

**Тестер ЭРЭ с AVR микроконтроллером
и минимумом дополнительных элементов**

Component Tester firmware

(c) 2012-2021 Markus Reschke (madires@theca-tabellaria.de)

Русский перевод (indman)

01 августа 2021 г.

Содержание

| | |
|--|----|
| Описание | 3 |
| Рекомендации по безопасности | 4 |
| Лицензия | 4 |
| <i>Дополнительная правовая оговорка</i> | 4 |
| Отличия версий ПО | 4 |
| Исходный код | 5 |
| Поддерживаемое оборудование | 5 |
| Создание прошивки | 6 |
| Шины и интерфейсы | 8 |
| <i>I2C/SPI</i> | 8 |
| <i>TTL(RS232)</i> | 8 |
| <i>1-Wire</i> | 9 |
| Дисплей | 9 |
| HD44780 | 10 |
| ILI9163 | 10 |
| ILI9341/ILI9342 | 11 |
| ILI9481 | 12 |
| ILI9486 | 13 |
| ILI9488 | 13 |
| PCD8544 | 13 |
| PCF8814 | 14 |
| SH1106 | 14 |
| SSD1306 | 15 |
| ST7036 | 15 |
| ST7565 | 16 |
| ST7735 | 16 |
| ST7920 | 17 |
| STE2007/HX1230 | 17 |
| VT100 терминал | 17 |
| Кнопки тестирования и управления | 18 |
| <i>Кнопка "Тест"</i> | 18 |
| <i>Поворотный энкодер (аппаратная опция)</i> | 18 |
| <i>Кнопки управления +/- (альтернатива энкодеру)</i> | 18 |
| <i>Сенсорный экран</i> | 18 |
| Пользовательский интерфейс | 19 |
| Соединение с компьютером | 19 |
| <i>Последовательный выход</i> | 20 |
| <i>Автоматизация</i> | 20 |
| VT100 | 20 |
| Включение | 20 |
| Тестирование | 21 |
| Контроль источника питания | 21 |
| Выключение | 21 |
| Меню | 21 |
| <i>ШИМ генератор</i> | 22 |
| <i>Генератор прямоугольного сигнала</i> | 22 |
| <i>Тестирование стабилитронов (аппаратная опция)</i> | 23 |
| <i>Внутрисхемное измерение ESR</i> | 23 |
| <i>Проверка конденсатора на ток утечки</i> | 23 |

| | |
|--|----|
| R/C/L монитор | 23 |
| LC-измеритель (аппаратная опция) | 24 |
| Частотомер (аппаратная опция) | 25 |
| Базовый вариант частотомера | 25 |
| Расширенный вариант частотомера | 25 |
| Счетчик событий (аппаратная опция) | 25 |
| Выход триггера | 26 |
| Тест поворотного энкодера | 26 |
| Контрастность дисплея | 26 |
| ИК-приёмник /Декодер..... | 26 |
| ИК-передатчик | 28 |
| Тестирование оптронов | 29 |
| Тестирование сервоприводов RC | 30 |
| Сканирование шины I-Wire | 31 |
| Тестирование температурного датчика DS18B20 | 31 |
| Тестирование датчиков DHTxx | 31 |
| Самотестирование | 32 |
| Автокорректировка | 32 |
| Сохранение/Загрузка | 33 |
| Информация о корректировке | 33 |
| Выключение | 33 |
| Выход | 33 |
| Резисторы | 33 |
| Конденсаторы | 34 |
| Индуктивности | 35 |
| Разряд компонентов | 35 |
| ADC супердискретизация | 35 |
| Отображение результатов на дисплее | 36 |
| Дополнительные подсказки | 36 |
| BJTs | 36 |
| Симисторы | 37 |
| Диоды CLDs | 37 |
| Неподдерживаемые компоненты | 37 |
| Известные проблемы и способы их решения | 37 |
| Поддержка | 38 |
| Журнал изменений | 38 |
| Команды управления | 38 |
| Полезные ресурсы | 43 |
| Ссылки на первоисточники | 43 |

ОПИСАНИЕ

Тестер компонентов создан на основе проекта Markus Frejek [\[1 и 2\]](#) и его преемника Karl-Heinz Kübbeler [\[3 и 4\]](#). Это альтернативное программное обеспечение для текущей схемы Karl-Heinz с некоторыми изменениями в пользовательском интерфейсе и методами, используемыми для тестирования и измерения. Karl-Heinz выпускает официальные релизы, поддерживающие также более старые процессоры ATmega, данная же версия работает на процессорах ATmega с объемом 32кБ памяти и более.

Подсказка: запускать режим автокорректировки при первом запуске, а так же, если вы сделали какую-либо модификацию, обновление прошивки или применили другие тестовые контакты (зажимы).

РЕКОМЕНДАЦИИ ПО БЕЗОПАСНОСТИ

Тестер компонентов - **это не профессиональный цифровой измеритель!**
Это простой тестер для измерения некоторых деталей. Используйте его для демонтированных электронных компонентов. Тестовые контакты не имеют защиты и не допускают работу на напряжениях выше, чем 5В.

Не используйте тестер для цепей под высоким напряжением!

Если вы тестируете конденсатор, обязательно разрядите его прежде, чем подключить! Это не просто безопасность, ваша жизнь может находиться в опасности, если вы соедините тестовые контакты с цепью под напряжением или источником питания, особенно в электросети!

ЛИЦЕНЗИЯ

Автор исходного кода не предоставил информации о лицензии, которая ограничивает использование микропрограммного обеспечения. Он утверждал, что это открытый источник и любой коммерческий пользователь должен связаться с ним. К сожалению, мы (Karl-Heinz и я) не нашли способа связаться с ним. Поэтому я выбрал лицензию с открытым исходным кодом 01.01.2016. Автор может связаться с нами, чтобы изъяснить свои пожелания по этому вопросу. Так как исходный код этой прошивки был обработан и переписан с массой новых функций, я думаю, что такой подход оправдан. Лицензия зарегистрирована под EUPL V.1.1.

Дополнительная правовая оговорка

Название продукта или название компаний – возможно торговые марки соответствующих владельцев.

ОТЛИЧИЯ ВЕРСИЙ ПО

Karl-Heinz написал действительно полную документацию на тестер, очень рекомендую прочитать её. Я объясню вам существенные отличия от его версии:

- - пользовательский интерфейс, никаких проблем;
- - поддержка дисплеев с сенсорным управлением;
- - команды управления тестером;
- - функция разрядки подключенного компонента;
- - измерение сопротивления:
+ специальный метод тестирования сопротивлений менее 10 Ом (вместо метода проверки ESR);
- - измерение ёмкости:
+ измерение ёмкости от 5pF;
+ дополнительный метод тестирования для конденсаторов от 4.7µF до 47µF;
+ метод корректировки/компенсации;
- - SamplingADC для измерений очень малой емкости или индуктивности не использовался;
- - диоды:
+ автоопределение;
- - биполярные плоскостные транзисторы (BJTs):
+ V_f определяется для более подходящего (виртуального) I_b на основе hFE;
+ обнаружение германиевых биполярных плоскостных транзисторов с высоким током утечки (JFETs);
+ обнаружение VJT с барьером Шоттки;
- - обнаружение JFETs с очень низким IDSS;
- - тиристоры (симисторы):
+ обнаружение MT1 и MT2;

- - приёмник ИК RC и декодер;
- - ИК RC передатчик;
- - проверка оптронов;
- - тестирование сервоприводов RC;
- - работа схемы возможна на тактовой частоте процессора 20 МГц;
- - 1-Wire (DS18B20);
- - ДНхх датчики;
- - счетчик импульсов (событий);
- - структурированный исходный код;
- - + ещё изменения, о которых я не могу вспомнить сейчас.

Более подробное описание функций в разделах ниже.

ИСХОДНЫЙ КОД

Начальная прошивка m-версии была основана на исходном коде Карла-Хайнца. Далее была сделана глубокая оптимизация, добавлено больше комментариев и переменных, реструктурированы функции. Большие функции разделены на несколько меньших частей. Вскоре моя прошивка стала независимой версией. Например, были добавлены простые каркасы для поддержки различных дисплеев и интерфейсных шин. Я надеюсь, что сейчас код легко читается и поддерживается.

Вы можете загрузить актуальное программное обеспечение с этих сайтов:

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

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

(<https://github.com/Mikrocontroller-net/transistortester/tree/master/Software/Markus>)

ПОДДЕРЖИВАЕМОЕ ОБОРУДОВАНИЕ

Прошивка работает на всех устройствах, которые совместимы со стандартной схемой, размещенной в документации Карла-Хайнца и которые используют один из следующих MCU:

- ATmega 328
- ATmega 324/644/1284
- ATmega 640/1280/2560

Вы можете настроить назначение портов процессора при необходимости. Дисплей может быть символьным или графическим (монохромный или цветной). Пожалуйста, посмотрите раздел «Дисплеи», где указаны поддерживаемые контроллеры. Доступны следующие аппаратные опции:

пользовательский интерфейс:

- поворотный энкодер;
- дополнительные кнопки (увеличение/уменьшение);
- сенсорный экран;
- последовательный интерфейс (TTL, RS232, USB-посл. адаптер);

дополнительные возможности:

- внешний 2.5-вольтный источник опорного напряжения;
- аппаратная калибровочная емкость;
- защитное реле для разрядки конденсаторов;

дополнительные тесты и измерения:

- проверка стабилитронов / измерение внешнего напряжения <50В;
- простой частотомер;

- расширенный вариант частотомера с прескалером и кварцевыми генераторами для низких и высоких частот;
- аппаратный ИК приёмник.

СОЗДАНИЕ ПРОШИВКИ

Сначала отредактируйте `makefile`, уточните вашу модель процессора, частоту, тип резонатора и настройки программатора. Остальные настройки перемещены в основной файл `config.h`, а конфигурация процессора в файл `config<mcu>.h`. Файл "Clones" содержит параметры настройки для различных версий/клонов тестера. Если у вас есть настройки для тестера, которого нет в списке, отправьте их по электронной почте автору, чтобы помочь другим пользователям.

В `config.h` выбирают аппаратные и программные опции, язык интерфейса и изменяют любые значения по умолчанию при необходимости. Все настройки и значения прокомментированы в файле, поэтому не буду обсуждать их здесь подробно.

Аппаратные опции:

- поворотный энкодер или кнопки управления;
- 2.5-вольтовый источник опорного напряжения;
- защитное реле (разрядка конденсаторов);
- измерение напряжения стабилизации для стабилитронов;
- частотомер (базовый и расширенный варианты);
- счетчик событий;
- LC-измеритель;
- приёмник/декодер ИК для пультов дистанционного управления (аппаратный модуль приемника ИК);
- поддержка аппаратного калибровочного конденсатора (только с ATmega644/1284) для корректировки смещений напряжения;
- SPI шина (программная и аппаратная поддержка);
- I2C шина (программная и аппаратная поддержка);
- TTL последовательный интерфейс(аппаратная и программная поддержка);
- 1-Wire шина (программная).

Внешний 2.5-вольтовый источник опорного напряжения необходимо использовать только, если он в 10 раз более точный, чем стабилизатор напряжения. Иначе это приведёт к худшим результатам. Если вы используете микросхему MCP1702 с типичным допуском 0.4% как стабилизатор напряжения, внешний ИОН в схеме не нужен!

Опции программного обеспечения:

- ШИМ генератор (2 варианта);
- измерение индуктивности;
- измерение ESR и внутрисхемное измерение ESR;
- проверка поворотных энкодеров;
- генератор прямоугольного сигнала (требуется доп. кнопки);
- приемник/декодер ИК для пультов дистанционного управления (модуль приемника ИК должен быть подключён к схеме);
- ИК RC передатчик (ИК светодиод с транзистором управления);
- проверка оптронов;
- проверка сервоприводов (требуется доп. кнопки и дисплей на более чем 2 строки);
- обнаружение UJTs;
- проверка тока утечки конденсаторов;
- тест температурного датчика DS18B20;

- цветное отображение тестовых контактов (для цветного графического дисплея);
- вывод информации о найденных компонентах через последовательный TTL(RS-232), например, на PC;
- команды управления для автоматизации с использованием последовательного TTL;
- измерение обратного hFE для BJTs;
- DHT11/22 датчик температуры и влажности.

Выберите необходимые опции внимательно, чтобы они соответствовали вашим потребностям, а также ресурсам процессора, т.е. размеру RAM, EEPROM и флэш-памяти. Если скомпилированная прошивка превышает размер флэш-памяти процессора, попробуйте отключить некоторые опции, в которых вы не нуждаетесь.

