modulů následuje za přepínačem úrovní další integrovaný obvod (U3). V takovém případě použij ovladač Semi-ST7735.

```
#define HW_ENCODER
#define ENCODER_PULSES 4 /* usually 4 pulses per step */
#define ENCODER_STEPS 20 /* usually 20 detents */
#define HW_REF25
#define HW_ZENER
#define HW_FREQ_COUNTER_BASIC
```

Výpis 7.5. HW nastavení

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

#### Výpis 7.6. Rotační kodér

- Vstup pro čítač kmitočtu je PD4 (T0)

```
#define LCD_ST7735
#define LCD_GRAPHIC /* graphic display */
#define LCD_COLOR /* color display */
#define LCD_SPI /* SPI interface */
#define LCD_PORT PORTD /* port data register */
#define LCD_DDR DDRD /* port data direction register */
#define LCD_RES PD0 /* port pin used for /RESX */
#define LCD_CS PD5 /* port pin used for /CSX (optional) */
#define LCD_DC PD1 /* port pin used for D/CX */
#define LCD_SCL PD2 /* port pin used for SCL */
#define LCD_SDA PD3 /* port pin used for SDA */
#define LCD_DOTS_X 128 /* number of horizontal dots */
#define LCD_DOTS_Y 160 /* number of vertical dots */
// #define LCD_OFFSET_X 4 /* enable x offset of 2 or 4 dots */
// #define LCD_OFFSET_Y 2 /* enable y offset of 1 or 2 dots */
#define LCD_FLIP_X /* enable horizontal flip */
// #define LCD_FLIP_Y /* enable vertical flip */
#define LCD_ROTATE /* switch X and Y (rotate by 90Grad) */
// #define LCD_LATE_ON /* turn on LCD after clearing it */
#define FONT_10X16_HF /* 10x16 font */
#define SYMBOLS_24X24_HF /* 24x24 symbols */
#define SPI_BITBANG /* bit-bang SPI */
#define SPI_PORT LCD_PORT /* SPI port data register */
#define SPI_DDR LCD_DDR /* SPI port data direction register */
#define SPI_SCK LCD_SCL /* port pin used for SCK */
#define SPI_MOSI LCD_SDA /* port pin used for MOSI */
```

#### Výpis 7.7. LCD modul

Pokud by měl tester začít s prázdným displejem, odstraň komentář před LCD\_LATE\_ON.

```
#define LCD_DOTS_X 160 /* number of horizontal dots */
#define LCD_DOTS_Y 128 /* number of vertical dots */
#define LCD_LATE_ON /* turn on LCD after clearing it */
#define FONT_8X8_HF /* 8x8 font */
#define SYMBOLS_30X32_HF /* 30x32 symbols */
```

#### Výpis 7.8. polokompatibilní displej ST7735

- Korekční kompenzace indukčnosti pro model 20 MHz - poskytl indman@EEVBlog  
 - Uprav sekci režimu vysokého proudu ve funkci MeasureInductor () v induktoru.c

```
#if CPU_FREQ == 20000000
/* 20 MHz */
if (Temp < 1500) /* < 1.5us / < 100uH */
{
 Offset = -10;
}
else if (Temp < 5000) /* 1.5-5us / 100-330uH */
{
 Offset = -10;
}
else /* > 5us / > 330uH */
{
 Offset = -30;
}
#endif
```

#### Výpis 7.9. Kompenzace indukčnosti

### 7.3. BSide ESR02

- (DTU-1701) - ATmega 328, 8MHz takt
- ST7565 display (bit-bang SPI)
- Externí 2,5-V referenční napětí (TL431)
- Nastavení jsou od indman@EEVblog

```
#define HW_REF25
```

Výpis 7.10. Hardwarové nastavení

```
#define BAT_DIVIDER
#define BAT_EXT_UNMONITORED
#define BAT_R1 47000
#define BAT_R2 47000
#define BAT_OFFSET 420
```

Výpis 7.11. Rozložení pinů k regulaci výkonu

```
#define LCD_ST7565R /* display controller ST7565R */
#define LCD_GRAPHIC /* graphic display */
#define LCD_SPI /* SPI interface */
#define LCD_PORT PORTD /* port data register */
#define LCD_DDR DDRD /* port data direction register */
#define LCD_RESET PD0 /* port pin used for /RES (optional) */
#define LCD_A0 PD1 /* port pin used for A0 */
#define LCD_SCL PD2 /* port pin used for SCL */
#define LCD_SI PD3 /* port pin used for SI (LCD's data input) */
#define LCD_DOTS_X 128 /* number of horizontal dots */
#define LCD_DOTS_Y 64 /* number of vertical dots */
#define LCD_OFFSET_X /* enable x offset of 4 dots */
#define LCD_FLIP_X /* enable horizontal flip */
#define LCD_FLIP_Y /* enable vertical flip */
#define LCD_START_Y 0 /* start line (0-63) */
#define LCD_CONTRAST 15 /* default contrast (0-63) */
#define FONT_8X8_VF /* 8x8 font */
#define SYMBOLS_24X24_VFP /* 24x24 symbols */
#define SPI_BITBANG /* bit-bang SPI */
#define SPI_PORT LCD_PORT /* SPI port data register */
#define SPI_DDR LCD_DDR /* SPI port data direction register */
#define SPI_SCK LCD_SCL /* port pin used for SCK */
#define SPI_MOSI LCD_SI /* port pin used for MOSI */
```

Výpis 7.12. LCD Modul

## 7.4. Fish8840 TFT

- ATmega328, 8 MHz Takt; ST7565 display s (Bit-Bang-SPI)
- Externí 2,5-V referenční napětí (TL431)
- Poslal indman@EEVBlog/bdk100@vrtp.ru

```
#define LCD_ST7735
#define LCD_GRAPHIC /* graphic display */
#define LCD_COLOR /* color display */
#define LCD_SPI /* SPI interface */
#define LCD_PORT PORTD /* port data register */
#define LCD_DDR DDRD /* port data direction register */
#define LCD_RES PD3 /* port pin used for /RESX (optional) */
// #define LCD_CS PD5 /* port pin used for /CSX (optional) */
#define LCD_DC PD2 /* port pin used for D/CX */
#define LCD_SCL PD0 /* port pin used for SCL */
#define LCD_SDA PD1 /* port pin used for SDA */
#define LCD_DOTS_X 128 /* number of horizontal dots */
#define LCD_DOTS_Y 156 /* number of vertical dots */
#define LCD_OFFSET_X /* enable x offset of 4 dots */
#define LCD_OFFSET_Y /* enable y offset of 2 dots */
#define LCD_FLIP_X /* enable horizontal flip */
// #define LCD_FLIP_Y /* enable vertical flip */
#define LCD_ROTATE /* switch X and Y (rotate by 90Grad) */
#define LCD_LATE_ON /* turn on LCD after clearing it */
#define FONT_10X16_HF /* 10x16 font */
#define SYMBOLS_30X32_HF /* 30x32 symbols */
#define SPI_BITBANG /* bit-bang SPI */
#define SPI_PORT LCD_PORT /* SPI port data register */
#define SPI_DDR LCD_DDR /* SPI port data direction register */
#define SPI_SCK LCD_SCL /* port pin used for SCK */
#define SPI_MOSI LCD_SDA /* port pin used for MOSI */
```

Výpis 7.13. LCD modul

## 7.5. GM328 !pozor! né GM328A

- ten je nahoře pod 7.2 jako „AY-AT“.
- ATmega328, 8 MHz Takt; ST7565 display s (Bit-Bang-SPI)
- Nastavení jsou od rddube@EEVblog

```
#define LCD_ST7565R
#define LCD_GRAPHIC /* graphic display */
#define LCD_SPI /* SPI interface */
#define LCD_PORT PORTD /* port data register */
#define LCD_DDR DDRD /* port data direction register */
#define LCD_RESET PD0 /* port pin used for /RES (optional) */
#define LCD_A0 PD1 /* port pin used for A0 */
#define LCD_SCL PD2 /* port pin used for SCL */
#define LCD_SI PD3 /* port pin used for SI (LCD's data input) */
#define LCD_CS PD5 /* port pin used for /CS1 (optional) */
#define LCD_DOTS_X 128 /* number of horizontal dots */
#define LCD_DOTS_Y 64 /* number of vertical dots */
#define LCD_START_Y 0 /* start line (0-63) */
#define LCD_CONTRAST 11 /* default contrast (0-63) */
#define FONT_8X8_VF /* 8x8 font */
#define SYMBOLS_24X24_VFP /* 24x24 symbols */
#define SPI_BITBANG /* bit-bang SPI */
#define SPI_PORT LCD_PORT /* SPI port data register */
#define SPI_DDR LCD_DDR /* SPI port data direction register */
#define SPI_SCK LCD_SCL /* port pin used for SCK */
#define SPI_MOSI LCD_SI /* port pin used for MOSI */
```

Výpis 7.14. LCD modul



## 7.6. Hiland M644

- ATmega 644, 8 MHz takt; ST7565 LCD s (Bit-Bang-SPI)
- Rotační kodér (PB7 & PB5, paralelně k displeji)
- Externí 2,5-V referenční napětí (TL431)
- Převodník pro Zenerův test
- Rozšířený čítač frekvence  
(ale bez vstupního vyrovnávacího stupně pro přímé měření frekvence)
- Pevný kondenzátor pro auto nastavení  
(v případě problémů nahraď MLCC filmovým kondenzátorem 220nF)
- Nastavení jsou od Horst O. (obelix2007@mikrocontroller.net)

```
#define HW_ENCODER
#define ENCODER_PULSES 4 /* 4 */
#define HW_REF25
#define HW_ZENER
#define HW_FREQ_COUNTER_EXT
#define FREQ_COUNTER_PRESCALER 16 /* 16:1 */
#define HW_ADJUST_CAP
```

Výpis 7.15. Hardwarové nastavení

```
#define NO_HFE_C_RL /* if hFE values too high */
```

Výpis 7.16. Jinak při hFE-C nadměrné výsledky! **Aktivuj NO HFE C RL!**

```
#define LCD_ST7565R
#define LCD_GRAPHIC /* graphic display */
#define LCD_SPI /* SPI interface */
#define LCD_PORT PORTB /* port data register */
#define LCD_DDR DDRB /* port data direction register */
#define LCD_RESET PB4 /* port pin used for /RES (optional) */
// #define LCD_CS PB2 /* port pin used for /CS1 (optional) */
#define LCD_A0 PB5 /* port pin used for A0 */
#define LCD_SCL PB6 /* port pin used for SCL */
#define LCD_SI PB7 /* port pin used for SI (LCD's data input) */
#define LCD_DOTS_X 128 /* number of horizontal dots */
#define LCD_DOTS_Y 64 /* number of vertical dots */
// #define LCD_OFFSET_X /* enable x offset of 4 dots */
// #define LCD_FLIP_X /* enable horizontal flip */
#define LCD_FLIP_Y /* enable vertical flip */
#define LCD_START_Y 0 /* start line (0-63) */
#define LCD_CONTRAST 3 /* default contrast (0-63) */
#define FONT_8X8_VF /* 8x8 font */
#define SYMBOLS_24X24_VFP /* 24x24 symbols */
#define SPI_BITBANG /* bit-bang SPI */
#define SPI_PORT LCD_PORT /* SPI port data register */
#define SPI_DDR LCD_DDR /* SPI port data direction register */
#define SPI_SCK LCD_SCL /* port pin used for SCK */
#define SPI_MOSI LCD_SI /* port pin used for MOSI */
```