Доступные языки интерфейса:

- Английский (по умолчанию);
- Немецкий;
- Чешский (поддержка Кара) - шрифт на базе кодировки ISO8859-1;
- Чешский 2 (поддержка Bohu) - шрифт на базе кодировки ISO8859-2;
- Датский (поддержка glennnk@mikrocontroller.net);
- Итальянский (поддержка Gino_09@EEVblog);
- Польский (поддержка Szpila);
- Румынский (поддержка Dumidan@EEVblog);
- Испанский (поддержка pere10000@EEVblog);
- Датский язык (поддержка glennnk@mikrocontroller.net) незначительные изменения в шрифте;
- Русский (поддержка indman@EEVblog) - шрифт с поддержкой кириллицы на базе кодировки Windows-1251;
- Русский 2 (поддержка hapless@EEVblog) - шрифт с поддержкой кириллицы на базе кодировки Windows-1251, альтернативный вариант.

Десятичная часть измерений обозначена точкой, но вы можете изменить её на запятую, если сделать соответствующую настройку. Определенные настройки процессора, назначения контактов и тип дисплея, редактируются в файле конфигурации - config<mcu>.h:

- ATmega328 - config_328.h
- ATmega324/644/1284 - config_644.h
- ATmega 640/1280/2560 - config_1280.h

Дисплей должен иметь, по крайней мере, 2 строки по 16 символов в каждой. Для графических дисплеев выбирают шрифт, который является достаточно компактным, чтобы соответствовать программным требованиям. После редактирования makefile, config.h или config<mcu>.h выполните команду 'make' для компиляции встроенного микропрограммного обеспечения. Это создаст два файла:

- ComponentTester.hex в формате Intel hex
- ComponentTester.eep EEPROM в формате Intel hex

Файл ComponentTester.hex необходимо записать во Flash память, а ComponentTester.eep - в область EEPROM. Данные EEPROM содержат два набора значений корректировки по умолчанию, тексты и таблицы. Если вы обновляете прошивку и хотите сохранить старые значения настроек в EEPROM, то можете включить опцию DATA_FLASH в config.h для переме-

щения текстов и таблиц во Flash память. В этом случае необходимо запрограммировать только Flash область, область данных EEPROM останется нетронутой.

Makefiles обеспечивает следующие дополнительные действия:

- clean - удаление все объектов и микропрограммных файлов;
- fuse - установка фьюзов для ATmega (через avrdude);
- upload - программирование кода и данных EEPROM(через avrdude);
- prog_fw - программирование только кода (через avrdude);
- prog_ee - программирование только данных EEPROM (через avrdude).

ШИНЫ И ИНТЕРФЕЙСЫ

I2C/SPI

Некоторым дисплеям и другим аппаратным средствам необходимы протоколы I2C или SPI для соединения с процессором. Поэтому прошивка включает драйверы для обоих протоколов. Программный и аппаратный режимы могут обеспечить назначения различных контактов для разных тестеров и поддержку драйверов шины. Программный режим может использовать любые выводы IO на одном и том же порту, в то время как аппаратный режим использует специальные контакты шины процессора. Недостаток программного режима - низкая скорость. Аппаратный режим намного быстрее. Вы можете легко определить различие в скорости при подключении жидкокристаллических цветных дисплеев. LCD модули с интерфейсом SPI могут управляться аппаратным SPI или с использованием программной эмуляции. Тестер на ATmega328 поддерживает только программную эмуляцию SPI из-за особенности схемы. ATmega324/644/1284 с большим количеством портов ввода-вывода позволяет использовать программную эмуляцию SPI а также намного более быстрый аппаратный SPI. Драйверы SPI или I2C включены в LCD модули и могут быть настроены в разделе конфигурации дисплея - config-<MCU>.h. Альтернативно, вы можете также включить I2C и SPI в config.h, настроить порты и конфигурацию в специальных разделах - config-<mcu>.h (смотрите I2C PORT или SPIPORT).

Если вы выберете программный SPI и включите режим чтения (SPI_RW), убедитесь, что установлен также SPI_PIN и SPI_MISO. См. раздел SPI в config- <MCU> .h для примера.

Последовательный TTL(RS-232)

Тестер может поддерживать вывод информации через последовательный интерфейс TTL. Если он используется для связи с PC, у вас должен быть также USB - TTL преобразователь или классический драйвер RS-232. Встроенное микропрограммное обеспечение использует аппаратный или программный UART. Последовательный интерфейс TTL включается и настраивается в файле config.h (смотрите секцию "Busses"), контакты порта определены в config <mcu>.h (смотрите SERIAL_PORT).

У программного UART есть недостаток - режим TX не имеет постоянно высокий уровень, если он не активен. Это происходит из-за способа, которым управляется вывод порта МК. Чтобы исправить недостаток, управление портом должно быть изменено, что вызовет увеличение объема встроенного микропрограммного обеспечения. Но эта проблема не доставляет неприятностей с большинством USB-TTL последовательных преобразователей. Если проблема всё же есть, вы можете добавить подтягивающий резистор (10-100k) на выход TX, чтобы поддерживать сигнал на высоком уровне в режиме ожидания.

Настройка по умолчанию для последовательного TTL 9600 8N1:
- 9600 бит/с, 8 бит данных, нечетный, 1 стоп бит, без контроля потока

1-Wire шина

Другая поддерживаемая шина – 1-Wire, которая может использовать любые щупы/тестовые контакты(ONEWIRE_PROBES) или специальный контакт ввода-вывода (ONEWIRE_IO_PIN). Драйвер разработан для стандартной частоты шины и устройств с внешним питанием.

Назначение контактов для подключения:

Тестовый контакт №1: Земля(Gnd)

Тестовый контакт №2: DQ (данные)

Тестовый контакт №3: Питание Vcc (ток ограничен резистором 680Ом)

Требуется внешний резистор подтяжки 4.7 кОм между контактами DQ и Vcc!

Для устройства, требующего подключения к шине только одного клиента, можно при желании узнать его код ПЗУ (ONEWIRE_READ_ROM). В случае CRC ошибки или если подключено несколько клиентов, на дисплее будет символ '-'. Если код ПЗУ равен нулю - есть проблема чтения. При удачном считывании первая часть кода ПЗУ показывает семейство устройства, вторая часть – его серийный номер.

ДИСПЛЕИ

На данный момент поддерживаются следующие контроллеры дисплея:

- HD44780 (символьный дисплей, 2-4 строки на 16-20 символов),
- ILI9163 (цветной графический дисплей 128x160),
- ILI9341/ILI9342 (цветной графический дисплей 240x320 или 320x240),
- ILI9481 (цветной графический дисплей 320x480, частично проверен),
- ILI9486 (цветной графический дисплей 320x480, частично проверен),
- ILI9488 (цветной графический дисплей 320x480, частично проверен),
- PCD8544 (графический дисплей 84x48),
- PCF8814 (графический дисплей 96x65),
- SH1106 (графический дисплей 128x64),
- SSD1306 (графический дисплей 128x64),
- ST7036 (символьный дисплей, 3 строки на 16 символов, не проверен),
- ST7565R (графический дисплей 128x64),
- ST7735 (цветной графический дисплей 128x160),
- ST7920 (графический дисплей 128x64),
- STE2007/HX1230 (графический дисплей 96x68),
- VT100 терминал (через последовательный интерфейс)

Заботьтесь о правильном напряжении питания LCD дисплея и допустимых логических уровнях на его шинах! Простой переключатель уровня с последовательно включенными резисторами, основанный на внутренних ограничительных диодах контроллера дисплея, может работать, но только на низкой скорости шины программного SPI. Поэтому я рекомендую использовать соответствующий преобразователь уровня на микросхемах.

Если дисплей ничего не показывает, проверьте правильность подключения и попробуйте различные настройки контрастности в файле(config<mcu>.h). Для большинства дисплеев можно аппаратно подключить выводы /CS и /RES через подтягивающие резисторы к GND/Vcc и закомментировать соответствующие настройки для портов IO, если присутствует только один дисплей на шине.

Большинство графических дисплеев имеют настройки для изменения ориентации изображения, например, для поворота изображения на 90° и зеркального отражения по горизонтали или вертикали. Таким образом, изображение можно настроить для каждого дисплея по мере необходимости. Для цветных графических дисплеев доступны дополнительные

настройки. В стандартном цветовом режиме тестер использует разные цвета, которые могут быть изменены путем редактирования файла colors.h . Если закомментировать LCD_COLOR, будет включен двухцветный режим и цвет шрифта будет COLOR_PEN, в то время как цвет фона - COLOR_BACKGROUND. В случае, если базовые цвета RGB красный и синий поменялись местами, включите LCD_BGR для реверса красного и синего цветов. У некоторых дисплеев есть обратные субпиксели RGB и контроллер дисплея не знает об этом.

Подсказка для ATМega328: если вы подключаете поворотный энкодер к PD2/PD3, пожалуйста, соедините вывод /CS модуля с PD5 и задействуйте настройку LCD_CS в config_328.h (применяется к графическим дисплеям). Иначе импульсы с поворотного энкодера, вмешавшись в шину данных, приведут к ошибкам в работе дисплея.

HD44780

HD44780 использует 4-х битный режим. Назначение контактов:

| Дисплей | Config<mcu>.h | По умолчанию |
|---------|---------------|--------------|
| | | ATMega328 |
| DB4 | LCD_DB4 | PD0 |
| DB5 | LCD_DB5 | PD1 |
| DB6 | LCD_DB6 | PD2 |
| DB7 | LCD_DB7 | PD3 |
| RS | LCD_RS | PD4 |
| R/W | | GND |
| E | LCD_EN1 | PD5 |

Можно также управлять данным ЖК-дисплеем через подключаемый I2C модуль PCF8574. Адрес I2C также должен быть настроен. Для пониженной подсветки, пожалуйста, включите LCD_BACKLIGHT_LOW. Назначение контактов для дисплея с подключенным модулем PCF8574:

| Дисплей | Config<mcu>.h | По умолчанию |
|---------|---------------|--------------|
| | | ATMega328 |
| 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 |

ILI9163

ILI9163 использует 4-х проводной SPI режим. Назначение контактов:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|-------------|
| | | ATMega328 | |
| /RESX | LCD_RES | PD4 | опционально |
| /CSX | LCD_CS | PD5 | опционально |
| D/CX | LCD_DC | PD3 | |
| SCL | LCD_SCL | PD2 | SPI clock |
| SDIO | LCD_SDA | PD1 | SPI MOSI |

Можно управлять настройками “x/y” - поворот, чтобы добиться корректной ориентации изображения для вашего дисплея. При необходимости можно настроить x-смещение.

Если активировать параметр LCD_LATE_ON, тестер запускается с очищенным дисплеем, вызывающим небольшую задержку при включении питания. Иначе возможно появление при старте некоторых случайных пикселей (шума) на экране.

ILI9341/ILI9342

ILI9341/ILI9342 использует 4-х строчный SPI или 8-битную параллельную шину. Назначение контактов для 4-х строчного SPI:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|--------------------|
| | | ATMega328 | |
| /RES | LCD_RES | PD4 | опционально |
| /CS | LCD_CS | PD5 | опционально |
| D/C | LCD_DC | PD3 | |
| SCK | LCD_SCK | PD2 | SPI clock |
| SDI | LCD_SDI | PD1 | SPI MOSI |
| SDO | LCD_SDO | - | Только для ILI9341 |

Для 8-битного параллельного:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|------------------|
| | | ATMega2560 | |
| | LCD_PORT | PORTB | |
| /RESX | LCD_RES | PB4 | опционально |
| /CSX | LCD_CS | PB5 | опционально |
| D/CX | LCD_DC | PB7 | |
| WRX | LCD_WR | PB0 | |
| RDX | LCD_RD | PB6 | опционально |
| | 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 |

Можно управлять настройками “x/y” - поворот и “x” - смещение, чтобы добиться корректной ориентации изображения для вашего дисплея. Не забывайте устанавливать координаты по “y” и “x” для контроллера (ILI9341 240x320 и ILI9342, 320x240). В некоторых модулях дисплея отключен расширенный набор команд ILI9341 (вывод EXTC подключен к Gnd). В этом случае вы можете увидеть размытое или малоконтрастное изображение, которое можно исправить, включив LCD_EXT_CMD_OFF.

Ввиду относительно большого числа пикселей, работа дисплея несколько медленная. Весь экран полностью обновляется с частотой приблизительно 3 секунды с кварцем на 8 МГц и при использовании программной эмуляции SPI. Лучше использовать аппаратный SPI или параллельную шину.