Výpis 7.17. LCD modul

```
#define POWER_CTRL PB1 /* controls power (1: on / 0: off) */
```

Výpis 7.18. Rozložení pinů k regulaci výkonu

```
#define ENCODER_A PB5 /* rotary encoder A signal */
#define ENCODER_B PB7 /* rotary encoder B signal */
```

Výpis 7.19. Rozložení pinů pro rotační kodér

```
#define KEY_INC PB5 /* increase push button (low active) */
#define KEY_DEC PB7 /* decrease push button (low active) */
```

Výpis 7.20. Rozložení pinů pro směr točení nahoru/dolu

```
#define ADJUST_RH PC6 /* Rh (470k) for fixed cap */
```

Výpis 7.21. Přiřazení vývodů pro pevný kondenzátor pro vlastní nastavení



## 7.7. M12864 DIY Transistor Tester

- ATmega328
- ST7565 display s (Bit-Bang-SPI); - rotační koder (PD1 & PD3, paralelně k display)
- Externí referenční napětí 2,5 V (TL431)

```
#define HW_ENCODER
#define ENCODER_PULSES 4 /* not confirmed yet, could be also 2 */
#define ENCODER_STEPS 24 /* not confirmed yet */
#define HW_REF25
```

Výpis 7.22. Hardwarové nastavení

```
#define LCD_ST7565R
#define LCD_GRAPHIC /* graphic display */
#define LCD_SPI /* SPI interface */
#define LCD_PORT PORTD /* port data register */
#define LCD_DDR DDRD /* port data direction register */
#define LCD_RESET PD0 /* port pin used for /RES */
#define LCD_A0 PD1 /* port pin used for A0 */
#define LCD_SCL PD2 /* port pin used for SCL */
#define LCD_SI PD3 /* port pin used for SI (LCD's data input) */
#define LCD_DOTS_X 128 /* number of horizontal dots */
#define LCD_DOTS_Y 64 /* number of vertical dots */
//#define LCD_OFFSET_X /* enable x offset of 4 dots */
#define LCD_FLIP_Y /* enable vertical flip */
#define LCD_START_Y 0 /* start line (0-63) */
#define LCD_CONTRAST 11 /* default contrast (0-63) */
#define FONT_8X8_VF /* 8x8 font */
#define SYMBOLS_24X24_VFP /* 24x24 symbols */
#define SPI_BITBANG /* bit-bang SPI */
#define SPI_PORT LCD_PORT /* SPI port data register */
#define SPI_DDR LCD_DDR /* SPI port data direction register */
#define SPI_SCK LCD_SCL /* port pin used for SCK */
#define SPI_MOSI LCD_SI /* port pin used for MOSI */
```

Výpis 7.23. LCD modul

```
#define ENCODER_PORT PORTD /* port data register */
#define ENCODER_DDR DDRD /* port data direction register */
#define ENCODER_PIN PIND /* port input pins register */
#define ENCODER_A PD3 /* rotary encoder A signal */
#define ENCODER_B PD1 /* rotary encoder B signal */
```

Výpis 7.24. Rotační koder

```
#define KEY_PORT PORTD /* port data register */
#define KEY_DDR DDRD /* port data direction register */
#define KEY_PIN PIND /* port input pins register */
#define KEY_INC PD3 /* increase push button (low active) */
#define KEY_DEC PD1 /* decrease push button (low active) */
```

Výpis 7.25. Rozložení pinů pro směr točení nahoru/dolu

## 7.8. MK-328

- ATmega328, 8 MHz Takt; ST7565 display s (Bit-Bang-SPI)
- Externí 2,5-V referenční napětí (TL431)
- Poslal brunosso@EEVblog a potvrdil Peeps@EEVblog

```
#define HW_REF25
```

Výpis 7.26. Hardwarové nastavení

```
#define LCD_ST7565R
#define LCD_GRAPHIC /* graphic display */
#define LCD_SPI /* SPI interface */
#define LCD_PORT PORTD /* port data register */
#define LCD_DDR DDRD /* port data direction register */
#define LCD_RESET PD0 /* port pin used for /RES (optional) */
#define LCD_CS PD5 /* port pin used for /CS1 (optional) */
#define LCD_A0 PD1 /* port pin used for A0 */
#define LCD_SCL PD2 /* port pin used for SCL */
#define LCD_SI PD3 /* port pin used for SI (LCD's data input) */
#define LCD_DOTS_X 128 /* number of horizontal dots */
#define LCD_DOTS_Y 64 /* number of vertical dots */
#define LCD_START_Y 0 /* start line (0-63) */
#define LCD_CONTRAST 11 /* default contrast (0-63) */
#define FONT_8X8_VF /* 8x8 font */
#define SYMBOLS_24X24_VFP /* 24x24 symbols */
#define SPI_BITBANG /* bit-bang SPI */
#define SPI_PORT LCD_PORT /* SPI port data register */
#define SPI_DDR LCD_DDR /* SPI port data direction register */
#define SPI_SCK LCD_SCL /* port pin used for SCK */
#define SPI_MOSI LCD_SI /* port pin used for MOSI */
```

## 7.9. T3/T4

Výpis 7.27. LCD modul

- ATmega328, 8 MHz takt; ST7565 display s (Bit-Bang-SPI)
- poskytl tom666@EEVblog

```
#define LCD_ST7565R
#define LCD_GRAPHIC /* graphic display */
#define LCD_SPI /* SPI interface */
#define LCD_PORT PORTD /* port data register */
#define LCD_DDR DDRD /* port data direction register */
#define LCD_RESET PD4 /* port pin used for /RES */
#define LCD_A0 PD3 /* port pin used for A0 */
#define LCD_SCL PD2 /* port pin used for SCL */
#define LCD_SI PD1 /* port pin used for SI (LCD's data input) */
#define LCD_CS PD5 /* port pin used for /CS1 (optional) */
#define LCD_DOTS_X 128 /* number of horizontal dots */
#define LCD_DOTS_Y 64 /* number of vertical dots */
#define LCD_START_Y 0 /* start line (0-63) */
#define LCD_CONTRAST 11 /* default contrast (0-63) */
#define FONT_8X8_VF /* 8x8 font */
#define SYMBOLS_24X24_VFP /* 24x24 symbols */
#define SPI_BITBANG /* bit-bang SPI */
#define SPI_PORT LCD_PORT /* SPI port data register */
#define SPI_DDR LCD_DDR /* SPI port data direction register */
#define SPI_SCK LCD_SCL /* port pin used for SCK */
#define SPI_MOSI LCD_SI /* port pin used for MOSI */
```

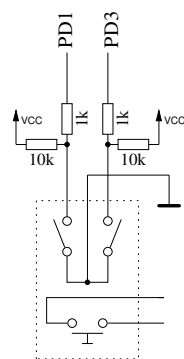
Výpis 7.28. LCD modul

Existují T4 varianty, které mají jiné nastavení:

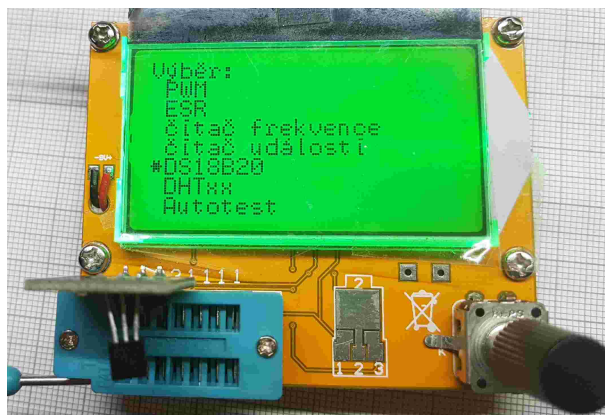
```
#define LCD_RESET PD0 /* port pin used for /RES */
#define LCD_A0 PD1 /* port pin used for A0 */
#define LCD_SCL PD2 /* port pin used for SCL */
#define LCD_SI PD3 /* port pin used for SI (LCD's data input) */
#define LCD_CS PD5 /* port pin used for /CS1 (optional) */
```

Výpis 7.29. LCD modul

a také přestavěný model s otočným koderem, podle Karl-Heinz Kübbeler [4].



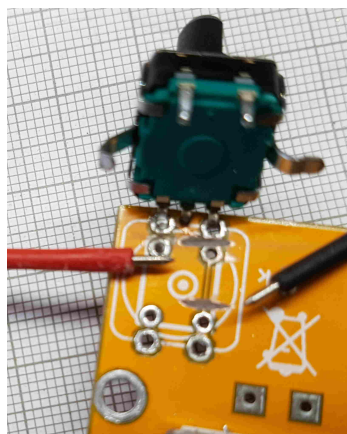
(a) Schema z khk knihy



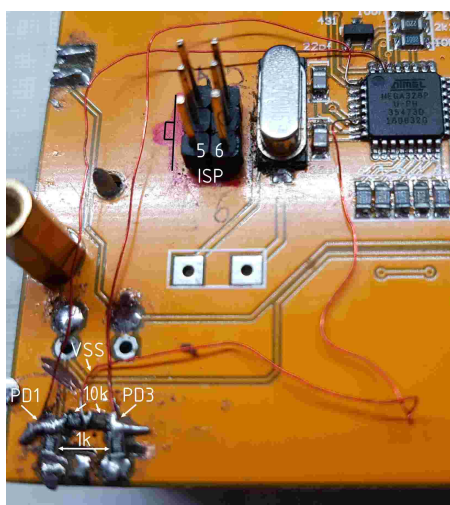
(b) přestavěný T4 s 6x8 ISO8859\_2 CZ textem.

Obrázek 7.1. T4 s kóděrem a část výběru.