ILI9481(частично проверен)

ILI9481 использует 8-битную, 16-битную параллельные шины или 4-х строчный SPI.
Назначение контактов для 8-битной шины:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|------------------|
| | | ATMega2560 | |
| | LCD_PORT | PORTB | |
| /RESX | LCD_RES | PB4 | опционально |
| /CSX | LCD_CS | PB5 | опционально |
| D/CX | LCD_DC | PB7 | |
| WRX | LCD_WR | PB0 | |
| RDX | LCD_RD | PB6 | опционально |
| | LCD_PORT2 | PORTL | |
| DB0 | LCD_DB0 | PL0 | LCD_PORT2 pin #0 |
| DB1 | LCD_DB1 | PL1 | LCD_PORT2 pin #1 |
| DB2 | LCD_DB2 | PL2 | LCD_PORT2 pin #2 |
| DB3 | LCD_DB3 | PL3 | LCD_PORT2 pin #3 |
| DB4 | LCD_DB4 | PL4 | LCD_PORT2 pin #4 |
| DB5 | LCD_DB5 | PL5 | LCD_PORT2 pin #5 |
| DB6 | LCD_DB6 | PL6 | LCD_PORT2 pin #6 |
| DB7 | LCD_DB7 | PL7 | LCD_PORT2 pin #7 |

Назначение контактов для 16-битной шины такое же, как и для 8-битной,
но есть дополнение:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|------------------|
| | | ATMega2560 | |
| | LCD_PORT3 | PORTC | |
| DB8 | LCD_DB8 | PC0 | LCD_PORT3 pin #0 |
| DB9 | LCD_DB9 | PC1 | LCD_PORT3 pin #1 |
| DB10 | LCD_DB10 | PC2 | LCD_PORT3 pin #2 |
| DB11 | LCD_DB11 | PC3 | LCD_PORT3 pin #3 |
| DB12 | LCD_DB12 | PC4 | LCD_PORT3 pin #4 |
| DB13 | LCD_DB13 | PC5 | LCD_PORT3 pin #5 |
| DB14 | LCD_DB14 | PC6 | LCD_PORT3 pin #6 |
| DB15 | LCD_DB15 | PC7 | LCD_PORT3 pin #7 |

Назначение контактов для 4-строчного SPI:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|-------------|
| | | ATMega644 | |
| /RES | LCD_RES | PB2 | опционально |
| /CS | LCD_CS | PB5 | опционально |
| D/C | LCD_DC | PB3 | |
| SCL | LCD_SCL | PB7 | |
| SDA | LCD_SDA | PB5 | |

Из-за высокого разрешения дисплея и цветовой схемы RGB666 (3 байта на пиксель) SPI работает довольно медленно, даже для аппаратного SPI и тактовой частоты MCU 16МГц. Поэтому я бы не рекомендовал использовать интерфейс SPI.

Можно управлять настройкой (LCD_ROTATE) чтобы повернуть изображение. При необходимости вы также можете управлять настройками поворота “x” и / или “y”.

ILI9486(частично проверен)

ILI9486 использует 8-битную или 16-битную параллельные шины или 4-х строчный SPI. Назначение контактов такое же, как у ILI9481.

ILI9488(частично проверен)

ILI9488 использует 8-битную или 16-битную параллельные шины или 4-х строчный SPI. Назначение контактов такое же, как у ILI9481.

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|-------------|
| | | ATMega644 | |
| /RES | LCD_RES | PB2 | опционально |
| /CS | LCD_CS | PB5 | опционально |
| D/C | LCD_DC | PB3 | |
| SCL | LCD_SCL | PB7 | SPI clock |
| SDA | LCD_SDA | PB5 | SPI MOSI |

Из-за высокого разрешения дисплея и цветовой схемы RGB666 (3 байта на пиксель) SPI работает довольно медленно, даже для аппаратного SPI и тактовой частоты MCU 16МГц. Поэтому я бы не рекомендовал использовать интерфейс SPI.

PCD8544

PCD8544 использует 3-х проводной SPI режим. Назначение контактов:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|-------------|
| | | ATMega328 | |
| /RES | LCD_RES | PD4 | опционально |
| /SCE | LCD_SCE | PD5 | опционально |
| D/C | LCD_DC | PD3 | |
| SCL | LCD_SCLK | PD2 | SPI clock |
| SDIN | LCD_SDIN | PD1 | SPI MOSI |

Так как у дисплея всего 84 пикселя в горизонтальной развертке, вы получите только 14 символов в строке при использовании шрифта 6x8. Таким образом, 2 символа не могут быть выведены на дисплей. Смягчить этот недостаток можно, если попробовать сократить некоторый текст в variables.h.

PCF8814

PCF8814 обычно использует 3-х проводной SPI режим. Назначение контактов (только программный режим):

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|-------------|
| | | ATMega328 | |
| /RES | LCD_RES | PD4 | |
| /CS | LCD_CS | PD5 | опционально |
| SCLK | LCD_SCLK | PD2 | SPI clock |
| SDIN | LCD_SDIN | PD1 | SPI MOSI |

Можно управлять настройками “y” - поворот, чтобы добиться корректной ориентации изображения для вашего дисплея и “x” - поворот вверх-вниз с помощью MX контакта.

SH1106(проверен не полностью)

SH1106 использует 3-х проводный, 4-х проводный SPI режим или I2C. 3-х проводной SPI требует программный режим и SPI_9 для включения. Назначение контактов для 4-х проводного SPI:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|-------------|
| | | ATMega328 | |
| /RES | LCD_RESET | PD4 | опционально |
| /CS | LCD_CS | PD5 | опционально |
| A0 | LCD_A0 | PD3 | |
| SCL(D0) | LCD_SCL | PD2 | SPI clock |
| SI (D1) | LCD_SI | PD1 | SPI MOSI |

Назначение контактов для 3-х проводного SPI:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|-------------|
| | | ATMega328 | |
| /RES | LCD_RESET | PD4 | опционально |
| /CS | LCD_CS | PD5 | опционально |
| A0 | | GND | |
| SCL(D0) | LCD_SCL | PD2 | SPI clock |
| SI (D1) | LCD_SI | PD1 | SPI MOSI |

Для I2C:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|---------------------------|-------------|
| | | ATMega328 | |
| /RES | LCD_RESET | PD4 | опционально |
| /CS | | GND | опционально |
| SCL(D0) | I2C_SCL | PD1 | |
| SDA(D1) | I2C_SDA | PD0 | |
| SA0(A0) | | GND(0x3c)/ 3.3V (0x3d) | |

Используя настройки поворота по оси “x/y”, вы можете изменить ориентацию вывода, если необходимо. Для многих дисплейных модулей на базе SH1106 необходимо, чтобы смещение по x было равно 2.

SSD1306

SSD1306 использует 3-х или 4-х проводной SPI режим а также I2C. 3-х проводной SPI поддерживается только программным способом, SPI_9 должен быть включен. Назначение контактов для 4-х проводного SPI:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|-----------|---------------|--------------|-------------|
| | | ATMega328 | |
| /RES | LCD_RESET | PD4 | опционально |
| /CS | LCD_CS | PD5 | опционально |
| DC | LCD_DC | PD3 | |
| SCLK(D0) | LCD_SCLK | PD2 | SPI clock |
| SDIN (D1) | LCD_SDIN | PD1 | SPI MOSI |

Для 3-х проводного SPI(только программный режим):

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|-----------|---------------|--------------|-------------|
| | | ATMega328 | |
| /RES | LCD_RESET | PD4 | опционально |
| /CS | LCD_CS | PD5 | опционально |
| SCLK(D0) | LCD_SCLK | PD2 | SPI clock |
| SDIN (D1) | LCD_SDIN | PD1 | SPI MOSI |

Для I2C:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|------------|---------------|--------------------------|-------------|
| | | ATMega328 | |
| /RES | LCD_RESET | PD4 | опционально |
| SCL (D0) | I2C_SCL | PD1 | |
| SDA (D1&2) | I2C_SDA | PD0 | |
| SA0 (D/C) | LCD_SCLK | Gnd (0x3c) / 3.3V (0x3d) | |

Можно управлять настройками “x/y” - поворот, чтобы добиться корректной ориентации изображения для вашего дисплея.

ST7036(не проверен)

ST7036 использует 4-х битный параллельный интерфейс или 4-х проводный SPI. Назначение контактов для параллельного 4-х битного интерфейса:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|--------------------|
| | | ATMega328 | |
| DB4 | LCD_DB4 | PD0 | |
| DB5 | LCD_DB5 | PD1 | |
| DB6 | LCD_DB6 | PD2 | |
| DB7 | LCD_DB7 | PD3 | |
| RS | LCD_RS | PD4 | |
| R/W | LCD_RW | GND | optional LCD_RW |
| E | LCD EN1 | PD5 | |
| XRESET | | VCC | optional LCD_RESET |

Для 4-х проводного SPI:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|----------|---------------|--------------|-------------|
| | | ATMega328 | |
| XRESET | LCD_RESET | PD4 | опционально |
| CSB | LCD_CS | PD5 | опционально |
| RS | LCD_RS | PD3 | |
| SCL(DB6) | LCD_SCL | PD2 | SPI clock |
| SI(DB7) | LCD_SI | PD1 | SPI MOSI |

ST7036i может использовать интерфейс I2C, но он пока не поддерживается. Отдельная функция ST7036 является специальным контактом для включения расширенной системы команд (пин EXT), которая обычно используется. В том случае, если настройки LCD_EXTENDED_CMD и LCD_CONTRAST отключены, вывод должен быть закомментирован.

ST7565R

ST7565R использует 4/5 строчный SPI режим. Назначение контактов:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|----------|---------------|--------------|-------------|
| | | ATMega328 | |
| /RES | LCD_RESET | PD0 | опционально |
| A0 | LCD_A0 | PD1 | |
| SCL(DB6) | LCD_SCL | PD2 | SPI clock |
| SI(DB7) | LCD_SI | PD3 | SPI MOSI |
| /CS1 | LCD_CS | PD5 | опционально |

Можно управлять настройками “x/y” - поворот и “x” - смещение, чтобы добиться корректной ориентации изображения для вашего дисплея.

ST7735

ST7735 использует 4-х проводной SPI режим. Назначение контактов:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|-------------|
| | | | ATMega328 |
| /RESX | LCD_RES | PD4 | |
| /CSX | LCD_CS | PD5 | опционально |
| D/CX | LCD_DC | PD3 | |
| SCL | LCD_SCL | PD2 | SPI clock |
| SDA | LCD_SDA | PD1 | SPI MOSI |

Можно управлять настройками “x/y” - поворот, чтобы добиться корректной ориентации изображения для вашего дисплея. Если активировать параметр LCD_LATE_ON, тестер запускается с очищенным дисплеем, вызывающим небольшую задержку при включении питания. Иначе вы можете наблюдать при старте некоторые случайные пиксели (шум) на экране.

Недавно появился полу-совместимый вариант ЖК-дисплея, который не будет работать со стандартным драйвером. Решение состоит в том, чтобы использовать модифицированный драйвер (lcd_semi_st7735 вместо lcd_st7735) с одинаковыми настройками.

ST7920

ST7920 использует параллельную 4-х битную шину или SPI. Назначение контактов для параллельного 4-х битного интерфейса:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|-------------|
| /XRESET | LCD_RESET | Vcc | опционально |
| E | LCD_EN | PD5 | |
| RS | LCD_RS | PD4 | |
| RW | LCD_RW | GND | опционально |
| D4 | LCD_DB4 | PD0 | |
| D5 | LCD_DB5 | PD1 | |
| D6 | LCD_DB6 | PD2 | |
| D7 | LCD_DB7 | PD3 | |

Назначение контактов для SPI:

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|----------|---------------|--------------|-------------|
| /XRESET | LCD_RESET | PD4 | опционально |
| CS (RS) | LCD_CS | PD5 | опционально |
| SCLK (E) | LCD_SCLK | PD2 | SPI clock |
| SID (RW) | LCD_SID | PD1 | SPI MOSI |

Из-за неудачной адресации для ST7920 могут применяться только шрифты шириной 8 пикселей. Чтобы справиться с горизонтальной 16-ти битной адресацией, я должен был добавить экранный буфер для символов.

STE2007/HX1230

STE2007 использует 3-х проводный SPI режим. Назначение контактов (только программная поддержка):

| Дисплей | Config<mcu>.h | По умолчанию | Примечание |
|---------|---------------|--------------|-------------|
| | | ATMega328 | |
| /RES | LCD_RESET | PD4 | опционально |
| /CS | LCD_CS | PD5 | опционально |
| SCLK | LCD_SCLK | PD2 | SPI clock |
| SDIN | LCD_SDIN | PD1 | SPI MOSI |

Можно управлять настройками “x/y” - поворот, чтобы добиться корректной ориентации изображения для вашего дисплея.

VT100 терминал

Драйвер VT100 заменяет ЖК-дисплей и выводит информацию посредством VT100 терминала. Раздел конфигурации для VT100 уже включает активацию последовательного интерфейса TTL. Помните, что драйвер VT100 отключит другие опции, которые имели отношение к последовательному интерфейсу и могли бы исказить информацию.

КНОПКИ ТЕСТИРОВАНИЯ И УПРАВЛЕНИЯ

Основное управление тестером – кнопка “Тест”, но дополнительные опции управления также поддерживаются для более удобной работы.

Кнопка "Тест"

Кнопка "Тест" запускает устройство, а также управляет пользовательским интерфейсом. Для этого тестер различает кратковременные и длительные нажатия кнопки (> 0.3 с). Быстрое нажатие обычно используется для продолжения работы или выбора пункта меню. Долгое нажатие выполняет зависящее от контекста действие. Если тестер будет ожидать, пока вы нажмете кнопку, то он даст подсказку и выведет на экран курсор на последней позиции во второй строке или в нижний правый угол LCD дисплея. Немигающий курсор сигнализирует о том, что на экран будет выведена дополнительная информация, а мигающий означает, что тестер возобновит циклическое тестирование. Курсор не задействован в меню и некоторых опциях, потому что это необходимо для правильного нажатия кнопки. Возможно включение подсказок действий, если у вашего прибора есть дополнительные кнопки и возможность отображать на дисплее достаточное количество строк текста (см. UI_KEY_HINTS в config.h). Подсказка отображается вместо курсора, если он доступен. На данный момент есть только одна такая подсказка для доп. кнопок (Меню / Тест).

Поворотный энкодер (аппаратная опция)

С поворотным энкодером вы получите некоторую дополнительную функциональность в пользовательском интерфейсе, но это зависит от контекста. Некоторые режимы работы используют изменение скорости вращения энкодера, чтобы управлять дополнительными настройками или регулировкой значений.

Алгоритм для поддержки работы энкодера основан на подсчёте кода Грэя за один импульсный шаг или остановку (ENCODER_PULSES). Большинство энкодеров выдают 2 или 4 кодовых импульса за один шаг. Количество шагов или остановок за полный оборот на 360 угловых ступеней также приняты во внимание (ENCODER_STEPS). Вы можете использовать это значение для точной настройки угловой скорости и оптимизировать обратную связь. Более высокое значение замедляет скорость, более низкое значение – увеличивает её.

В случае если необходимо изменить направление вращения на обратное, просто поменяйте назначение выводов процессора для контактов А и В в файле конфигурации config_<mcu>.h. Изменение угловой скорости измеряет время для двух шагов. Вы должны повернуть энкодер, по крайней мере, на два шага для средней скорости. Для более высокой скорости - на три шага. Одиночный шаг приводит к самой низкой скорости.

Кнопки управления +/- (аппаратная опция)

Если вы предпочитаете кнопки вместо энкодера, то есть возможность добавить две альтернативные. Кнопки имеют такое же подключение как и энкодер (резисторы подтяжки, низкий уровень). Длительное нажатие кнопки увеличит "скорость", если продолжать её удерживание.

Сенсорный экран (аппаратная опция)

Альтернативная опция управления - сенсорный экран. Обратите внимание на то, что экран должен быть достаточно большой, с поддержкой приблизительно 8 текстовых строк по 16 символов в каждой. Чтобы оставить драгоценное информационное пространство на дисплее, пользовательский интерфейс не отображает иконки сенсора управления. Для этого есть невидимые сенсорные зоны слева и справа (каждая шириной в 3 символа), а также в верхней и нижней части дисплея (высотой в 2 строки) и один в центральной области. Левая и верхняя зоны служат для уменьшения значения или перемещения вверх по меню, в то время как

нижняя и правая зоны - для увеличения или перемещения вниз по меню. На самом деле они делают то же, что и энкодер. Более длительное касание сенсорной зоны увеличивают скорость изменения, если это поддерживается функцией или инструментом (аналогично быстрому вращению энкодера). Центральная зона действует аналогично кнопке "Тест", но она не активирует, например, опцию проверки стабилитронов.

Сенсорному экрану нужна настройка и калибровка для правильного функционирования. Калибровка автоматически запускается после включения тестера, если нет сохраненных данных в EEPROM. Вы можете также выполнить настройку через главное меню. Процедура простая. Если вы видите звездочку (желтую "*" на цветном дисплее), просто коснитесь её. После этого тестер удаляет звездочку и отображает исходную позицию x/y. Первая точка регулировки находится сверху справа, вторая точка - внизу слева. Вы можете использовать регулировку до трех раз. Можно пропустить процедуру в любое время если нажать тестовую кнопку. Если есть проблемы с калибровкой в виде странных "x/y" позиций, проверьте ориентацию сенсорного экрана относительно дисплея.

У драйвера есть опции зеркального отражения или изменения ориентации. Предполагается что верхний левый угол дисплея - это нулевая позиция. Некоторые подсказки о необходимых настройках для конкретных значений x и y:

первая точка регулировки: сверху справа

| | x | y | настройки |
|---|------|------|-----------------------------------|
| • | низ | низ | TOUCH_поворот_X |
| • | низ | верх | TOUCH_поворот_X & TOUCH_поворот_Y |
| • | верх | низ | |
| • | верх | верх | TOUCH_поворот_Y |

Не забывайте сохранять данные после успешной настройки (главное меню: сохранить).

Поддерживаемые контроллеры сенсорного экрана:

- ADS7843/XPT2046

Вы найдете параметры конфигурации дисплея в файле config-<mcu>.h(в настоящее время это config_644.h и config_1280.h из-за отсутствия свободных портов в ATmega328).

ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

Есть несколько способов для того, чтобы сделать вывод результатов на дисплей более удобным и легко читаемым.

Для дисплеев с более чем 2 текстовыми строками:

- показать кнопки подсказки вместо курсора, если включена настройка (UI_KEY_HINTS).

Для графических дисплеев:

- все параметры перечисленные ранее;
- иконки с распиновкой для 3-х выводных полупроводников (SW_SYMBOLS, неприменимо для дисплеев с низким разрешением).

Для цветных графических дисплеев:

- все параметры перечисленные ранее;
- цветовое обозначение для тестовых контактов (UI_PROBE_COLORS);
- настраиваемый цвет для заголовков (UI_COLORED_TITLES);
- настраиваемый цвет для курсора и подсказок для кнопок (UI_COLORED_CURSOR);
- цветовой код для резисторов (SW_R_E24_5_CC, SW_R_E24_1_CC и SW_R_E96_CC).

СОЕДИНЕНИЕ С КОМПЬЮТЕРОМ

Тестер может поддерживать последовательный интерфейс TTL для связи с ПК. Это может быть соединение только по TX для вывода информации о найденных компонентах

или двунаправленный интерфейс для автоматизации. В обоих случаях необходим последовательный интерфейс TTL, который настраивается в файле config.h (см. раздел «Busses»).

Специальные символы заменяются стандартными, например, омега (Ом) становится простой R. Для меню и инструментов нет последовательного вывода, кроме результатов проверки оптопары.

Таблица преобразования:

символы диода |> <|

символ конденсатора ||

омега R

micro / μ u

символ резистора []

Подсказки:

- 9600 8N1;

- новая строка <CR> <NL>

Последовательный выход

Тестер определяет компоненты и передает информацию на ПК посредством простой терминальной программы, если эта функция включена (см. UI_SERIAL_COPY в разделе «misc settings» в файле config.h). Последовательный вывод следует за выводом информации на ЖК-дисплее, но только для найденных компонентов.

Автоматизация

Функция автоматизации позволяет управлять тестером с помощью удаленных команд через двунаправленное последовательное соединение. Для включения этой функции см. UI_SERIAL_COMMANDS в разделе «misc settings» в файле config.h. По умолчанию поведение тестера немного изменится. Автоматизация обеспечивает режим ожидания и прибор не будет автоматически проверять компонент после включения. Командный интерфейс довольно прост. Вы отправляете команду - тестер отвечает. Связь основана на текстовых строках ASCII и командах, чувствительных к регистру. Каждая командная строка должна заканчиваться символом <CR> <NL> или <NL>. Имейте в виду, что тестер будет принимать команды только в режиме ожидания обратной связи от пользователя после включения питания, отображения компонента или выполнения функции меню. Строки отклика заканчиваются символом <CR> <NL>. См. раздел «Команды управления» для объяснения списка команд.

VT100

Тестер может выводить информацию на терминал VT100 вместо жидкокристаллического дисплея (см. VT100 в разделе "Дисплеи"). Для сохранения корректной информации все другие опции для последовательного интерфейса отключены.

ВКЛЮЧЕНИЕ

Долгое нажатие кнопки при запуске прибора выбирает режим автоудержания показаний. В этом режиме устройство ожидает быстрого нажатия кнопки после отображения результата прежде, чем продолжить работу. Иначе тестер выбирает непрерывный (циклический) режим работы по умолчанию. Вы можете инвертировать выбор режима работы в config.h (UI_AUTOHOLD). После включения будет кратко показана микропрограммная версия. Более длительное нажатие кнопки (> 2 с) сбросит тестер к микропрограммным значениям по умолчанию. Это может быть полезным, если вы установили недопустимую контрастность дисплея. Если тестер обнаружит проблему с сохраненными значениями корректировок, то он выведет на экран ошибку контрольной суммы. Эта ошибка указывает на поврежденный EEPROM и устройство будет использовать микропрограммные значения по умолчанию.

Для тестера с ручным выключателем питания вместо программного, пожалуйста, включите опцию `POWER_SWITCH_MANUAL` в `config.h`. В этом режиме прибор не сможет отключиться автоматически!

ТЕСТИРОВАНИЕ

После запуска прибор входит в режим тестирования и определяет подключенный компонент. В непрерывном (циклическом) режиме это действие автоматически повторится после короткой паузы. Если несколько раз компонент не будет определён, то тестер выключится. В режиме автоудержания – “autohold”(подсказка курсором) тестер выполняет один запуск и ожидает нажатия кнопки или поворота вправо энкодера. Задержка цикла и автоматическое выключение питания для непрерывного режима могут быть скорректированы настройками `CYCLE_DELAY` и `CYCLE_MAX` в `config.h`. Существует дополнительное автоматическое выключение питания для режима “autohold” (`POWER_OFF_TIMEOUT`), которое активно только во время циклического тестирования. В обоих режимах можно вывести меню с дополнительными функциями или выключить устройство. Для получения дополнительной информации читайте ниже.

КОНТРОЛЬ ИСТОЧНИКА ПИТАНИЯ

Каждый цикл тестирования начинается с определения напряжения батареи и вывода на дисплей её состояния (норма, слабая, разряжена). Тестер выключится при достижении нижнего порогового напряжения. Батарея контролируется регулярно во время работы. Конфигурация по умолчанию для контроля питания установлена для 9-ти вольтовой батареи, но она может быть изменена для большинства других источников питания. Посмотрите секцию "power management" в `config.h` для всех вариантов настройки. Контроль питания может быть полностью отключен опцией `BAT_NONE`, установлен на проверку постоянного напряжения для источников питания менее 5 В опцией `BAT_DIRECT` или установлен для контроля через делитель напряжения - опции `BAT_R1`, `BAT_R2` и `BAT_DIVIDER`. Некоторые тестеры поддерживают дополнительный внешний источник питания, но не позволяют его контролировать. В этом случае предусмотрена опция `BAT_EXT_UNMONITORED`, которая позволяет предотвратить проблемы с автоматическим выключением питания при низком уровне заряда. В этом режиме состояние батареи будет отображаться на дисплее как "ext"(внешняя). Для слабого и низкого уровня заряда предусмотрены настройки `BAT_WEAK` и `BAT_LOW`, а опция `BAT_OFFSET` учитывает падение напряжения, например, если в схеме установлен защитный диод и транзистор управления питанием структуры `rpr`.

ВЫКЛЮЧЕНИЕ

При отображении результата последнего теста долгое нажатие кнопки “Тест” приводит к выключению прибора. Он покажет сообщение “bye!”(до свидания) а затем выключится. При более коротком нажатии кнопки тестер останется включенным. Это вызвано особенностью схемы управления питанием.

МЕНЮ

Вы можете войти в меню двумя короткими нажатиями кнопки “Тест” после отображения последнего найденного компонента или выполненной функции (может потребоваться некоторая практика). Если поддержка поворотного энкодера включена, поворот влево также позволит войти в меню. Старый метод входа в меню путем короткого замыкания всех трех тестовых зондов также может быть включен через опцию (см. `UI_SHORT_CIRCUIT_MENU`).

Если вызвано меню, то короткое нажатие кнопки выбирает следующий пункт, а долгое нажатие выполняет его. На дисплее с 2 строками можно видеть навигационную подсказку в нижнем правом углу. Символ “>”, если есть следующий пункт, или ”<” для самого

последнего пункта (возврат к первому пункту). В дисплеях с более чем 2 строками выбранный пункт отмечен символом ‘*’ с левой стороны.

С поворотным энкодером можно перемещаться по пунктам вверх или вниз, изменяя направление вращения и коротким нажатием кнопки запускать выбранный пункт. Возврат к первому пункту также возможен. Некоторые функции подсказывают вам схему расположения тестовых контактов, которые необходимо использовать при тестировании. Эта информация будет выведена на экран в течение нескольких секунд, но может быть пропущена коротким нажатием кнопки "Тест".