Obrázky z úpravy:



(a) Nutná přerušení na přední straně



(b) a na zadní straně

Obrázek 7.2. T4 Úprava na kódér a ISP

Příklad fungující konfigurace v config.h:

```
#define HW_ENCODER
#define ENCODER_PULSES 4
#define ENCODER_STEPS 24
#define HW_REF25
#define UREF_25 2495
#define HW_FREQ_COUNTER_BASIC
#define HW_EVENT_COUNTER
#define EVENT_COUNTER_TRIGGER_OUT
#define SW_PWM_SIMPLE
#define SW_INDUCTOR
#define SW_ESR
#define SW_DS18B20
#define SW_REVERSE_HFE
#define SW_DHTXX
#define UI_COMMA
#define UI_AUTOHOLD
#define UI_KEY_HINTS
#define POWER_OFF_TIMEOUT 30
#define SW_POWER_OFF
#define UI_ROUND_DS18B20
#define ONEWIRE_PROBES
```

Výpis 7.30. Příklad konfigurace

## 7.10. Multifunktionstester TC-1 a rodina (T7) s ATmega324/644

- ATmega324/644 (velmi špatné rozložení pinů), - 16-MHz-Takt
- Barevný displej ST7735 (Bit-Bang-SPI)
- Externí 2,5-V referenční napětí (TL431)
- pevný modul IR přijímače; zesilovací převodník pro Zenertest (běží neustále, ne standardní dělič napětí 100k/12k) - Pevný kondenzátor pro vlastní ladění (v případě problémů nahraďte MLCC filmovým kondenzátorem 220nF) - Napájení pomocí Li-Ion baterie 3,7 V
- testovací kopie od jellytot@EEVblog a joystick@EEVblog
- první informace od indman@EEVblog

### Poznámky:

- Řídicí MCU U4 (STC15L104W) musí být nahrazen jednoduchým dvoutranzistorovým obvodem (TC1-Mod, viz zdrojové úložiště pro hardware/Markus/ TC1-Mod.kicad.tgz, pohotovostní režim 5μA proud celkem) nebo přeprogramován upraveným firmwarem (viz <https://github.com/atar-axis/tc1-u4>). Identifikátor může být také U3 (T7 Plus) nebo U5 u novějších desek.
- Nejnovější revize desky plošných spojů má také patičky pro alternativní MCU (podrobnosti neznámé). Hardwarová úprava by měla stále fungovat.
- Nastavte rozšířený bajt pojistky na 0xfd (detekce zhasnutí).
- Pokud se D2 (usměrňovací dioda pro Zenerovo testovací napětí) zahřeje, vyměňte ji za Schottkyho diodu určenou pro zpětné napětí 80 V nebo vyšší, např. SS18.
- Vyměňte C11 a C12 (uzávěry filtrů pro Zenerovo testovací napětí) za 10 nebo 22μF nízko-ESR elektrolytický kondenzátor dimenzovaný na 100 V nebo vyšší kvůli MLCC DC problému snížení výkonu kapacity předpětí.
- Na základě použitého modulu LCD možná budete muset místo LCD nastavit LCD\_FLIP\_X LCD\_FLIP\_Y.
- TC-1 nemůže poskytovat výstup signálu (PWM/squarewave/atd.) Na sondě # 2. Použití PD4 (OC1B) jako výstup vyhrazeného signálu (přidejte rezistor k omezení proudu) a povolit HW\_FIXED\_SIGNAL\_OUTPUT v config.h.
- Pokud chcete přidat otočný kodér nebo tlačítka pro/snížení, použijte prosím PB5 (displej D/C) a PB6 (displej SDA).
- Čítač kmitočtů můžete získat také pomocí PB0 (T0) jako vstupu a přidání jednoduchá vstupní fáze.
- pájecí můstek PD0 (nevyužitý firmwarem m)

```
#define HW_REF25
#define HW_ZENER
#define ZENER_DIVIDER_CUSTOM
#define ZENER_R1 100000
#define ZENER_R2 12000
#define ZENER_UNSWITCHED
#define HW_IR_RECEIVER
#define HW_ADJUST_CAP
```

Výpis 7.31. Hardwarové nastavení

```
#define HW_PROBE_ZENER
#define ZENER_VOLTAGE_MIN 1000 /* min. voltage in mV */
#define ZENER_VOLTAGE_MAX 40000 /* max. voltage in mV */
```

Výpis 7.32. Různé nastavení

```
#define IR_DATA PD3 /* data signal */
```

Výpis 7.33. Rozložení pinů pro pevný IR-Detektor/Decoder

```
#define ADJUST_RH PC6 /* Rh (470k) for fixed cap */
```

Výpis 7.34. Rozložení pinů pro pevný kondensator autotestu

```

#define TP_ZENER PA4 /* test pin with 10:1 voltage divider */
#define TP_REF PA3 /* test pin with 2.5V reference */
#define TP_BAT PA5 /* test pin with 4:1 voltage divider */
#define TP_CAP PA7 /* test pin for self-adjustment cap */

```

Výpis 7.35. Rozložení pinů

```

#define R_RL_1 PC0 /* Rl (680R) for test pin #1 */
#define R_RH_1 PC1 /* Rh (470k) for test pin #1 */
#define R_RL_2 PC2 /* Rl (680R) for test pin #2 */
#define R_RH_2 PC3 /* Rh (470k) for test pin #2 */
#define R_RL_3 PC4 /* Rl (680R) for test pin #3 */
#define R_RH_3 PC5 /* Rh (470k) for test pin #3 */

```

Výpis 7.36. Rozložení pinů pro testovací odpory

```

#define POWER_CTRL PD2 /* controls power (1: on / 0: off) */

```

Výpis 7.37. Rozložení pinů u regulace výkonu

```

#define TEST_BUTTON PD1 /* test/start push button (low active) */

```

Výpis 7.38. Rozložení pinů pro testovací tlačítko

```

#define LCD_ST7735
#define LCD_COLOR /* color graphic display */
#define LCD_SPI /* SPI interface */
#define LCD_PORT PORTB /* port data register */
#define LCD_DDR DDRB /* port data direction register */
#define LCD_RES PB4 /* port pin used for /RESX (optional) */
// #define LCD_CS PB? /* port pin used for /CSX (optional) */
#define LCD_DC PB5 /* port pin used for D/CX */
#define LCD_SCL PB7 /* port pin used for SCL */
#define LCD_SDA PB6 /* port pin used for SDA */
#define LCD_DOTS_X 128 /* number of horizontal dots */
#define LCD_DOTS_Y 160 /* number of vertical dots */
#define LCD_OFFSET_X 2 /* enable x offset of 2 or 4 dots */
#define LCD_OFFSET_Y 1 /* enable y offset of 1 or 2 dots */
// #define LCD_FLIP_X /* enable horizontal flip */
#define LCD_FLIP_Y /* enable vertical flip */
#define LCD_ROTATE /* switch X and Y (rotate by 90Grad) */
#define LCD_LATE_ON /* turn on LCD after clearing it */
#define FONT_10X16_HF /* 10x16 font */
#define SYMBOLS_30X32_HF /* 30x32 symbols */
#define SPI_BITBANG /* bit-bang SPI */
#define SPI_PORT LCD_PORT /* SPI port data register */
#define SPI_DDR LCD_DDR /* SPI port data direction register */
#define SPI_SCK LCD_SCL /* port pin used for SCK */
#define SPI_MOSI LCD_SDA /* port pin used for MOSI */

```

Výpis 7.39. LCD modul



## 7.11. Multifunktionstester T7 s ATmega328

- ATmega328 - 16-MHz-Takt
- Barevný displej ST7735 (hardwarové SPI)
- /RESET linka na samostatném portu MCU
- pevný modul IR přijímače
- zesilovací převodník pro Zenertest  
(běží neustále, cca 26V, ne standardní dělič napětí 100k/12k) - konektor pro sériové rozhraní TTL (J5, 5V)
- nastavení poskytl lhlad@EEVblog (viz [https://gitlab.com/a11059/t7\\_328\\_m](https://gitlab.com/a11059/t7_328_m)).

### Poznámky:

- Tento model nemůže poskytovat žádný výstupní signál (PWM/čtvercová vlna atd.)
- Účel přídavného MCU (STC15L104W) není znám.

```
#define HW_ZENER
#define ZENER_DIVIDER_CUSTOM
#define ZENER_R1 100000
#define ZENER_R2 12000
#define ZENER_UNSWITCHED
#define HW_IR_RECEIVER
```

Výpis 7.40. Hardwarové nastavení

```
#define HW_PROBE_ZENER
#define ZENER_VOLTAGE_MIN 500
#define ZENER_VOLTAGE_MAX 25500
```

Výpis 7.41. Různá nastavení

```
#define BAT_DIRECT
#define BAT_OFFSET 0
#define BAT_WEAK 3600
#define BAT_LOW 3400
```

Výpis 7.42. Rozložení pinů k regulaci výkonu

```
#define TP_ZENER PC3 /* test pin with 10:1 voltage divider */
#define TP_BAT PC4 /* test pin with 4:1 voltage divider */
```

Výpis 7.43. Rozložení testovacích pinů

```
#define R_RL_1 PD2 /* Rl (680R) for test pin #1 */
#define R_RH_1 PD3 /* Rh (470k) for test pin #1 */
#define R_RL_2 PD4 /* Rl (680R) for test pin #2 */
#define R_RH_2 PD5 /* Rh (470k) for test pin #2 */
#define R_RL_3 PD6 /* Rl (680R) for test pin #3 */
#define R_RH_3 PD7 /* Rh (470k) for test pin #3 */
```

Výpis 7.44. Rozložení pinů testovacích odporů

```
#define POWER_CTRL PB1 /* control pin (1: on / 0: off) */
```

Výpis 7.45. Rozložení pinu napájení

```
#define TEST_BUTTON PB4 /* test/start push button (low active) */
```

Výpis 7.46. Rozložení pinů pro testovací tlačítko

```
#define IR_DATA PB0 /* data signal */
```