Для функций, генерирующих сигнал на тестовый контакт № 2 - он используется в качестве вывода по умолчанию. В этом случае контакты № 1 и № 3 связаны с общим проводом (GND). Если тестер будет сконфигурирован для вывода специального сигнала на порт (OC1B), тестовые контакты не используются и их схема не будет выведена на экран.

ШИМ генератор

Это та функция, которую ждали. Прежде, чем скомпилировать прошивку, выберите генератор ШИМ с простым пользовательским интерфейсом или с альтернативным интерфейсом для тестеров с поворотным энкодером и крупным дисплеем. Возможно отображение длительности импульса, если включить в конфигурации PWM_Show_duration.

Конфигурация выхода:

Тестовый контакт №2: выход ШИМ сигнала (с резистором на 680 Ом для ограничения тока);

Тестовый контакт №1 и №3: общий GND(земля).

Простой интерфейс генератора ШИМ

Сначала вы должны выбрать желаемую частоту ШИМ в простом меню. Короткое нажатие кнопки - выбор нужной частоты, длинное - запускает вывод ШИМ для выбранной частоты. Коэффициент ШИМ регулировки устанавливается по умолчанию на 50%. Короткое нажатие кнопки "Тест" увеличивает скважность на 5%, долгое нажатие - уменьшает на 5%. Выход из функции - нажатие кнопки "Тест" дважды быстро. Если подключён энкодер, можете использовать его, чтобы выбрать частоту в меню и сделать ШИМ регулировку с шагом в 1%.

Альтернативный интерфейс генератора ШИМ

Переключение между частотой и длительностью осуществляется нажатием кнопки "Тест". Выбранный параметр отмечается звёздочкой. Поверните энкодер по часовой стрелке, чтобы увеличить значение или против часовой чтобы уменьшить его. Более быстрый поворот энкодера увеличивает размер шага настройки. Долгое нажатие кнопки устанавливает значение по умолчанию (частота - 1кГц, скважность - 50%). Два коротких нажатия кнопки "Тест" - выход из режима генератора.

Генератор прямоугольного сигнала

Сигнальный генератор выдаёт меандр с переменной частотой до 1/4 от тактовой частоты процессора (2МГц для кварца на 8МГц). Частота по умолчанию 1000Гц, можно изменять её, вращая энкодер. Скорость вращения определяет изменение частоты, т.е. медленное вращение - малый шаг изменения, быстрое вращение - больший шаг изменения частоты. Так как генерация сигналов основана на тактовом режиме работы процессора, возможна только пошаговая установка частоты. Для низких частот шаги довольно мелкие, но для высоких частот они становятся больше и больше. Долгое нажатие кнопки "Тест" сбрасывает частоту по умолчанию до 1кГц, а два коротких нажатия кнопки – выход из режима генерации сигналов.

Конфигурация выхода:

Тестовый контакт №2: вывод (с ограничительным резистором на 680Ом);

Тестовый контакт №1 и №3: общий GND(земля).

Подсказка: для работы в этом режиме необходим энкодер!

Тестирование стабилитронов (аппаратная опция)

Если в устройстве присутствует встроенный преобразователь DC-DC, который создает высокое испытательное напряжение для измерения напряжения пробоя стабилитрона, то он должен быть соединён со специальными тестовыми контактами. Если кнопка “Тест” нажата и работает преобразователь, то на дисплее тестера будет отображаться текущее напряжение. Если кнопка удерживалась нажатой достаточно долго для стабильного теста напряжения, то после её отпускания кнопки будет показано минимальное измеренное напряжение. Можно повторять этот тест, пока вам нравится. Для выхода из режима дважды быстро нажмите кнопку "Тест". Если в приборе используется делитель напряжения 10:1 без преобразователя для измерения внешнего напряжения или повышающий DC-DC работает все время, вы можете выбрать альтернативный режим (ZENER_UNSWITCHED), в котором напряжение измеряется периодически без нажатия кнопки “Тест”. Когда вы видите курсор справа внизу, то можно выйти из режима двумя быстрыми нажатиями тестовой кнопки. Есть еще один вариант, чтобы запустить проверку стабилитронов во время автоматического тестирования (HW_PROBE_ZENER). Если ни один из компонентов не будет определен на стандартных тестовых контактах, Тестер проверит напряжение на контактах ZENER(K-A).

Если ваш тестер имеет нестандартный делитель напряжения (не 10:1), можно включить конфигурацию ZENER_DIVIDER_CUSTOM и указать номиналы резисторов (ZENER_R1 и ZENER_R2).

Как подключить стабилитрон:

Контакт + : катод;

Контакт - : анод.

Внутрисхемное измерение ESR

Конденсаторы тестируются внутрисхемно и на экран выводится значение ёмкости и ESR, если измерение обнаруживает допустимый конденсатор. **Убедитесь, что конденсатор разряжен прежде, чем подключить к нему тестер!** Значения могут отличаться от стандартных измерений (вне схемы), потому что любой компонент в схеме, расположенный параллельно с тестируемым конденсатором, будет влиять на результат. Для начала измерения нажмите кнопку "Тест". Два быстрых коротких нажатия кнопки – выход из этого режима.

Подключение конденсатора:

Тестовый контакт №1: плюс конденсатора;

Тестовый контакт №3: минус конденсатора GND(земля).

Проверка конденсатора на ток утечки

Тест утечки в конденсаторе заряжает его и выводит на экран ток и напряжение через известное сопротивление. Первоначально, заряд конденсатора осуществляется через R1 (680Ом) а затем через резистор Rh (470кОм), когда ток оказывается ниже определенного порога. Каждый цикл начинается с напоминания схемы контактов. После подключения конденсатора нажмите кнопку “Тест” (или поверните ручку энкодера вправо), чтобы запустить процесс зарядки. Затем, после окончания заряда, нажмите кнопку снова, тестер разрядит конденсатор и будет отображать напряжение на нём до тех пор, пока разряд не достигнет нижнего порога. Чтобы выйти из проверки, нажмите кнопку “Тест” дважды.

Подсказка: обращайтесь внимание на полярность электролитических конденсаторов!

Как подсоединить конденсатор:

Тестовый контакт №1: положительный вывод;

Тестовый контакт №3: отрицательный вывод Gnd(земля).

R/C/L - монитор

Монитор непрерывно измеряет пассивный компонент, подключенный к контактам №1 и 3. После запуска режима прибор на несколько секунд отображает подсказку распиновки контактов, которую можно пропустить, нажав кнопку тестирования. Между измерениями есть задержка в одну или две секунды, обозначенная значком курсора в правом нижнем углу, во время которой можно выйти из мониторинга двумя короткими нажатиями тестовой кнопки.

Доступные варианты мониторов:

- R-Monitor (сопротивление);
- C-Monitor (емкость + опционально ESR);
- L-Monitor (индуктивность);
- R/C/L-Monitor (R + опционально L или C + опционально ESR);
- R/L-Monitor (сопротивление + индуктивность).

LC-измеритель(аппаратная опция)

Аппаратная опция LC-измерителя основана на простой схеме генератора LC, используемой в недорогих измерителях на PIC контроллере. Обычная схема (82мкГн и 1нФ) имеет базовую частоту около 595кГц, а подключение дополнительного конденсатора или индуктивности уменьшает эту частоту. С помощью эталонного конденсатора с известным значением, измеренными частотами и математической формулы неизвестный конденсатор или индуктивность могут быть вычислены. Измерители PIC LC обычно имеют диапазоны измерений 10нГн - 100мГн и 0,1пФ - 900нФ. Они используют время стробирования 100мс для частотомера. ПО использует автоматическое переключение диапазона со временем стробирования 100мс и 1000мс чтобы улучшить разрешение для компонентов с малым значением. Таким образом, диапазоны начинаются примерно с 1нГн и 10fF(0.01пФ). Максимальная измеряемая индуктивность составляет примерно 150мГн.

Что касается верхнего предела емкости, то у меня возникла проблема с печатной платой. При емкости около 33нФ на выходном сигнале начинали появляться выбросы на переднем и заднем фронтах, искажая работу частотомера, который видит больше импульсов, чем есть на самом деле. Это известная проблема с цепями генератора на LM311. После испытания нескольких модификаций без видимого успеха, компаратор на основе триггера Шмидта кажется лучшим выбором для очистки выходного сигнала генератора. CMOS 4093 работает нормально в этом качестве. С чистым выходным сигналом максимальная измеряемая емкость составляет около 120нF (выше генератор LC становится нестабильным). Тестер обеспечивает более низкую границу частоты чем 10 кГц, то есть возможны теоретически максимальные значения 250нН или 3,5 мкФ, если генератор LC будет работать стабильно.

При запуске LC-метра тестер выполнит самонастройку, указанную сообщением «настройка ...». После этого можно подключить измеряемый конденсатор или индуктивность. Короткое нажатие кнопки переключает C и L режимы измерения (режим по умолчанию - C). Частота LC-генератора дрейфует со временем (до 100Гц) и требует повторной настройки. Если вы увидите увеличение нулевого значения или "-" без подключенных компонентов, запустите самонастройку долгим нажатием кнопки. Если есть проблема с самонастройкой или вы отмените ее, нажав какую-либо кнопку, тестер выйдет из режима измерений и сообщит об ошибке. Два коротких нажатия кнопки – выход из режима LC-измеритель.

Подсказки: в качестве эталонного нужно использовать высококачественный пленочный конденсатор емкостью 1нФ (с хорошей стабильностью). Вы также можете использовать обычный пленочный конденсатор на 1нФ, просто измерьте его емкость с помощью надежного LCR-измерителя и обновите значение `LC_METER_C_REF`. Если вас интересует частота LC-генератора и ее дрейф, то включите опцию `LC_METER_SHOW_FREQ`.

Частотомер (аппаратная опция)

Есть два варианта частотомера. Базовый – простой пассивный вход на контакте T0 MCU. У расширенного варианта есть входной буфер, два генератора для тестирования кварцев (с низкой и высокой частотами) а также дополнительный делитель частоты. Принципиальные схемы для обоих вариантов изображены в документации от Karl-Heinz.

Базовый вариант

Частотомером можно измерить частоты приблизительно от 10Гц до 1/4 тактовой частоты процессора, с разрешением 1Гц для частот ниже 10кГц. Частота измеряется и выводится на экран дисплея постоянно, пока вы не закончите измерение двойным нажатием кнопки. Алгоритм автоматически выбирает диапазон длительности между 10мс и 1000мс на основе частоты. Вывод T0 может использоваться совместно с дисплеем.

Расширенный вариант

Расширенный вариант частотомера имеет дополнительный делитель и позволяет измерять более высокие частоты. Теоретический верхний предел - 1/4 от тактовой частоты процессора, умноженная на делитель (16:1 или 32:1). Настройки определены в файле `config_<mcu>.h`. Так же не забывайте устанавливать корректное значение делителя частоты в `config.h`. Тестовый канал частотомера (буферизованный вход, кварцевый генератор для низкой и высокой частоты) настраивается нажатием кнопки “Тест” или вращением энкодера. Как обычно, два коротких нажатия кнопки “Тест” – выход из режима частотомера.

Счетчик событий (аппаратная опция)

Счетчик событий использует вывод T0 в качестве выделенного входа и срабатывает от переднего фронта сигнала. Вывод T0 нельзя использовать совместно с дисплеем. Рекомендуется простой метод подсчета. Режим счетчика выбирается в небольшом меню, которое также отображает счетчик значений. Пункты меню выбираются коротким нажатием, настройки меняются поворотным энкодером или дополнительными кнопками.

- Счетчик - подсчет времени и событий;
- Время (n) - подсчет событий за заданный период времени;
- События(t) - подсчет времени для заданного количества событий.

Первый пункт - режим счетчика. Второй пункт «n» - это количество событий. В режиме событий он покажет порог срабатывания, который можно изменить. Длительное нажатие кнопки сбрасывает порог к значению по умолчанию (100). В других режимах подсчета этот пункт заблокирован. Следующий пункт «t» - это период времени в секундах. Тот же алгоритм и для режима времени (значение по умолчанию 60с). Последний элемент стартует или останавливается после долгого нажатия кнопки. Когда счетчик запускает подсчитанные события, прошедшее время обновляется каждую секунду и после остановки отображается результат. Ограничение на период времени составляет 43200с (12 часов), а для событий - $4 * 10^9$. Если какой-либо из этих пределов превышен, подсчет автоматически останавливается. Предел или порог события (в режиме событий) проверяется каждые 200мс, поэтому может быть некоторое превышение в случае более 5 событий/сек.

Выход триггера

Дополнительно можно включить выход триггера (EVENT_COUNTER_TRIGGER_OUT), чтобы управлять каким-нибудь другим устройством. Выход триггера установлен на высокий уровень при подсчете, т. е. на подъем фронта при старте и спаде при остановке.

Назначение контактов выход триггера:

Тестовый контакт №2: выход (с резистором 680Ом для ограничения тока);

Тестовый контакт №1 и №3: GND(земля).

Тест поворотного энкодера

Этот тест проверяет поворотные энкодеры для определения схемы его контактов. Вам необходимо подсоединить контакты А, В и общий контакт энкодера к тестовым контактам и повернуть энкодер на несколько шагов по часовой стрелке. Для алгоритма тестирования нужен поворот энкодера минимум на четыре шага, чтобы определить надлежащую функцию и схему контактов. Изменение направления вращения важно для правильного определения неизвестных А и В контактов.

Когда поворотный энкодер будет обнаружен, прибор выведет на экран схему контактов и будет ожидать нажатия кнопки (или момент для циклического режима) прежде, чем продолжить тестирование. Чтобы выйти из теста энкодера нажмите кнопку "Тест" один раз во время тестирования.

Контрастность дисплея

Вы можете скорректировать уровень контрастности для некоторых графических LCD модулей. Короткое нажатие кнопки увеличивает уровень, долгое - уменьшает его. Два коротких нажатия кнопки - выход из настройки. Поворотный энкодер также позволяет скорректировать уровень контрастности.

Приёмник ИК RC/Декодер

Эта функция обнаруживает и декодирует сигналы от пультов дистанционного управления IR и требует подключения модуля приемника IR, например серии TSOP. При компиляции прошивки вы можете выбирать между двумя вариантами - приёмником TSOP, который подключен к тестеру и аппаратным встроенным модулем приёмника ИК. Первый требует соединения модуля ИК со стандартными тестовыми контактами. Второй - аппаратный модуль ИК, соединенный со специальным портом микропроцессора. На дисплее тестера будет отображён обнаруженный протокол, а также адрес (если доступен), команда и, в некоторых случаях, дополнительные данные в шестнадцатеричном виде.

Формат:

<протокол> < зона данных>

Для ошибочного или неизвестного пакета и протокола будет показан символ "?". Для неизвестного протокола тестер выводит на экран число пауз и импульсов, продолжительность первого импульса и первой паузы в модулях 50µs:

? <импульсы>:<первый импульс>-<первая пауза>

Если число импульсов одинаково для различных кнопок RC - наиболее вероятна PDM или ШИМ модуляция. Изменяющееся число импульсов указывает на двухфазную модуляцию. Чтобы выйти из этого режима нажмите кнопку "Тест".

Поддерживаемые протоколы и их массив данных:

- JVC
<адрес >:<команда>
- Kaseiкуо (японский код, 48 бит)
<код производителя >:<система>-<устройство>:<функция>
- Matsushita(Panasonic MN6014, C6D6 / 12 бит)
<выборочный код>:<кодовые данные>

- Motorola
<команда>
- NEC (стандарт и расширенный)
<адрес>:<команда>
R для повтора цикла
- Proton / Mitsubishi (M50560)
<адрес>:<команда>
- RC 5 (стандарт)
<адрес>:<команда>
- RC 6 (стандарт)
<адрес>:<команда>
- Samsung / Toshiba (32 бита)
<выборочный код>:<кодовые данные>
- Sharp / Denon
<адрес>:<команда>
- Sony SIRC (12, 15 и 20 бит)
12 15: <команда >:<адрес>
20: <команда >:<адрес>:<дополнительно>

Дополнительные протоколы:

- IR60 (SDA2008/MC14497)
<команда>
- Matsushita (Panasonic MN6014, C5D6 / 11 бит)
<выборочный код>:<кодовые данные>
- NEC μ PD1986C
<кодовые данные>
- RECS80 (standard & extended)
<адрес>:<команда>
- Sanyo (LC7461)
< выборочный код >:<ключ>
- RCA
<адрес>:<команда>
- Thomson
<устройство>:<функция>

Несущая частота модуля приемника TSOP не обязательно должна точно соответствовать частоте RC. Несоответствие уменьшает возможный диапазон, но это не имеет большого значения для этого приложения.

Модуль приёмника ИК, соединенный с тест контактами

Пожалуйста, подключайте модуль приемника ИК только после входа в режим! Как соединить модуль TSOP:

Тестовый контакт №1: общий GND(земля);

Тестовый контакт №2: Vs – питание (с ограничительным резистором на 680Ом);

Тестовый контакт №3: выход/данные.

Подсказка: токоограничивающий резистор по Vs подразумевает применение модуля приемника ИК с диапазоном напряжения питания от 2.5 до 5 В. Если вы используете только 5В

модуль, то можно отключить ограничительный резистор в config.h файле (на ваш собственный риск). **Любое короткое замыкание в схеме может повредить процессор!**

Аппаратный модуль приёмника ИК

Для аппаратного модуля приёмника ИК используемый порт процессора назначается и настраивается в файле конфигурации config_<MCU>.h.

ИК RC Передатчик

Передатчик ИК RC отправляет ИК-коды, которые вы ввели и предназначен для проверки ИК RC- ресиверов или устройств с дистанционным управлением. Этот инструмент требует для работы дополнительные кнопки либо поворотный энкодер, дисплей с более чем 4 строками и несложную схему для управления ИК-светодиодом.

На дисплее отображается протокол, несущая частота, скважность и нескольких полей для ввода данных. Коротким нажатием кнопки тестирования вы можете переключаться между пунктами. Выбранный элемент обозначается символом '*'. Используйте энкодер (или другой режим ввода), чтобы изменить настройку/значение пункта. Длительное нажатие кнопки "Тест" отправляет ИК-код до тех пор, пока вы держите кнопку нажатой. Как обычно, два коротких нажатия кнопки - выход из функции.

При изменении протокола несущая частота и скважность устанавливаются в значения по умолчанию для протокола. Но вы можете изменить их, если хотите. Несущая частота может быть установлена от 30 до 56кГц, а скважность 1/2 (50%), 1/3 (33%) или 1/4 (25%). Поля данных представляют собой настраиваемые пользователем части ИК-кода и поясняются ниже. В большинстве случаев это всего лишь адрес и команда.

Поддерживаемые протоколы и поля данных:

- JVC
<адрес:8> <команда:8>
- Kaseiyo (Japanese Code)
< код производителя:16> <система:4> <продукт:8> <функция:8>
- Matsushita (Panasonic, MN6014 12 бит)
< пользовательский код:6> <данные:6>
- Motorola
<команда:9>
- Стандарт NEC
<адрес: 8> <команда: 8>
- NEC Extended
<адрес: 16> <команда: 8>
- Proton / Mitsubishi (M50560)
<адрес:8> <команда:8>
- RC-5 Standard
<адрес:5> <команда:6>
- RC-6 Standard, Mode 0
<адрес:8> <команда:8>
- Samsung / Toshiba (32 бит)
<пользовательский код: 8> < данные: 8>
- Sharp / Denon
<адрес:5> <команда:8> <маска:1>
- Sony SIRC-12
<команда: 7> <адрес: 5>
- Sony SIRC-15
<команда: 7> <адрес: 8>
- Sony SIRC-20
<команда: 7> <адрес: 5> <дополнительный: 8>

Дополнительные протоколы (SW_IR_TX_EXTRA):

- Thomson
<устройство:4> <функция:7>

Поля данных разделяются пробелами, а их синтаксис:

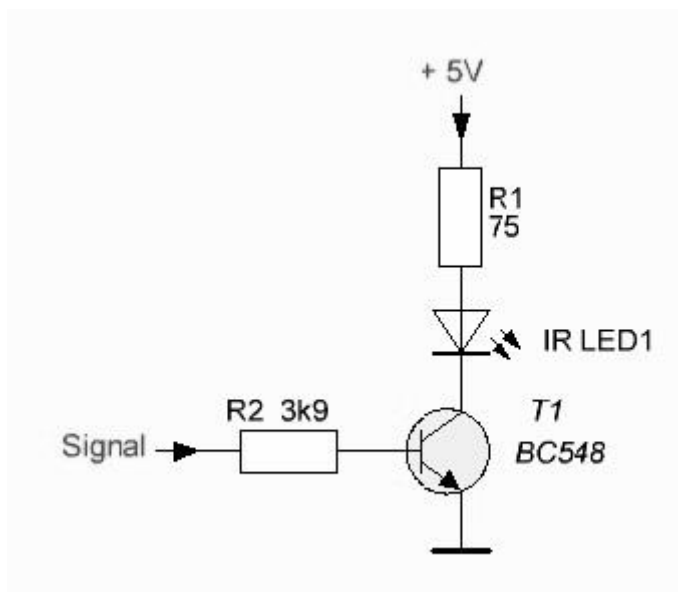
<наименование поля>: <битность>

Конфигурация выхода:

Тестовый контакт №2: выходной сигнал (с сопротивлением 680 Ом для ограничения тока);

Тестовый контакт №1 и №3: GND(земля).

Выход сигнала (контакт № 2) имеет резистор ограничения тока и может управлять ИК-светодиодом с током не более 5mA, что недостаточно для ИК-светодиодов с типичным потреблением 100mA. Поэтому вам нужна простая схема управления на основе переключающего транзистора, ИК-светодиода и резистора для ограничения тока. Пример схемы управления ИК-светодиодом (V_f 1,5 В, при токе 100 мА) с потреблением тока примерно 50mA:



Подсказка: если длительность импульса/паузы является ошибочной, активируйте альтернативную циклическую задержку опцией SW_IR_TX_ALTDELAY. Она может потребоваться, если компилятор Си оптимизирует стандартный цикл задержки, несмотря на определенные команды для хранения ассемблерного кода.

Тестирование оптронов

Эта функция проверяет оптроны и показывает вам значения V_f LED, CTR (также I_f), задержки t_{on}/t_{off} (BJT тип). Она поддерживает стандартные оптроны с BJT транзисторами NPN структуры, NPN структуры Дарлингтона и оптосимисторы. Для измерения параметра “CTR” порт ввода-вывода процессора испытывает перегрузку, приблизительно, в течение 3 мс. Спецификация определяет максимальный выходной ток в 20mA, но мы перегружаем порт ввода-вывода приблизительно до 100mA. Поэтому, максимальное значение CTR огра

ничено и любой оптрон со значением CTR выше 2000% нужно тестировать с осторожностью. Максимальный управляющий ток для LED, который нужен для определения оптосимистора - 5mA. В элементах с задержкой (MOSFET back to back) определение ВJT и CTR будет бессмысленно. Тестирование оптронов со встречно включенными светодиодами проигнорировано. Для тестирования вам нужен простой адаптер со следующими 3 контрольными точками:

- Тип – оптотранзистор ВJT:
 - Анод LED;
 - Катод LED и эмиттер ВJT соединить вместе;
 - Коллектор ВJT;
- Тип – оптосимистор:
 - Анод LED;
 - Катод LED и МТ1 оптосимистора соединить вместе;
 - МТ2 оптосимистора.

Можно подключить адаптер к любым 3 тестовым контактам прибора. Тестер определит расположение контактов автоматически. После входа в режим подключите адаптер и нажмите кратковременно кнопку "Тест" для начала тестирования оптрона. Если он будет найден, на дисплее появится тип оптрона и дополнительные параметры. Если компонент не обнаружен, на дисплее будет надпись "не определён". Мигающий курсор указывает на то, что вы должны нажать тестовую кнопку (или повернуть энкодер) для старта нового тестирования. Два коротких нажатия кнопки "Тест" - выход из режима, как обычно.

Тестирование сервоприводов RC

Эта функция формирует ШИМ сигнал для сервоприводов RC, которые управляются 1-2мс импульсами. Поддерживаются фиксированные частоты ШИМ 50, 125, 250 и 333Гц, а длительность импульса может быть между 0.5 и 2.5мс. Есть также свип-режим с изменяемой длительностью импульса между 1 и 2мс с регулируемой скоростью перестройки. Вы можете скорректировать длительность импульса энкодером. По часовой стрелке для установки более длинного импульса, против часовой стрелки - для более короткого импульса. Длинное нажатие кнопки сбрасывает импульс к 1.5мс. Можно переключаться между режимами выбора длительности импульса и выбора частоты коротким нажатием кнопки "Тест" (отмечается звездочкой). В частотном режиме используется энкодер для выбора частоты ШИМ. Длительное нажатие кнопки "Тест" включает или отключает возможность перестройки (появляется символ "<->" после частоты). Если этот режим включен - энкодер позволяет изменять период. Как обычно, два коротких нажатия кнопки "Тест" - выход из функции.

Конфигурация выхода:

Тестовый контакт №2: выход ШИМ (с резистором на 680Ом, ограничивающим ток);
Тестовый контакт №1 и №3: общий GND(земля).

Подсказка: вы должны обеспечить дополнительный источник питания для сервопривода.