Výpis 7.47. Rozložení pinů pro pevný IR-Detektor/Decoder

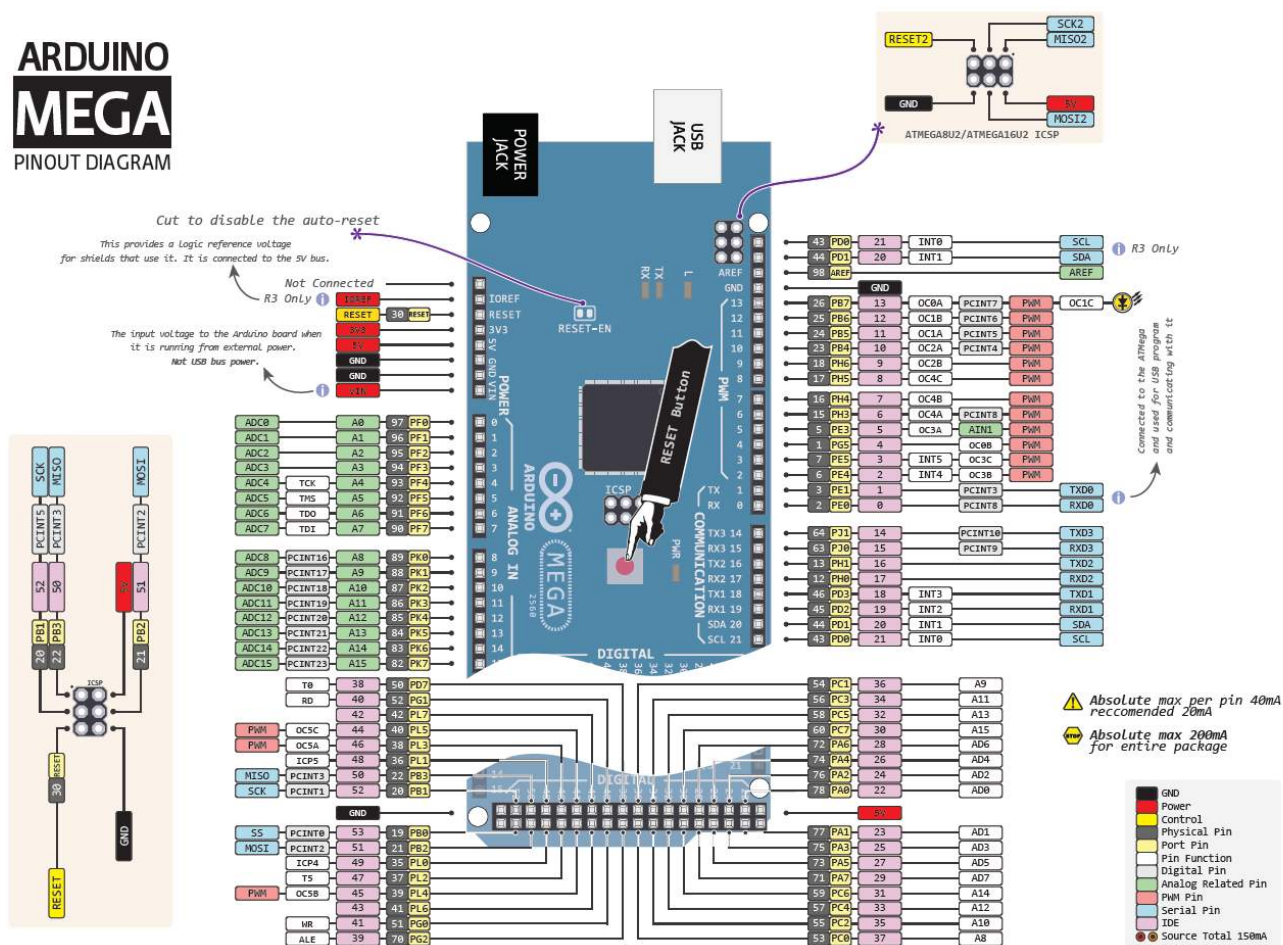
```

#define LCD_ST7735 /* display controller ST7735 */
#define LCD_GRAPHIC /* graphic display */
#define LCD_COLOR /* color display */
#define LCD_SPI /* SPI interface */
#define LCD_PORT PORTB /* port data register */
#define LCD_DDR DDRB /* port data direction register */
#define LCD_RES PC5 /* port pin used for /RESX (optional) */
#define LCD_RES_PORT PORTC /* separate port for /RESX (optional) */
#define LCD_RES_DDR DDRC /* separate port for /RESX (optional) */
// #define LCD_CS PB? /* port pin used for /CSX (optional) */
#define LCD_DC PB2 /* port pin used for D/CX */
#define LCD_SCL PB5 /* port pin used for SCL */
#define LCD_SDA PB3 /* port pin used for SDA */
#define LCD_DOTS_X 128 /* number of horizontal dots */
#define LCD_DOTS_Y 160 /* number of vertical dots */
#define LCD_OFFSET_X 2 /* enable x offset of 2 or 4 dots */
#define LCD_OFFSET_Y 1 /* enable y offset of 1 or 2 dots */
#define LCD_FLIP_X /* enable horizontal flip */
// #define LCD_FLIP_Y /* enable vertical flip */
#define LCD_ROTATE /* switch X and Y (rotate by 90Grad) */
// #define LCD_BGR /* reverse red and blue color channels */
#define LCD_LATE_ON /* turn on LCD after clearing it */
#define FONT_10X16_HF /* 10x16 font */
#define SYMBOLS_30X32_HF /* 30x32 symbols */
#define SPI_HARDWARE /* hardware SPI */

```

Výpis 7.48. LCD display

## 7.12. Arduino MEGA zapojení



Obrázek 7.3. Arduino Mega

Tento obrázek byl použit od: <https://duino4projects.com/arduino-mega-pinout-diagram/>

---

## Kapitola 8 Programování testeru Componentů

---

Aby zůstalo všem ostatním kolegům zoufalství a „bezesné noci“, kterými trpěl autor této kapitoly poté, co získal klone testera 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 testera.

Tato příležitost je zároveň využita, poděkovat autorovi a vývojáři tranzistorového testeru Karlovi-Heinzy Kübbelerovi viz [3] 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 [3].

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

### 8.1. Konfigurace testeru

K tomu si přečti kapitolu 1.8 od stránky 9.

### 8.2. 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 [9], nepotřebuješ mít žádnou přesnou znalost detailů pojistek.

Pokud si nejsi s nastavením pojistek jistý, nech je na poprvé nastavit standartě a nech testera 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.

### 8.3. 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.



## 8.4. 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 myši 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.

## 8.5. 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.

## 8.6. Stáhnutí zdrojů

a dokumentace

z archivu Git se dosáhne pomocí příkazu:

```
git clone https://github.com/Mikrocontroller-net/transistortester
```

Soubory jsou nyní v Linuxu [Osobní složka] na (/home/ „user“) pod názvem „transistortester“.

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" odstranit nebo přejmenovat, a poté celý archiv znovu stáhnout podle výše uvedených pokynů.

## 8.7. 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", ATTRS {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“.

## 8.8. 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:

**Nabídka menu** » **Správa systému** » **Uživatelé a skupiny** » **?heslo** se zobrazí okno se dvěma záložkami.

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

## 8.9. 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 v systémové liště se zelenou ikonou (Nemo) složky do /transistortester/Software/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 **ST**:

```
cd transistortester/Software/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 **Enter**!:

```
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 **ST** na myši. Teprve nyní použij **Enter**. 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/Software//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.**

## 8.10. 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)      |

Nyní zbývá jen radost po dosaženém úspěchu.

**9.1. v1.47m 2022-07**

Přidána možnost zobrazení symbolu křemenného krystalu pro režimy LF/HF v rozšířeném režimu (UI\_QUARTZ\_CRYSTAL, navrhl indman@EEVblog).

- Přidány fonty font\_8x12t\_iso8859-2\_hf.h (FONT\_8X12T\_ISO8859\_2\_HF) a

font\_8x16\_iso8859-2\_hf.h (FONT\_8X16\_ISO8859\_2\_HF, díky b0hoon@EEVblog).

- Opraven chybný odkaz na ' \ ' ve font\_8x8\_iso8859-2\_h.h, rozbitý symbol diody A-C a 'W' v souboru font\_6x8\_iso8859-2\_hf.h (díky b0hoon@EEVblog).

- Přidána možnost zobrazit symbol Zenerovy diody v libovolné kontrole Zenerovy diody (UI\_ZENER\_DIODE, navrhl indman@EEVblog).

- Nový symbol 32x39 nastavený tak, aby odpovídal 26 pixelů vysokým písmům (SYMBOLS\_32X39\_HF, díky indman@EEVblog).

- Přidány možnosti automatického udržování ESR a indukčnosti pro monitor C a L (SW\_MONITOR\_HOLD\_ESR, SW\_MONITOR\_HOLD\_L, navrhl theHWcav@EEVblog).

- Do funkce Serial\_Char() byly přidány překlady znaků pro inverzní čísla pinů.

- Do ovladače ST7735 byla přidána volba pro linku /RESX na samostatném portu MCU (LCD\_RES\_PORT, LCD\_RES\_DDR, na základě nápadu od lhlad@EEVblog).

- Přidána možnost zobrazení symbolu otazníku v případě neúspěšného testu. (UI\_QUESTION\_MARK, návrh indman@EEVblog).

- Přidány symboly otazníku, Zenerovy diody a křemenného krystalu do všech funkcí. (díky indman@EEVblog).

- Přidány další příznaky umístění pro efektní vývody (PIN\_CENTER, PIN\_ALT\_CENTER, navrženo indman@EEVblog).

- Zobrazení malého symbolu diody před Vf pro MOSFETy a IGBT (navrhl indman@EEVblog).

- Přidány číslice 1-3 a znak x s obrácenou barvou plus znak stupně do souboru

font\_8x16alt\_win1251\_hf.h. Rovněž aktualizováno číslo 1-3 a x s obrácenou barvou v textu font\_8x8alt\_win1251\_vf.h, totéž pro font\_16x26\_win1251\_hf.h (díky indman@EEVblog).

- Přidána možnost zakázat zobrazování textu na základě vývodů diody těla pro MOSTFET (UI\_NO\_BODYDIODE\_TEXTPINOUT, navrhl indman@EEVblog).

- Přidána sada symbolů 32x39 (SYMBOLS\_32X39\_HF, díky indman@EEVblog).

- Přidány alternativní sady symbolů 24x24 (SYMBOLS\_24X24\_ALT1\_H, SYMBOLS\_24X24\_ALT2\_H, SYMBOLS\_24X24\_ALT1\_HF, SYMBOLS\_24X24\_ALT2\_HF, SYMBOLS\_24X24\_ALT1\_VFP, SYMBOLS\_24X24\_ALT2\_VFP, SYMBOLS\_24X24\_ALT1\_VP\_F a SYMBOLS\_24X24\_ALT12\_VP\_F), 30x32 (SYMBOLS\_30X32\_ALT1\_HF a SYMBOLS\_30X32\_ALT2\_HF) a 32x32 (SYMBOLS\_32X32\_ALT1\_HF a SYMBOLS\_32X32\_ALT2\_HF, díky indman@EEVblog).

- Do efektního rozpisu vývodů byla přidána možnost zobrazovat čísla sond vpravo nahoře/pod nimi. symbolu (UI\_PINOUT\_ALT, návrh indman@EEVblog).

- Opraven problém ve funkci FontTest() s chybějícími mezerami u některých ovladačů displeje.

- Přidán testovací výstup symbolů komponent (SW\_SYMBOL\_TEST), navrhl uživatel indman@EEVblog.

- Aktualizovány ruské texty (díky indman@EEVblog).

- Opraven chybějící znak ')' a nefunkční znak '\*' v souboru font\_10x16\_hf.h (nahlásil uživatel pepe10000@EEVblog).

## 9.2. v1.46m 2022-04

- Možnost vypnutí textového rozpisu vývodů pro 3pinové polovodiče, když jsou aktivovány symboly komponentů (UI\_NO\_TEXTPINOUT, návrh podle bffargo@EEVblog).
- Identifikátor vývodu pro diodu MOSFET, volnoběžnou diodu IGBT, volnoběžnou diodu BJT, bázeový emitorový odpor a bázeová kolektorová dioda Schottkyho tranzistoru, jsou nyní také barevně označeny.
- Vylepšený výstupní symbol pro symetrický JFET (nyní zobrazuje 'x' pro drain a source).
- Programovací logika a výstupní označení vývodů pro 3pinové polovodiče revidované.
- Aktualizované rumunské texty (díky Dumidan@EEVblog).
- Font\_8x12t\_win1251\_vfp.h aktualizován (díky indman@EEVblog).
- Revize symbolů složek. Odstranění kroužků a přidání označení kolíků ( návrh indman@EEVblog). Staré symboly lze stále používat ( set SYMBOLS\_<size>\_OLD\_<format>.h in config\_<MCU>.h).
- Zobrazení testovacích pinů v invertovaném formátu (UI\_PROBE\_REVERSED, návrh Feliciano@EEVblog).
- Přidány číslice 1-3 v invertované reprezentaci v několika znakových sadách ( Díky Feliciano@EEVblog).
- Aktualizovány ruské texty (díky indman@EEVblog).
- Aktualizovány španělské texty (díky pepe10000@EEVblog).
- Aktualizované rumunské texty (díky Dumidan@EEVblog).
- Polské texty #2 aktualizovány (díky Jacon@EEVblog).

## 9.3. v1.45m 2021-12

- Nové možnosti hardwaru: Termočlánekový převodník MAX6675 a MAX31855 HW\_MAX6675, HW\_MAX31855, (návrh ricktendo@EEVblog).
- Volitelný třetí profil pro hodnoty nastavení UI\_THREE\_PROFILES (návrh podle indman@EEVblog).
- Nová funkce: kontrola kontinuity SW\_CONTINUITY\_CHECK, (návrh by indman@EEVblog).
- Nová hardwarová možnost: akustický signál, tj. aktivní bzučák HW\_BUZZER, (návrh indman@EEVblog).
- Nová hardwarová volba: logický tester HW\_LOGIC\_PROBE, (navrženo uživatelem Dumidan@EEVblog). - Symbol diody A-C, Ä a Ü ve font\_12x16\_hf.h a font\_12x16\_iso8859-2\_hf.h opraveno (uvádí Feliciano@EEVblog).
- Nová hardwarová volba: tester kroužků (LOPT/FBT tester, HW\_RING\_TESTER).
- Možnost pro hlavní menu: buď zůstat v menu, nebo automaticky opustit menu (UI\_MAINMEN). (UI\_MAINMENU\_AUTOEXIT).
- Font\_8x8\_hf.h aktualizován (díky Feliciano@EEVblog).

## 9.4. v1.44m 2021-08

- Výstup kódů EIA-96 pro standardní hodnoty odporu E96 SW\_E96\_EIA96, (návrh Chris02@EEVblog).
- V ovladačích pro ILI9481 a ILI9486. byla přidána podpora čtyřvodičového SPI.
- Opraven problém s časovým limitem u ovladače ST7920. (uvádí ralleiner@mikrocontroller.net).
- Změna vybíjecí funkce: V případě problému s vybíjením se napětí bez zátěže na nezatížené napětí DUT je měřeno a indikováno chybovým hlášením o vybití. např. napětí bez zatížení baterie.
- Symbol zkroucené diody pro druhou diodu při antiparalelním výstupu diody (uvádí horo@EEVblog).
- Možnost nízkoaktivního podsvícení v ovladači HD44780 pro adaptér PCF8574-I2C. přidáno LCD\_BACKLIGHT\_LOW, (navrhl Andreas Deimel).
- Nesprávné symboly pro p-kanálové tranzistory MOSFET v bitových mapách 24x24 symbolů opraveno (informoval Peeps@EEVblog).

## 9.5. v1.43m 2021-03

- Přidán R&D ovladač displeje pro výzkum a vývoj pro identifikaci řadičů displeje.
- Řídicí signál LCD\_RD je nyní volitelný pro ILI9341, ILI9481, ILI9486 a ILI9486.
- Přidán upravený ovladač pro polokompatibilní displeje ST7735 (navržený uživatelem b0hoon4@gmail.com).
- Možnost spustit Zenerovu kontrolu během normálního sondování (HW\_PROBE\_ZENER, navrženo. indman@EEVblog). Vyžaduje ZENER\_UNSWITCHED.
- Přidána také odpovídající příkaz pro vzdálenou kontrolu (V\_Z).
- Přidána možnost zadat vlastní dělič napětí pro Zenerovu kontrolu ( ZENER\_DIVIDER\_CUSTOM, ZENER\_R1 a ZENER\_R2).
- Odstraněna možnost nízkého rozlišení při Zenerově kontrole a také příkaz ZENER\_HIGH\_RES. přepínač konfigurace.
- Aktualizovány ruské texty (díky indman@EEVblog).
- Volitelné zobrazení doby trvání pulzů pro generátory PWM (PWM\_SHOW\_DURATION, navrhl host „hamburger“ @mikrocontroller.net).
- Archiv zdrojových kódů obsahuje adresář „dep“, aby byly některé IDE spokojené (navrženo DAIRVINE@EEVblog).
- Přejmenován konfigurační přepínač SW\_PROBE\_COLORS na UI\_PROBE\_COLORS.
- Nová funkce pro různé barvy nadpisů (UI\_COLORED\_TITLES) a také kurzoru a klíčové nápovědy (UI\_COLORED\_CURSOR).
- Aktualizované rumunské texty (díky Dumidan@EEVblog).
- Aktualizovány alternativní polské texty (díky Jacon@EEVblog).
- Aktualizovány španělské texty (díky pepe10000@EEVblog).

## 9.6. v1.42m 2020-12

- Aktualizovaná písma 6x8 (díky Bohu).
- Nová znaková sada 6x8 podle ISO8859-2 FONT\_6X8\_ISO8859\_2\_HF, (díky Bohu).
- Nová možnost hardwaru: LC metr (HW\_LC\_METER). Na základě jednoduchého Obvodu s LC oscilátorem, podobně jako cenově dostupná stavebnice LC měřiče.
- Display\_Value () nyní také podporuje Femto (f).
- Pokud je CYCLE\_MAX nastaven na 255, je automatické vypnutí deaktivováno a tester běží, dokud se ručně nevypne.
- Problém se zobrazením podivných frekvencí v rozšířeném počítadle frekvencí v určité konstelaci byla odstraněna (uvádí Szybkijanek@EEVblog).
- Jednoduchý čítač kmitočtů nyní zobrazuje „-“ místo „0Hz“, pokud není žádný Signál zapojený a nebo je frekvence příliš nízká.
- Možnost výstupu zkušebního proudu I\_C nebo I\_E měření hFE ( SW\_HFE\_CURRENT, návrh od Obelix2007@EEVblog).
- Také byly přidány příkazy dálkového ovládání (I\_C a I\_E).
- Rumunské texty (díky Dumidan@EEVblog). - Přepínač konfigurace pro nástroj ESR (SW\_ESR\_TOOL, návrh od indman @ EEVblog).
- Textový výstup hodnot E-normy nyní také vydává jednotku (návrh od indman @ EEVblog).
- Alternativní polské texty aktivovány (díky Jacon@EEVblog).
- Poslední řádek textu na displeji byl jednoduše přepsán, když UI\_KEY\_HINTS je aktivován (nahlášeno Obelix2007@EEVblog).
- Nová funkce integrovaného čekání na uživatele před odstraněním posledního řádku.
- Španělské texty aktualizovány (díky pepe10000@EEVblog).
- Opravená chyba v "#define" pro standardní hodnoty E96 (hlášené uživatelem Obelix2007@EEVblog).
- Ovladač pro displej s ILI9488 8/16 bitový paralelní a 4vodičový SPI, (díky Bohu za zkušební kopii).

## 9.7. v1.41m 2020-09

- Nový příkaz dálkového ovládání "MHINT" pro informace o měření (aktuálně pouze Typ zku-

šebního obvodu pro h\_FE).

- Automatický režim (automatická aktualizace) zabudovaný pro DS18B20 (návrh od Obelix2007@EEVblog).
  - Další funkce vykládky, která u některých eliminuje problém s ESR Testerclones (uvádí indman@EEVblog). ESR byla příliš vysoká  $Elko > = 470\mu F$  na testovacích pinech # 1 a # 2, v nástroji ESR, C monitoru a RCL monitoru. A pro  $Elko > = 4700\mu F$  ve všech režimech. Dotčenými testery jsou Hiland M644 a Varianty TC-1.
  - Aktualizované znaky stupně v font\_16x26\_hf.h a font\_16x26\_iso8859-2\_hf.h (Změny provedené Bohu).
  - Ovladač pro řadič OLED SH1106 3vodičový SPI, 4vodičový SPI a I2C; (díky Old-Papa pro zkušební zobrazení).
  - Chybějící inicializace signálu / RES pro I2C v ovladači SSD1306 přidán.
  - Výstup typu testovacího obvodu pro hFE.
  - Změněné zobrazení hexadecimálních hodnot na malá písmena s možností přepnutí na aktivaci velkých písmen (UI\_HEX\_UPPERCASE).
  - Možnost funkcí OneWire pro další odečítání a zobrazení Kódy ROM ONEWIRE\_READ\_ROM, (návrh od indman@EEVblog).
  - DQ a Vcc testovací kolíky pro sběrnici OneWire vhodné pro přiřazení pinů senzorům vyměněno (návrh od indman@EEVblog).
  - Detekce Schottkyho tranzistorů (SW\_SCHOTTKY\_BJT).
  - Vylepšení detekce typů vyčerpání FET pomocí filtrů pro Schottkyho tranzistory.
  - Správa textových řádků optimalizovaná pro vymazání displeje.
  - Speciální ovladač displeje pro identifikaci řadičů displeje.
  - Monitor R/C/L SW\_MONITOR\_RCL, (návrh od indman@EEVblog).
  - Vyřešen problém se změnou hodnot indukčnosti v monitoru L a R/L (nahlášeno od indman@EEVblog).
  - Měření V\_GS (vypnuto) pro FET typu vyčerpání (návrh od joshto@EEVblog).
- Příkazy dálkového ovládání rozšířené o příkaz "V\_GS\_off".
- ovladač ILI9341 rozšířen o podporu 8bitové paralelní sběrnice. Dále Konfigurační přepínač pro ILI9341 s deaktivovaným vysunutým Přidána sada příkazů LCD\_EXT\_CMD\_OFF, (díky Bohu za zkušební kopii).
  - Sada znaků cyrilice 16x26 (FONT\_16X26\_WIN1251\_HF, díky Yuriy\_K@VRTP.RU).
  - Problém s chybějícím  $\mu$  (mikro) v několika znakových sadách ISO8859-2 opraven. (nahlášeno indman@EEVblog a Obelix2007@EEVblog).
  - Opravené rotace čísel pro ATmega 2560 v Makefile (nahlášeno Bohu).
  - Barevný stav baterie pro barevné displeje (návrh od indman@EEVblog).
  - Alternativní znaková sada 8x16 Win1251 prostřednictvím aktualizované verze indman@EEVblog nahrazen.
  - Problém s validací malých odporů ve Windows CheckResistor () odstraněn (nahlášeno indman@EEVblog).
  - SmallResistor (), MeasureInductance () a GetGateThreshold () pro dílčí podpora ADC\_LARGE\_BUFFER\_CAP aktualizována.
  - Chybějící aktualizace referenčního zdroje ADC v MeasureInductance () a přidáno GetGateThreshold ().
  - Možnost výběru kalibračního profilu po zapnutí UI\_CHOOSE\_PROFILE, (návrh Bohu).
  - Vlastní monitorovací funkce pro rezistory a indukčnosti SW\_MONITOR\_R, SW\_MONITOR\_L, (návrh od indman@EEVblog).
  - Španělské texty aktualizovány (díky pepe10000@EEVblog).

## 9.8. v1.40m 2020-06

- Ovladač pro displeje s ILI9481 nebo ILI9486.
- Logická chyba pro barevné displeje s vypnutou funkcí barev.
- Přidán konfigurační přepínač pro barevné displeje pro výměnu Barevné kanály pro červenou a modrou (LCD\_BGR).