Некоторые схемы распиновки типичных 3-х контактных соединителей сервопривода:

| Производитель | контакт 1 | контакт 2 | контакт 3 |
|---------------|-----------------|-----------|-----------|
| Airtronics | PWM White/Black | Gnd Black | Vcc Red |
| Futaba | PWM White | Vcc Red | Gnd Black |
| Hitec | PWM Yellow | Vcc Red | Gnd Black |
| JR Radios | PWM Orange | Vcc Red | Gnd Black |

Сканирование 1-Wire шины

1-Wire сканирование определяет коды ПЗУ всех подключенных устройств. Пожалуйста, см. раздел «Шины и интерфейсы» для настройки шины 1-Wire. При использовании щупов тестер сообщит о назначении контактов и будет ждать, пока не обнаружит внешний подтягивающий резистор. Можете пропустить это нажатием кнопки «Тест». Каждый раз, когда вы нажимаете кнопку тестирования, прибор будет сканировать текущее (следующее) подключенное устройство и отображать его ПЗУ код (в шестнадцатеричном формате). Первая часть кода – информация о производителе, вторая часть - его серийный номер. CRC опущен. Код производителя $\geq 0x80$ (установлен бит 7) указывает на уникальный код клиента, а первые три (слева) шестнадцатеричные цифры серийного номера - это идентификатор клиента. Тестер сообщит о том, что найдено последнее устройство, а также выдаст информацию об ошибках CRC и шины. В случае завершения сканирования или ошибки шины вы можете начать новый процесс сканирования, нажав кнопку тестирования. И, как обычно, два коротких нажатия кнопки «Тест» - выход из режима.

Температурный датчик DS18B20

Эта функция проверяет температурный датчик 1-Wire DS18B20 и выводит на экран температуру. См. раздел "Шины и интерфейсы" для настройки шины 1-Wire. После запуска тестер сообщит вам назначение контактов и ожидает, что будет обнаружен внешний подтягивающий резистор. Можно пропустить это нажатием кнопки «Тест». После подключения DS18B20 к тестовым контактам нажмите кнопку «Тест» ещё раз - датчик будет считан (это может занять почти секунду). Чтобы выйти из этого режима нажмите кнопку «Тест» дважды быстро. Длительным нажатием кнопки вы можете выбрать автоматический режим (автоматическое обновление), который будет обозначен знаком '*' после наименования датчика.

Назначение контактов для датчиков:

Тестовый контакт №1: GND (земля);

Тестовый контакт №2: DQ (данные);

Тестовый контакт №3: Vcc (ток ограничен резистором 680Ом);

Требуется внешний подтягивающий резистор 4,7кОм между DQ и Vcc!

Датчики температуры и влажности DHTxx

Эта функция для тестирования DHT11, DHT22 и совместимых датчиков температуры и влажности. Сначала тестер подсказывает назначение контактов, а затем ожидает внешнего подтягивающего резистора. После обнаружения резистора, на дисплее отображается выбранный тип датчика (по умолчанию: DHT11) и короткое нажатие тестовой кнопки считывает его показания. При успешном прочтении прибор выводит измеренные значения, при любой ошибке результат будет равен "-". Длительное нажатие кнопки изменяет тип датчика, а два коротких нажатия кнопки - выход из опции. При смене типа датчика у вас также есть возможность активировать его автоматическое считывание (каждую секунду), которое обозначается знаком '*' после типа датчика.

Поддерживаемые типы датчиков:

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

Назначение контактов для тестирования:

Тестовый контакт №1: GND(земля);

Тестовый контакт №2: Данные;

Тестовый контакт №3: Питание Vdd (ток не ограничен);

Необходим внешний резистор подтяжки 4,7кОм между контактами №3 и №2!

Некоторые датчики уже включают в себя подтягивающий резистор 10кОм, который работает также хорошо с короткими кабелями.

Подсказка: из-за требований к питанию датчика измерительный резистор 680Ом не может быть использован для ограничения тока. Имейте в виду, что любое короткое замыкание может повредить MCU!

Самотестирование

Если вы вошли в режим самотестирования, необходимо закоротить все три тестовых контакта. Тестер ожидает, пока это будет сделано. В случае любой проблемы можно прервать процедуру нажатием кнопки "Тест". Этим действием вы пропустите полную проверку. Функция самопроверки запускает каждый тест по 5 раз. Можно пропустить любой тест коротким нажатием кнопки или пропустить весь долгим нажатием. В тесте T4 нужно удалить "закоротку" тестовых контактов. Тестер будет ожидать, пока вы действительно не удалите короткое замыкание.

Описание тестов:

- T1 - проверка напряжения внутреннего ИОНа (в мВ);
- T2 - сравнение резисторов R1 (смещение в мВ);
- T3 - сравнение резисторов Rh (смещение в мВ);
- T4 - **удалить короткое замыкание тестовых контактов!**
- T5 - проверка контактов на утечку в режиме "pull-down" (напряжение в мВ);
- T6 - проверка контактов на утечку в режиме "pull-up" (напряжение в мВ).

Автокорректировка

Автокорректировка измеряет сопротивление и ёмкость тестовых контактов или щупов, т.е. плата, печатные проводники и контакты образуют суммарное значение для создания нулевого смещения. Эта функция также измеряет внутреннее сопротивление тестового порта процессора в режиме ввода-вывода. Если эти тесты пропущены, или будут странные значения измеренных значений, то по умолчанию будут использованы параметры из config.h. Если все прошло нормально, тестер будет отображать и использовать новые значения, полученные при автокорректировке (они не будут автоматически сохранены в EEPROM, смотрите пункт "Сохранение/Загрузка" ниже). Смещение напряжения аналогового компаратора автоматически скорректировано при измерении ёмкости (в нормальном режиме тестирования без учёта автокорректировки), если конденсатор находится в диапазоне 100nF до 3.3µF. Смещение для встроенного ИОНа определено таким же образом. Прежде, чем сделать автокорректировку в первый раз, измерьте калибровочный конденсатор со значением ёмкости между 100nF и 3.3µF, по крайней мере три раза, чтобы тестер смог скорректировать упомянутые выше смещения. Обычно, первое измерение приведет к пониженному значению ёмкости, второе – к повышенному, а третье будет наиболее точным. Обе величины смещения будут выведены на экран в конце процедуры. С аппаратным калибровочным конденсатором (только с ATmega644/1284) автоматическая обработка смещения при измерении емкости заменена специальной функцией, выполняемой во время процедуры корректировки. Таким образом, вы не должны замерять пленочный конденсатор дополнительно. Если смещения емкости различаются между парами тестовых контактов, можно включить проверку смещения пары в файле config.h (CAP_MULTIOFFSET). Такая же поправка возможна для смещения величины сопротивления (R_MULTIOFFSET). Автокорректировка очень похожа на самотестирование по процедуре и пользовательскому интерфейсу.

Описание этапов корректировки:

- A1 - определение смещения для ИОНа и аналогового компаратора (только для опции с аппаратным конденсатором);
- A2 - определение сопротивления щупов/зажимов (от 10 миллиОм);
- A3 - **удалить короткое замыкание тестовых контактов/щупов!**
- A4 - внутреннее сопротивление портов процессора по отношению к земле Gnd (напряжение RiL);
- A5 - внутреннее сопротивление портов процессора по отношению к питанию Vcc (напряжение RiH);
- A6 - определение ёмкости щупов/зажимов (в pF);

Ограничения:

- сопротивление тестовых контактов < 1.50Ом для двух соединённых контактов;
- ёмкость тестовых контактов < 100pF.
- - Внутреннее сопротивление порта ввода-вывода в режиме (RiL) < 25 Ом
- - Внутреннее сопротивление порта ввода-вывода в режиме (RiH) < 29 Ом

Подсказка: если сопротивление для тестовых щупов/зажимов изменяется слишком сильно - может быть проблема в контактах!

Помните: корректировка - это не калибровка! Калибровка - это процедура сравнения результатов измерений с известным измерительным стандартом и сохранения этого различия. Цель состоит в том, чтобы контролировать дрейф в течение долгого времени. Корректировка – это процедура для настройки устройства согласно определенным задачам.

Сохранение/Загрузка

После выполнения корректировки вы можете обновить константы настроек в EEPROM по умолчанию, используя режим “Save”(Сохранение). В следующий раз, когда вы включите прибор, обновленные значения (профиль # 1) будут загружены и использованы автоматически. Для удобства можно сохранить и загрузить два профиля, например, если используется два набора различных тестовых контактов или зажимов. Идея режима сохранения состоит в том, чтобы предотвратить автоматическую запись данных корректировки. Если нужно использовать другие зажимы для некоторых измерений, тестер будет скорректирован для работы с этими временными зажимами. Если вернуться назад к стандартным контактам, прибор не должен приспособливаться, потому что старые значения всё еще сохранены. Просто выключите и включите питание тестера. Есть опция (UI_CHOOSE_PROFILE) для автоматического входа в меню загрузки профиля после включения тестера.

Информация о корректировке

Выводит на экран текущие значения корректировок и используемые смещения. Использование внешнего 2.5-вольтового источника опорного напряжения обозначено символом ‘*’ после значения Vcc.

Выключение

Эта функция выключит тестер, если активна настройка SW_POWER_OFF.

Выход

Если вы вошли в меню по ошибке, можно выйти из него этой командой.

РЕЗИСТОРЫ

Резисторы измеряются дважды (в обоих направлениях) и значения сравниваются. Если значения отличаются намного, тестер предполагает, что есть два резистора вместо одного. В этом случае прибор выводит на экран результат как два резистора с номерами контактов "1 - 2 - 1", и отличающимися двумя значениями сопротивления. Для резисторов с сопротивлением ниже, чем 100 Ом выполняется дополнительное измерение с более высоким разрешением. В некоторых, редких случаях, прибор не может обнаружить очень малое сопротивление. Тогда повторно выполните тест.

Если включена дополнительная проверка значений норм ряда E (SW_R_E *), тестер берет следующее более низкое и следующее более высокое значение нормы и сравнивает их с измеренным сопротивлением и с учетом допусков компонентов. Есть два режима вывода. В текстовом режиме прибор отображает ряд E и применяемый допуск, сопровождаемый соответствующими значениями нормы. Символ «-» означает, что нет соответствия значению нормы. В режиме вывода цветового кода тестер отображает ряд E и цветовой код резистора, включая цветовую полосу для допуска. Помните, что цвета могут отличаться в зависимости от модуля дисплея и используемого цветового режима.

Если какой-либо цвет отключен, просто отрегулируйте значение цвета (COLOR_CODE_*) в файле colors.h. Поиск в Интернете по запросу "RGB565 tool" покажет много онлайн инструментов для создания / выбора значений цвета RGB565. Для резисторов SMD есть дополнительная опция для отображения кода EIA-96 (подразумевает E96 1%).

КОНДЕНСАТОРЫ

Для измерения ёмкости используется три метода. Конденсаторы ёмкостью $> 47\mu\text{F}$ измеряются методом зарядки импульсами длительностью 10 мс. Конденсаторы в диапазоне от $4.7\mu\text{F}$ и $47\mu\text{F}$ тестируются таким же образом, но с зарядным импульсом длительностью 1 мс. Конденсаторы с малой ёмкостью тестируются методом аналогового компаратора. Таким образом, точность измерения различных конденсаторов оптимизирована.

Большие емкости требуют коррекции. Без исправления измеренные значения слишком большие. По моему скромному мнению, это вызвано методом измерения. Начиная с преобразования ADC после зарядки импульсом, требуется некоторое время для того, чтобы конденсатор потерял некоторый заряд из-за внутренних резистивных потерь. Также самому преобразователю ADC требуется некоторый заряд для работы. Из-за этого процесса необходимо больше времени для зарядки электролита и у конденсатора измеряется большее значение ёмкости. Измерение разряда затем пытается компенсировать этот эффект, но оно может сделать это только частично. Поправочные коэффициенты (CAP_FACTOR_SMALL, CAP_FACTOR_MID и CAP_FACTOR_LARGE в файле config.h) подобраны для работы с большинством моделей тестеров. В некоторых случаях вам, возможно, придется их изменить. Была добавлена логика при измерении больших ёмкостей, которые могут быть определены как резисторы. Резисторы $< 100\text{ Ом}$ дополнительно тестируются на предмет того, не являются ли они большими ёмкостями. Нижний диапазон измерения ёмкостей начинается от 5 pF (включая нулевое смещение), и эти значения допустимы. Более низкие значения слишком сомнительны и могут быть вызваны расположением щупов (тестовых контактов) и будут немного отличаться.

Тестер пытается измерить ESR для конденсаторов с номиналом от 10 nF и выше. Альтернативно можно включить старый метода замера ESR конденсаторов с номиналом от 180 nF и выше. Но так как измерение ESR не сделано сигналом переменного тока с определённой частотой, не ожидайте стабильных результатов. Используемый метод может быть сопоставим с тестом на частоте 1 кГц. Так или иначе, результаты достаточно хороши при проверке электролитических конденсаторов. Для небольших емкостей вы можете получить различные результаты в зависимости от тактовой частоты процессора. Я предположил бы, что г-н Фурье в состоянии объяснить это. Другие проведенные измерения определяют ток утечки (саморазряда) для конденсаторов с ёмкостью более чем $4.7\mu\text{F}$. Это дает подсказку о

состоянии электролитического конденсатора. В моих тестах типичное значение саморазряда для исправных электролитов составило:

- 10-220 μ F 1-3 μ A;
- 330-470 μ F 4-5 μ A;
- 470-820 μ F 4-7 μ A;
- $\geq 1000\mu$ F 5-7 μ A на каждые 1000 μ F.