- Vylepšený monitor R / L pro snížení šíření naměřených hodnot. Nepomáhá ve všech případech. (hlášeno indmanem @ EEVblog).
- Problém s funkcí „#ifdef“ pro funkce Display\_HexByte() a Display\_HexDigit() opraveno. (hlášeno AlcidePiR2@EEVBlog).
- Podpora pro ATmega 640/1280/2560.
- Detekce směru otáčení v ReadEncoder() byla obrácena. Správně Směr byl změněn a nastavení v config\_<MCU>.h a klonech aktualizováno.
- Opraven problém s inicializací s hardwarovým SPI.
- Alternativní režim pro Zenertest, když je spuštěn převodník posilovače nebo neexistuje vůbec (ZENER\_UNSWITCHED, (navrhl indman @ EEVblog).
- Konfigurační přepínač pro 100nF AREF vyrovnávací kondenzátor namísto 1nF (ADC\_LARGE\_BUFFER\_CAP). Vyžadováno pro některé desky MCU.
- Alternativní cyrilická písma 8x16 (FONT\_8X16ALT\_WIN1251\_HF) a 8x8 (FONT\_8X8ALT\_WIN1251\_VF, díky indmanu@EEVblog).
- Aktualizované ruské texty (díky indmanu@EEVblog).
- Možnost použít jednoduchý vypínač místo standardního softwarového řešení (POWER\_SWITCH\_MANUAL).
- Přidána Detekce dvou krátkých doteků ve středu v ReadTouchScreen().
- Chyba algoritmu týkající se TOUCH\_FLIP\_X/TOUCH\_FLIP\_Y v Touch\_CharPos() opraveno (hlášeno Bohu).
- Více ISO8859-2 znakových sad prostřednictvím aktualizovaných verzí od Bohu nahrazeno.
- Možnost otestovat odpory pro standardní hodnoty E (SW\_R\_E\*).
- Platí také pro kondenzátory (SW\_C\_E\*) a induktory (SW\_L\_E\*).

### 9.9. v1.39m 2020-03

- Polské texty podle ISO 8859-2 (díky Jacon).
- Problémy s makrem preprocesoru v SPI.c a syntaktickými chybami v ADS7843.c opraven (hlášeno Bohu).
- Možnost uložit data firmwaru na flash místo na EEPROM DATA\_FLASH, (Navrhl Vitaliy).
- Cyrilské znakové sady přejmenované na „win1251“ a české znaky- sady nahrazeny znakovými sadami ISO8859-2 (díky Bohu).
- Funkce pro výstup znakové sady pro účely testování (SW\_FONT\_TEST).
- Test OneWire pro výstup kódů ROM uživatelů sběrnice (SW\_ONEWIRE\_SCAN).
- Možnost přesného offsetu odporu je pár testovacích kabelů. (R\_MULTIOFFSET), Navrhl Vitaliy).

### 9.10. v1.38m 2019-12

- Volitelné zaokrouhlení teploty pro DS18B20 (UI\_ROUND\_DS18B20, (návrh Obelix2007@EEVblog).
  - Podpora DHT11, DHT22 a kompatibilních senzorů (SW\_DHTXX. (Díky indman@EEVblog a Obelix2007@EEVblog za testování).
  - Byly přidány dvě tenké znakové sady s cyrilikou (díky Andrey@EEVblog).
  - Výkon bipolárních tranzistorů se změnil tak, že V\_BE a hFE jsou nyní také v případě s B-E-odporem ukázané.
- Také příkazy pro dálkové ovládání bylo odpovídajícím způsobem upraveno.
- České texty plus několik znakových sad s češtinou, (díky Bohu)
  - Funkce pro monitorování R/L a C (SW\_MONITOR\_RL a SW\_MONITOR\_C, (navrhl indman@EEVblog).
  - Spouštěcí výstup pro počítadlo událostí (doporučeno od Bohu).
  - Aktualizované české texty (díky Bohu).
  - Měření hFE se společným kolektorovým obvodem a Rl jako základní odpor je možné deaktivovat (NO\_HFE\_C\_RL), aby určití testéři neukazovaly nadměrné výsledky, (hlášeno Obelix2007@EEVblog).
  - Možnost výstupu Zenerova napětí ve vysokém rozlišení (ZENER\_HIGH\_RES,



(navrhl Andbro@EEVblog).

- OneWire\_Probes () byl vylepšen, aby se minimalizovala detekce chyb.
- Aktualizované ruské texty (díky indman@EEVblog).
- Aktualizované španělské texty (díky pepe10000@EEVblog).

### 9.11. v1.37m 2019-09

- Opravena chyba v DS18B20\_Tool(), když je povoleno ONEWIRE\_IO\_PIN (hlášeno bm-magic).
- Opraven problém se zobrazením chybové zprávy Watchdog na barevných displejích.
- Nová funkce: Počítadlo událostí (HW\_EVENT\_COUNTER). (Navrženo od bm-magic).
- Jednoduchý čítač frekvence nyní používá TestKey() pro vstup uživatele. Nyní dvojitý stisk tlačítka (dříve jedno stisknutí tlačítka).
- možnost zobrazit inverzní hFE hodnotu tranzistorů SW\_REVERSE\_HFE. (Návrh od towe96@EEVblog).
- Příkazy pro dálkové ovládání rozšířeno o příkaz "h\_FE\_r".
- Nastavení bitclocku (BITCLOCK) pro avrdude v Makefile. (Návrh bm-magic).
- Problém s detekcí TRIAC v případě příliš vysokého I\_GT ve Q3 nebo příliš vysokým I\_H opraven. (Problém I\_GT hlásil petroid).
- Texty Tester\_str, PWM\_str, Hertz\_str a CTR\_str byly přesunuty do jazyka specifických souborů záhlaví. (Doporučeno indman@EEVblog).
- Výstup hodnot frekvence (hertz) změněn na pevný řetězec (dříve "H" jako jednotka pro DisplayValue() plus další "z").
- Možnost usnadnění přístupu (UI\_KEY\_HINTS). V současné době pouze „Menu/Test“. (Návrh Carrascoso@EEVblog).
- Polské texty byly aktualizovány (C szpila@EEVblog).
- Ruské texty (díky indman@EEVblog).
- Španělské texty (díky pepe10000@EEVblog).

### 9.12. v1.36m 2019-05

Do ovladače ST7565R byla přidána volitelná znaková sada 6x8.

- Volitelná položka nabídky pro vypnutí testeru (SW\_POWER\_OFF).
- TestKey() a Zener\_Tool() byly přidány do monitorování baterie.
- Detekce dvou krátkých stisknutí testovacího tlačítka nainstalovaná do TestKey(), a dvojí funkčnost v několika funkcích odstraněna pro zmenšení velikosti firmwaru.
- Ovladač s ST7036 (4bitové paralelní & 4-vodičové SPI, netestované).
- Vlastní funkce pro napájení a monitorování baterie k lepší integraci s dalšími funkcemi.
- Ovladač s PCF8814 (třířádkový SPI,) (díky Mahmoud Laouar za testování).
- Ovladač s STE2007/HX1230 (3řádkový SPI).
- Opravena chyba ve funkci LCD\_Clear ovladače PCD8544. - Chybějící cyrilické písmo zadané v ovladači ST7565R (hlášeno Andrey@EEVblog).
- Aktualizace font\_8x16\_cyrillic\_vfp.h (díky Andrey@EEVblog).
- Problém s nesprávným znakem ve font\_HD44780\_cyr.h. byl vyřešen.

### 9.13. v1.35m 2019-02

- Posun kapacity lze použít namísto předchozího průměru pro všechny testovací piny, nyní také použít kompenzační odchylky specifické pro testovací pin \_\_ (CAP\_MULTIOFFSET).
- Opravena definice pinů pro ST7920 ve 4bitovém paralelním režimu v config\_644.h (hlášeno od jakeisprobably@EEVblog).
- Podpora 3-řádkového SPI zabudovaného v ovladači SSD1306.
- Ovladač SPI nyní může také odesílat 9bitová slova (pouze bitbang).
- Problém se zvyšující se odchylkou odporů mezi 7k5 a 19k5  $\Omega$  v CheckResistor() vyřešen (nahlásil Vitaliy).
- Alternativní smyčka zpoždění vestavěná do IR\_Send\_Pulse \$ (\$), které za SW\_IR\_TX\_ALTDELAY je aktivován (díky Vitaliy).
- Prošel konfigurační přepínač pro další IR protokoly SW\_IR\_EXTRA SW\_IR\_RX\_EXTRA pro přijímač / dekodér a SW\_IR\_TX\_EXTRA pro

IR vysílač vyměněn.

- Problém s chybějícím novým řádkem pro příkazy dálkového ovládání v `Display_NextLine()` odstraněn.
- Výstup pro SIRC změněn na `IR_Decode()`, aby byl blíže protokolu (návrh Vitaliy).
- Chyba v `IR_Send_Code()` pro SIRC-20 opraven. (hlášeno Vitaliy).
- Aktualizován `var_russian.h` (díky indman@EEVblog).
- Automatické vypnutí pro režim automatického pozastavení (`POWER_OFF_TIMEOUT`).
- Konfigurace pinu pro testovací tlačítko a ovládání napájení rozdělena (`CONTROL_PORT` -> `POWER_PORT` a `BUTTON_PORT`).
- Několik malých vylepšení.

#### 9.14. v1.34m 2018-10

Zkouška svodového proudu pro kondenzátory.

- Výchozí hodnota pro `RH_OFFSET` byla změněna na  $350\Omega$ .
- Opraven problém s chybějící položkou nabídky pro pevný IR přijímací modul.
- polský text (díky Szpila).
- Ovladač pro výstup na terminál VT100.
- Podpora teplotního senzoru DS18B20.
- Ovladač pro sběrnici OneWire.

#### 9.15. v1.33m 2018-05

Opravená orientace symbolu TRIAC ve `symbols_32x32_hf.h`.

- příkazy pro dálkové ovládání pro automatizaci (přes sériové TTL rozhraní).
- Posun X&Y pro ovladač ST7735 lze nyní změnit.
- Volání nabídky zkratováním testovacích pinů je nyní možné (`UI_SHORT_CIRCUIT_width`).
- Eliminován problém s vybíjecím relé ve spojení s rotačním enkodérem.
- Přidán přepínač konfigurace pro vypnutí režimů spánku MCU.
- Příjem dat pro sériové rozhraní TTL (Bit-Bang & Hardware USART).
- Opravené chyby v sériovém textovém výstupu a sériový výstup pro výsledky testů opto spojky.
- Dánský text (od glennndk@mikrocontroller.net).
- Nastavení korekčních faktorů pro kondenzátory.

#### 9.16. v1.32m 2018-02

Výstup nalezených komponent dodatečně přes sériové rozhraní.

- Serial TTL Interface Driver (Hardware & Bit-Bang).
- Aktualizováno z `var_russian.h` (díky indmanu@EEVblog).
- Podpora kompenzací X&Y v ovladači ST7735.
- Nastavení monitorování baterie se změnilo. Doplněno o spínač pro vypnutí monitorování baterie a pro nesledování externího napájení.
- Přepínač konfigurace pro výběr alternativního provozního režimu, při spuštění (`UI_AUTOHOLD`).
- Filtr germaniových tranzistorů s vysokým únikem v detekční funkci pro FET typu chudoby vylepšen.
- Přidáno grafické pinout v ovladači pro PCD8544. Opravené chyby ve funkci `LCD_CharPos()` pro otočný výstup s ovladačem PCD8544.
- Funkce grafického pinoutu byly vylepšeny a částečně přesunuty do `display.c`. V případě potřeby oddělte výstup od pinu.
- Indikátor při použití externí reference napětí (hodnoty čtení).
- Vylepšený IR dekodér a vestavěné volitelné protokoly.
- Další protokoly pro IR dálkové ovládání.

#### 9.17. v1.31m 2017-12

IR dálkové ovládání (vysílač).

- Podpora pevného výstupu signálu přes OC1B, pokud OC1B není pro testovací odpor používán testovacím pinem # 2.
- Nastavení sledování baterie se změnilo vzhledem k podpoře jiných možností napájení.

- Ovladač pro OLED moduly založené na SSD1306.
- Podpora barev pro výběr položek nebo parametrů nabídky.
- Ovladač pro moduly LCD založené na ILI9163.
- Opraven problém v generátoru obdélníků.
- LCD ovladač pro PCD8544 rozšířený o 180° otočený výstup.
- Chyba úprav v servo\_Check() opravena.

### 9.18. v1.30m 2017-10

Možnost čárky místo tečky pro desetinná místa.

- Podpora rozšířeného čítače kmitočtů se vstupní vyrovnávací pamětí, LF a RF krystalový oscilátor.
- Malá vylepšení jednoduchého čítače kmitočtů.
- Problém s časem brány v čítači kmitočtů pro kmitočty pod 10 kHz v taktu MCU 20MHz eliminován.
- měření ESR upravené pro kratší nabíjecí impulzy, tzn. ESR lze nyní použít pro kondenzátory od 10nF. Kdo dává přednost staré metodě měření, může tuto alternativu aktivovat.
- Opravena chyba v detekci zkratu testovacích pinů.
- LCD ovladač pro ST7920 rozšířený o 180° otočený výstup.

### 9.19. v1.29m 2017-07

Podpora dotykových obrazovek a ovladačů pro ADS7843 kompatibilní Controller.

- opravená chyba v nastavení kontrastu pro PCD8544.
- Opravená hloupá chyba v CheckSum().
- Ovladač pro ST7920 založený na 64x128 pixelech LCD modulech.
- Vylepšený SmallResistor() a zlepšená logika detekce v CheckResistor(), aby hodně malé odpory ve spojení s kontaktními odpory testovacích kabelů byly lépe rozpoznávány.
- Řídicí logika a prahová hodnota pro tranzistory Darlingston v Get\_hFE\_C() změněna, k odstranění problému s některými typy NPN.
- Centrální ovladač SPI. Ovladač a konfigurace LCD modulů odpovídajícím způsobem upraveny.
- Italský text od Gina\_09@EEVblog.
- podpora pro HD44780 s cyrilickou znakovou sadou od hapless@EEVblog.

### 9.20. v1.28m 2017-04

Více méně kláves jako alternativa k rotačnímu kodéru (HW\_INCDEC\_KEYS).

- Obnovení standardní frekvence v přidaném generátoru obdélníků.
- Další vylepšení detekce rotační rychlosti kodérů (ENCODER\_STEPS). Změny funkcí, které používají rotační rychlost.
- Obnovení výchozích hodnot v přidaném alternativním generátoru PWM.
- Ruský text od indman @ EEVblog (znaková sada 8x16 pouze vodorovně vyrovnána)
- Pevná podpora filmového kondenzátoru pro automatické vyvážení Ofset napětí.
- Potenciální chyba v V\_ref offset ve SmallCap() odstraněna.
- Možnost konfigurace pro LCD moduly s displejem ST7735 začínat s prázdným displayem opravena.(Žádné náhodné body).

### 9.21. v1.27m 2017-02

GetLeakageCurrent() rozšířený o měření vysokého proudu pro CLD. (Díky texaspyro@EEVblog) za několik testovacích diod.

- Opravená chyba v MilliSleep().
- Odstraněn problém s velkou indukčností při detekci diod.
- Kompenzace pro měření indukčnosti v rozsahu mH.
- Podpora pro LCD adaptér založený na PCF8574 v ovladači pro HD44780.
- ovladač pro bit-bang a hardware I2C.
- Zpracování chyb Variabilní Pinout pro HD44780 LCD moduly eliminovány.
- Barevný pinout pro více funkcí menu.
- Modeling Servo Checking.

- Alternativní generátor PWM s proměnnou frekvencí a šířkou impulzu. Vyžaduje Rotační kodér a větší displej.
- Výstup R\_DS pro MOSFET a V\_f vnitřní diody.
- Podpora pevného modulu IR přijímače v IR detektoru/dekodéru.
- Edice byla odstraněna z názvu, protože verze Classic je nyní zastaralá.

## 9.22. v1.26m 2016-12

Nainstalovaná kompenzace na měření indukčnosti (vyžaduje další práci).

- Přizpůsobený FrequencyCounter() pro podporu ATmega 324/644/1284.
- Vyřešený problém v logice měření indukčnosti. (Poznámka od indman @ EEVblog.)
- Chyba při zpracování referencí napětí pro ATmega 324/644/1284 vyřešena.
- Detekce otáček rotačních kodérů vylepšena pro lehčí ovládání různých hodnot impuls/krok nebo impuls/aretači.
- Všechny ovladače pro LCD moduly založené na SPI rozšířené o hardwarové SPI.

## 9.23. v1.25m 2016-09

Spousta změn na podporu ATmega 324/644/1284.

- Správa testovacích rezistorů přepnuta na piny proměnných portů.
- Možnost softwaru pro barevné kódování zkušebních pinů.
- Centralizovaná správa barev.
- Soubor se seznamem nastavení pro různé verze testeru nebo klony.
- Opraven menší problém se symboly 24x24-VP v config.h. (Poznámka od lordstein@EEVblog a hapless@EEVblog).

## 9.24. v1.24m 2016-08

Měření svodového proudu kondenzátorů větších než  $4,7\mu\text{F}$ .

- Detekce typu bipolárních tranzistorů s diodou na stejném substrátu.
- Měření svodového proudu pro proudy rozšířené do rozsahu nA. Pro diody a bipolární tranzistory jsou indikovány svodové proudy nad 50nA.
- Zobrazení volnoběžných diod v tranzistoru nyní kontroluje správnost diody. (piny a polarita).
- Opravena chyba v zobrazení volnoběžných diod v bipolárních tranzistorech.
- Napsaná funkce pro vyhledávání konkrétní diody a několik dalších funkcí odpovídajícím způsobem upraveny.
- Detekce diod se zlepšila dokonce na germaniové diody s velmi nízkým V\_f při nízkých proudech.
- Problém s LCD\_ClearLine(0) pro ILI9341 a ST7735 vyřešen.
- Zlepšená detekce ochuzovacích FET. Germanium tranzistory s vysokou netěsností jsou odfiltrovány. Také FET s nízkým I\_DSS rozpoznány. Měření I\_DSS.

## 9.25. v1.23m 2016-07

Podpora LCD modulů kompatibilních s PCD8544 a ST7735. (Díky hansibull@EEVblog pro zobrazení PCD8544.)

- wait.s přidán 20MHz MCU takt.
- MeasureESR() nyní podporuje jiné taktové frekvence ADC než 125 kHz.
- Detekce PUT (programovatelný unijunkční tranzistor) a vestavěný UJT (Unijunction Transistor). (Díky edavid@EEVblog) za několik UJT k testování).
- Drobné optimalizace pro ILI9341 a ST7565R.
- Opětně opraven problém se znaky většími 8x8 pro ST7565R.
- Pin/RES port pro ILI9341 byl ignorován. Zpoždění hardwarových resetů opraveno.
- Podpora jednotlivých datových linek pro HD44780 LCD moduly.
- Uživatelem definovatelný dělič napětí pro napětí baterie.
- Výdej I\_f pro optočleny rozšířen.
- Testovací piny z menu ESR byly změněny na 1-3 tak, aby byly kompatibilní s k-firmwarem.
- Pro MCU specifické globální nastavení byly vytvořeny jejich vlastní záhlavové soubory.
- Několik malých úprav na podporu ATmega664/1284.
- Aktualizovány české texty. (Díky Kapa).

### 9.26. v1.22m 2016-03

Test opto-spojky s výstupem V<sub>f</sub> LED, CTR a t<sub>on</sub> nebo t<sub>off</sub> časů (typy s tranzistorovým výstupem). (Díky všem\_repair@EEVblog za opto-členy k testování.)

### 9.27. v1.21m 2016-01

Licence na základě licence EUPL V.1.1

- Optimalizace načítání a ukládání hodnot sladění a podpora vytvoření dvou profilů úprav.
- IR detektor rozšířen o RC-6. Problém s tlačítkem při předčasném odstranění IR přijímacího modulu eliminován.
- Konfigurační spínač pro vypnutí Vs odporu pro omezení proudu u 5V IR přijímacích modulů.

### 9.28. v1.20m 2015-12

Implementovaná funkce pro detekci a dekódování IR dálkových ovladačů.

Vyžaduje modul přijímače TSOP IR.

- Změněno MainMenu(), aby se snížilo využití RAM.

### 9.29. v1.19m 2015-11

Grafický pinout pro 3pinové polovodiče. Zobrazuje ikonu plus zkušební piny.

- Nainstalovaná podpora barev.
- Přímý výstup počtu diod v ShowDiode() při více než 3 diodách byly nalezeny (již ne pomocí Show\_Fail()). (Oznámení od hapless@EEVblog)
- LCD\_ClearLine() ve všech ovladačích LCD modulů tak rozšířený, aby šlo smazat pouze zbytek řádku k urychlení grafických LCD. Cílem je nejprve vypsat text a poté odstranit zbytek řádku namísto prvního odstranění celého řádku a poté vydat text.
- Napsaný ovladač pro LCD moduly založené na ILI9341/ILI9342. (Díky Overtuner@EEVblog-fórum za dva LCD moduly pro testování.)
- Opraven problém s  $\mu$ /micro značkou v souborech fontů. - Opravená chyba se znaky většími než 8x8 na LCD\_Char() pro ST7565R.
- České texty aktualizovány (díky Kapa).
- Odstraněna malá chyba v MenuTool(), při skoku z posledního na první bod.

### 9.30. v1.18m 2015-07

Vylepšené MenuTool(), takže bude aktualizován pouze změněný seznam. Jinak bude aktualizován pouze indikátor výběru.

- Opravena chyba ve správě proměnných v config.h.
- Možnost resetování na výchozí hodnoty firmwaru při zapnutí.
- Funkce pro ukládání/čtení optimalizovaných hodnot.
- Ovladač pro grafické moduly ST7565R.
- Jednoduché prostředí navržené pro začlenění více ovladačů LCD. Obecné funkce zobrazení byly přesunuty do display.c. Každý ovladač získá svůj vlastní zdrojový a hlavičkový soubor. Starý ovladač pro HD44780 byl přizpůsoben novému prostředí.
- Uživatelské rozhraní pro flexibilní manipulaci s víceřádkovými LCD moduly přestavěn.
- Byly odstraněny závislosti zdroje na ATmega168 (příliš malé;).
- Operační logika optimalizovaná v MenuTool().
- Byla spuštěna nová verze firmwaru, která podporuje také grafické LCD moduly. Tato verze se nazývá „Trendy Edition“. Stará verze firmwaru se nyní nazývá „Classic Edition“.

### 9.31. v1.17m 2015-02

Vylepšení CheckDiode(). Při měření odporu je na zřeteli naměřené V<sub>cc</sub>.

Kromě toho, problém detekce rezistorů kolem 2k s volitelným převodníkem DC-DC (HW\_ZENER) vyřešen.

- Opravené nesprávné komentáře.
- Vyčistí celočíselných typů dat.

### 9.32. v1.16m 2014-09

Test rotačních kodérů.

- Několik maličkostí v MeasureInductance() zlepšeno k zvýšení přesnosti.
- ShowAdjust() pro zobrazení absolutních hodnot Vcc a interní referenční hodnoty napětí (návrh Vlastimila Valoucha).
- Několik malých vylepšení.

### 9.33. v1.15m 2014-09

Rozšíření TestKey() pro detekci dynamické rychlosti otáčení z volitelným rotačním kódérem.

- Realizován generátor signálu s proměnnou frekvencí.
- MeasureInductance() pro vrácení času v ns se změnil a výpočet upraven na MeasureInductor(). (Díky Vlastimilu Valouchovi).

### 9.34. v1.14m 2014-08

Uživatelské rozhraní pro rotační kódér.

- Upozornění kompilátoru týkající se R\_Pin2 v ShowDiode() bylo opraveno (díky Milan Petko). Odporů mezi 1,5k a 3Ω byly detekovány jako dvojité diody. Tolerance detekce odporu v CheckDiode() upravené (díky nessatse).
- ShortCircuit() upraveno tak, aby u vytvoření úmyslného zkratu, bylo možné zrušení.
- Vestavěný čítač kmitočtů (volitelný hardware).

### 9.35. v1.13m 2014-07

České texty (díky Kapa).

- Přímé měření ESR a PWM generátor vdají použité testovací piny.
- Optimalizované zpracování příkazů předkompilátoru pro možnosti.
- Podpora rotačních enkodérů pro provoz (možnost hardware).

### 9.36. v1.12m 2014-03

Problém s přehlasovanými znaky v německých textech vyřešen (díky Andreas Hoebel).

- měření ESR pro kondenzátory  $> 0,18\mu\text{F}$ .
- Výstup modulu LCD optimalizovaný pro ušetření několika bajtů ve flash.

### 9.37. v1.11m 2014-03

Vylepšená detekce triaků (G a MT1). Display ukazuje MT1 a MT2.

- Výstupní funkce pro přiřazení pinů změněna, kvůli čitelnosti, na formát „123 =“.
  - Optimalizováno více výstupních funkcí.
  - Test bipolárních tranzistorů vylepšen k rozpoznání tranzistorů s ochrannou diodou na stejném substrátu (vytvoří parazitický druhý transistor). Výstup označí tento speciální případ s „+“ po specifikaci typu.
  - Diodový výstup pro označení možného bipolárního tranzistoru s ochrannou diodou a báseň-emitor odporem. Ten je rozpoznán jako dvojitá dioda. Výstup tento speciální případ signalizuje.
  - Výstup bipolárních tranzistorů je rozšířen o zobrazení základny-emitoru odporu.
- Pokud je nalezen rezistor základního emitoru, bude výstup z hFE a V\_BE přeskočen, protože obě hodnoty nemohou být správné,
- Vylepšená detekce integrované diody u ochuzených FET při testu diod.
  - Problém zjištění odtoku a zdroje u ochuzených FET odstraněn..
  - Detekce symetrických odtoků a zdrojů u FET s vyčerpáním.
  - Vth je nyní pro FET P-kanálu negativní.
  - Měření V\_GT pro tyristory a triaky.
  - Kvůli rostoucí velikosti firmwaru existuje generátor PWM pouze pro ATMEGA328.

### 9.38. v1.10m 2013-10

Nainstalovaná podpora pro externí referenční napětí 2,5 V (volitelný hardware).

- Nainstalovaná podpora pro ochranné relé (vybití kondenzátoru) (Volba hardware).
- Vítací a vypínací text, změněn k usnadnění detekce příliš nízkého napájecího napětí a poklesu napětí přes DC-DC převodník, při zapnutí.
- Zenerova diodová zkouška implementována (hardwarová volba).
- V hlavní nabídce integrovaná možnost konec, při omylném vyvolání nabídky.

- Podpora 16MHz MCU taktu.

### 9.39. v1.09m 2013-07

Integrovaná detekce IGBT.

- Přidáno ověření měření MOSFETů.
- HFE měření pro bipolární tranzistory zohledňuje svodový proud během měření v emitorovém obvodu.
- U MOSFETů je zobrazen směr integrované diody.
- Problém s kroucenými vypouštěcími a zdrojovými piny u vylepšených MOSFETů vyřešen.
- Řešení problémů některých IDE s Makefile. Důležité hodnoty nebo nastavení lze také nastavit v config.h.

### 9.40. v1.08m 2013-07

Protože SmallResistor() neposkytuje pro některé induktory správné vyrovnaní hodnoty odporu, byla funkce CheckResistor() zozšířena o detekci problémových případů pro zachování hodnot standardního měření.

- nainstalované měření indukčnosti (pouze pro ATmega328/P)
- Drobné vylepšení zobrazování diod a bipolárních tranzistorů.
- Vestavěné měření svodového proudu.
- Problém s germaniovými tranzistory s vysokým svodovým proudem byl vyřešen. Tyto Byli detekovány jako P-kanál JFET.
- Několik funkcí přejmenováno a komentáře přidány nebo přeformulovány.

### 9.41. v1.07m 2013-06

Diodový výstup optimalizovaný na Vf zobrazení pro vestavěné nízké proudy.

- Vylepšena detekce diod. Kondenzátory a rezistory budou jasně lépe vyloučeny. Součástí je detekce kondenzátoru, detekovaná dioda je přeskočena, aby se proces vyhledávání zkrátil.
- Opravena chyba přetečení pole v CheckResistor().
- Logika zobrazení kurzoru se zlepšila na přítomnost dalších informací nebo pro zobrazení obnoveného vyhledávání součástí.
- Provoz PWM generátoru se zlepšil, aby se zabránilo neúmyslnému vypnutí (nyní je třeba dvou krátkých stisků kláves).
- Vestavěná obecná funkce nabídky a všechna menu v ní integrována (rozvržení změněno!).
- TestKey() nyní vytváří pěkný blikající kurzor.

### 9.42. v1.06m 2013-03

Několik malých vylepšení a trochu uklizení.

- TestKey() funkce rozšířena, takže je uživatel informován o očekávaný vstup.
  - TestKey() - vylepšená funkce pro krátké stisknutí klávesy.
  - Implementován PWM generátor pro generování signálů modulovaných šířkou impulzu s různou frekvencí a libovolným pracovním cyklem.
  - Implementace funkce spánku ke snížení spotřeby energie testeru.
- Průměrná spotřeba energie se tak sníží na asi polovinu (vyloučeno podsvícení).
- Vylepšena funkce koncového načítání. Pokud se vyprázdnění nezdaří, jsou zobrazeny ovlivněné piny a zbytkové napětí. To by mělo pomoci k zjištění příliš nízké hodnoty pro CAP\_DISCHARGED.
  - možnost nastavení typů chyb.

### 9.43. v1.05m 2012-11

- LargeCap\_table[] a SmallCap\_table[] byly přesunuty z EEPROM do Flash, k snížení EEPROM požadavek. Firmware s německými texty spotřebuje více než 512 bajtů ATmega168.

### 9.44. v1.04m 2012-11

Jednoduchá logika zabudovaná do výstupu diody, takže u antiparalelních diod odpadá měření kapacity.

#### **9.45. v1.03m 2012-11**

Detekční problém výkonných diod vyřešen. Diody s únikem byly zjištěny jako odpor.

- Opravena varování kompilátoru týkající se neinicializovaných proměnných.
- zvětší firmware o 44 bajtů :-)

#### **9.46. v1.02m 2012-11**

Horní mez pro odpor zkušebních kabelů 1,00  $\Omega$  v automatickém ladění.

- Funkce autotestu a ladění provedou test na zkrat a poskytují zpětnou vazbu.
- Hlavní nabídka poskytuje zpětnou vazbu o úspěchu/neúspěchu vybrané akce.

#### **9.47. v1.01m 2012-10**

Kontrolní součet uložených hodnot seřízení plus ověření.

- Měřicí funkce pro malé odpory (rozlišení: 0,01  $\Omega$ ).
- Automatické ladění rozšířené o nulový offset pro odpor zkušebních vodičů.
- CheckResistor() provádí další měření malých odporů (<10  $\Omega$ ).
- Přidána funkce pro porovnání škálovaných hodnot.
- Několik funkcí přizpůsobených proměnnému škálování hodnot.

#### **9.48. v1.00m 2012-09**

- Jednoduché menu pro výběr autotestu,
- samoladění,
- Uložení hodnot nastavení do EEPROM
- Zobrazení hodnot nastavení.
- hFE se změnil ze 16 na 32 bitů (žádný 65kB limit).

#### **9.49. v0.99m 2012-09**

- První publikovaná verze založená na Karl-Heinze.



---

## *Literatura*

---

- [1] <http://www.mikrocontroller.net/articles/AVR-Transistortester>  
*Online Dokumentace Transistortesteru*, Online Article, 2009-2011
- [2] Markus Frejek <http://www.mikrocontroller.net/topic/131804>  
*Forum od Markuse Frejka*, 2009.
- [3] [http://www.mikrocontroller.net/articles/AVR\\_Transistortester](http://www.mikrocontroller.net/articles/AVR_Transistortester)  
*Krátký popis vlastností TransistorTesteru od Karla-Heinze K.*, Online Article, 2012
- [4] <http://www.mikrocontroller.net/topic/248078>  
*Thread od Karl-Heinz K.*, 2012
- [5] <https://github.com/madires/Transistortester-Warehouse>  
*Kompletní softwarová sbírka i s dokumentací v různých jazycích*
- [6] <https://github.com/kubi48/TransistorTester-source/tree/master/Markus>  
*Kompletní softwarová sbírka*
- [7] <https://github.com/Mikrocontroller-net/transistortester/tree/master/Software/Markus>  
*Kompletní softwarová sbírka*
- [8] [madires@theca-tabellaria.de](mailto:madires@theca-tabellaria.de)  
*Autor Mail*
- [9] <http://www.mikrocontroller.net/articles/AVRDUDE>  
*Online Dokumentace avrdude IDE*
- [10] <https://svetelektro.com/phpbb/?w3=dmld3RvcGljLnBocD9mPTE4JnQ9MzAyODU=>  
*Sme najnavštevovanejší portál zo zameraním na elektroniku na Slovensku! od 2006*
- [11] <https://www.mikrocontroller.net/topic/248078>  
*Hlavní řeč je němčina, anglicky je ale také ok.*
- [12] [https://www.eevblog.com/forum/testgear/\\$20-lcr-esr-transistor-checker-project/](https://www.eevblog.com/forum/testgear/$20-lcr-esr-transistor-checker-project/)  
*Jen anglicky.*
- [13] <https://vrtp.ru/index.php?showtopic=16451> - Ruské Forum  
*Informace o různých klonech, jako jsou obrázky, schémata a firmware*
- [14] <https://disk.yandex.ru/d/yW8xa5NJgUo5z> - Ruská stránka (od indman@EEVblog)  
*Informace o různých klonech, jako jsou obrázky, schémata a firmware*
- [15] [https://drive.google.com/file/d/1-IJA8uTcsCA\\_6SYHEuMydjfS2vNgmwdH/edit](https://drive.google.com/file/d/1-IJA8uTcsCA_6SYHEuMydjfS2vNgmwdH/edit)  
*Návod a další informace o WinAVR (od indman@EEVblog)*
- [16] <https://github.com/kubi48/TransistorTester-source>  
*Aktuální k-Firmware*
- [17] <https://github.com/kubi48/TransistorTester-documentation>  
*Aktuální dokumentace pro k-Firmware*