Дополнительная проверка значений норм ряда E также доступна для конденсаторов (SW_C_E *), но только в текстовом режиме, потому что существует слишком много разных вариантов цветовых кодов.

ИНДУКТИВНОСТИ

Измерения индуктивности не очень точны, так как частота работы процессора, конструкция печатной платы тестера оказывают влияние на результаты. Базовый метод основан на измерении времени между подачей электрического тока и достижением его определенного уровня. Для больших индуктивностей есть проверка небольшим током, для малых индуктивностей нужна проверка большим током, который превышает нагрузочный предел порта MCU в течение очень короткого времени (приблизительно до 25 микросекунд). При исследовании эффекта изменения частоты работы MCU и других вещей я нашел образец отклонений, который может использоваться для компенсации. Поэтому добавлена возможность пользовательской тонкой настройки.

В файле inductor.c в функцию "MeasureInductor" введена переменная "Offset" для компенсации. Эта переменная - смещение для опорного напряжения. Положительное смещение будет уменьшать значение индуктивности, отрицательное – увеличит его. Компенсация при проверке большим током зависит от частоты работы MCU и разделена на три отдельных диапазона, каждый со специальным смещением. При тесте небольшим током - только простая компенсация в данный момент, необходимы дополнительные тесты. Если вы видите какие-либо большие отклонения результатов измерений при сравнении с образцовым LCR - измерителем, можно скорректировать смещения, чтобы получить более точные параметры. Если вы хотите проверить значения норм ряда E, включите в config.h опцию (SW_L_E *) (только текстовый режим).

Подсказка: при получении неожиданных результатов повторно запустите тест.

Измерение индуктивности выполняется только если её активное сопротивление менее 2кОм.

РАЗРЯД КОМПОНЕНТОВ

Тестер пытается разрядить любой подключенный компонент до и во время измерения. Когда он не может разрядить компонент ниже указанного порога (CAP_DISCHARGED), то выдаст ошибку, отображающую номер т. контакта и оставшееся напряжение.

Режим разряда основан не на фиксированной паузе, он адаптирует себя к разрядному уровню. Таким образом, батарея будет идентифицирована быстрее (около 2с), а конденсаторы большой ёмкости будут разряжаться дольше. Если такой конденсатор будет определён как батарея, пожалуйста, повторите проверку. При помехе или ошибке, возможно, понадобится скорректировать параметр CAP_DISCHARGED приблизительно до 3 мВ. Индикация остаточного напряжения поможет вам выбрать нужное значение.

ADC СУПЕРДИСКРЕТИЗАЦИЯ

Функция ADC изменена, чтобы поддерживать супердискретизацию переменной (1-255 отсчётов). Значение по умолчанию – 25 выборков. Можно попытаться улучшить точность из

мерения, увеличивая число выборок. Обратите внимание на то, что большое число выборок будет занимать больше времени, приводя к более медленным измерениям.

ПРЯМОЕ ПАДЕНИЕ НАПРЯЖЕНИЯ V_{BE} BJT_s

Во время проверки диодов значение V_f будет измерено с R_I (большой тестовый ток) и R_h (малый тестовый ток), а также сохранены оба напряжения. Процедура тестирования для BJT ищет соответствующий переход для параметра V_{BE} и интерполирует два измерения V_f на основе коэффициента h_{FE} для виртуального текущего теста. Таким образом, мы добиваемся наиболее подходящих результатов для различных видов транзисторов, начиная с V_f для маломощных BJT и заканчивая более мощными BJT.

ОТОБРАЖЕНИЕ РЕЗУЛЬТАТОВ НА ДИСПЛЕЕ

Изменены некоторые обозначения и применены сокращения. Отображение информации может быть разделено на несколько частей и использовать многостраничный режим для того, чтобы поддерживать дисплей всего с несколькими строками.

Для одного диода низкое значение действующего V_f (измеренное током 10µA) показано в фигурных скобках, если напряжение ниже 250мВ. Это должно подсказать вам, что это, возможно, германиевый диод. Большинство таблиц спецификации для германиевых диодов определяет V_f при токе в 0.1mA, который тестер не поддерживает. При более высоком действующем V_f, как правило, будет приблизительно 0.7В и трудно отличить германий от кремниевых диодов. Ток утечки I_R для одного диода или IC_{EO} для BJT будет выведен на экран, если он превышает 50nA. Германиевые транзисторы BJT_s имеют ток утечки от нескольких µA до, приблизительно, 500µA. Германиевые диоды обычно имеют ток утечки в несколько µA.

Для некоторых элементов также отображается ёмкость. В случае если ёмкость ниже 5pF или измерение приводит к сбою по некоторым причинам, выведенное на экран значение будет 0pF. Если обнаружен FET транзистор со встроенным каналом, симметричным стоком и истоком, например, JFET, схема контактов показывает 'x' вместо 'D' или 'S', потому что оба вывода могут быть функционально идентичными. Посмотрите спецификацию на FETs, если вам нужно больше информации о цоколёвке. Цоколёвка для симистора показана как IDs 'G', '1' и '2'. '1' - это вывод MT1, '2' - MT2. Для однопереходных UJT транзисторов, в случае, если их детектирование включено, '1' - база B1, '2' - база B2 и 'E' - эмиттер.

Если включена нестандартная опция определения цоколёвки (выбран SW_SYMBOLS в config.h), будет показан соответствующий символ с соответствующей распиновкой для 3-х контактных полупроводников. Если на дисплее недостаточно пространства для символа, его вывод будет пропущен.

Дополнительные подсказки

- **BJT_s**

Строчная буква после значения h_{FE} указывает тип тестовой цепи, которая используется для измерения h_{FE}:

- e: схема с общим эмиттером
- c: схема с общим коллектором

Если вы включили вывод тестового тока h_{FE} (SW_HFE_CURRENT), то тестер отобразит I_C для схемы с общим эмиттером и I_E для схемы с общим коллектором.

При проверке диодов V_f измеряется с R_I (высокий испытательный ток) и R_h (низкий испытательный ток), и оба напряжения сохраняются. Функция вывода для BJT ищет соответствующий диод для V_{BE} и интерполирует два измерения V_f на основе транзисторов h_{FE} для виртуального тестового тока. Так мы получим более подходящие результаты для различных типов транзисторов, так как V_{BE} для слаботочного BJT не измеряется таким же испытательным током, что и для мощного BJT.

Для BJT с внутренним резистором (база-эмиттер) тестер отобразит на экране этот резистор и пропустит измерение hFE. Если у BJT есть встроенный защитный диод, такой транзистор может быть обнаружен как BJT или как два диода вместе с резистором база-эмиттер (низкоомный резистор - 2 диода). Тогда тестер покажет эти два диода и резистор с намеком на возможную NPN - структуру или PNP BJT. Проблема состоит в том, что низкоомный встроенный резистор мешает нормальной идентификации BJT.

Другой особый случай - BJT с интегрированным защитным диодом той же структуры что и BJT. Этот диод создает эффект “паразитного” транзистора. У BJT NPN-структуры будет “паразитный” PNP и наоборот. Такой транзистор будет помечен символом + после типа BJT. Для транзистора Шоттки (BJT с барьером Шоттки) ограничивающий диод между базой и коллектором и V_f определяются, если детектирование включено (SW_SCHOTTKY_BJT). Обратите внимание, что транзистор Шоттки имеет повышенный I_{CEO} .

- **Симисторы**

Симисторы могут использоваться в трех или четырех различных режимах работы, также известных как квадрант. Обычно некоторые параметры будут отличаться для каждого квадранта, например, ток срабатывания затвора (I_{GT}). В некоторых случаях возможно, что тестовый ток достаточен для срабатывания затвора в одном квадранте, но не в другом. Так как два тестовых прогона необходимы, чтобы вычислить контакты для MT1 и MT2 тестер не сможет определить различие между ними, т.е. контакты могли поменяться местами. Вы также можете найти симисторы, которые имеют слишком высокий ток удержания (I_H), препятствующий их правильному обнаружению. Если симистор имеет слишком высокий ток срабатывания затвора - тестер обычно обнаруживает только резистор.

- **Диоды CLD**

Диодная проверка идентифицирует CLD (диод со стабилизацией тока) как стандартный диод и отображает I_F как ток утечки. Обратите внимание на то, что анод и катод у CLD инвертированы по сравнению со стандартными диодами. Специальная проверка на CLD сложна в реализации, начиная с определения тока утечки для германиевого или высоковольтного диода Шоттки, находящегося в диапазоне I_F ($\geq 33\mu A$). Если у диода есть необычное прямое падение напряжения, довольно низкий V_f при небольшом тестовом токе (2-е значение в фигурных скобках) и емкость совсем не измерена, тогда, скорее всего, это CLD.

НЕПОДДЕРЖИВАЕМЫЕ КОМПОНЕНТЫ

Любой полупроводник, требующий высокого напряжения или большого тока переключения, не может правильно определиться, так как тестер обеспечивает максимальный ток до 7mA и максимальное напряжение 5В. Таким образом, такие компоненты как DIAC с V_{BO} 20-200V невозможно проверить. То же самое для тиристорov и симисторов с большим током запуска.

ИЗВЕСТНЫЕ ПРОБЛЕМЫ И СПОСОБЫ ИХ РЕШЕНИЯ

Если у вашего тестера есть одна из следующих проблем, вы можете попробовать включить обходной путь:

Проблема: - hFE слишком высокое.

При использовании схемы с общим коллектором с R_I в качестве базового резистора, базовое напряжение по неизвестной причине измеряется как очень малое. Так появляется базовый ток, который вызывает слишком высокое значение hFE. Тестеры, у которых замечена эта проблема - Hiland M644! Обходной путь - включите настройку “NO_HFE_C_RL” в config.h!

- Конденсаторы (серия Panasonic NF) могут быть обнаружены как диод или два встречных диода. Измерение ёмкости также не даёт определить приемлемое значение.
- При использовании SMPS или преобразователя DC-DC в качестве источника питания, тестер иногда обнаруживает конденсатор ёмкостью 50µF, даже если не подключено никакого компонента.
- ESR у конденсаторов небольшой емкости может меняться в зависимости от частоты работы процессора.
- Проблема с измерением ESR у твердотельных LowESR электролитических конденсаторов.

ПОДДЕРЖКА

Есть два форума для поддержки пользователей:

- <https://www.mikrocontroller.net/topic/248078>

Основной язык форума - немецкий, но и английский также возможен.

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

Основной язык форума - только английский.

ЖУРНАЛ ИЗМЕНЕНИЙ

Пожалуйста, прочитайте файл CHANGES!

КОМАНДЫ УПРАВЛЕНИЯ

Когда тестер принимает команды управления, ответ может содержать следующие текстовые строки, кроме командных ответов, содержащих данные:

ERR

- неизвестная команда;
- команда не поддерживается в контексте компонента;
- переполнение буфера.

OK

- команда выполнена
(некоторым командам может потребоваться некоторое время для обработки).

N/A

- информация / значение не определено.

Ответ с данными никогда не отображается ни с одной из стандартных текстовых строк выше, чтобы предотвратить любую возможную путаницу.

Основные команды:

VER

- определяет версию прошивки;
- проверить подключение и определить набор команд на основе версии;
- пример ответа: «1.33m».

OFF

- выключает тестер;
- тестер отвечает «ОК» перед отключением питания;
- пример ответа: «ОК» < тестер отключен>.

Команды тестирования:

PROBE

- тестирует т.контакты и пропускает любые паузы, ожидающие обратной связи с пользователем;
- тестер отвечает «ОК» после завершения проверки;
- пример ответа: < истекает время для тестирования> «ОК».

COMP

- определяет идентификатор типа компонента (см. COMP_* в файле common.h для идентификаторов);
- пример ответа для BJT: «30».

MSG

- возвращает сообщение об ошибке;
- применяется только при возникновении ошибки (COMP: 1);
- ответ может отличаться в зависимости от языка пользовательского интерфейса;
- пример ответа: «Батарея? 1:1500 мВ».

QTY

- определяет количество компонентов;
- пример ответа для BJT: «1»

NEXT

- выбор следующего компонента;
- применяется, если найдены два компонента (QTY: 2);
- пример ответа: «ОК».

TYPE

- определяет более конкретный тип компонента;
- применяется к BJT, FET и IGBT;
- доступные типы:
- NPN NPN (BJT);
- PNP PNP (BJT);
- JFET JFET (FET);
- МОП-транзистор (полевой транзистор);
- N-канальный n-канал (FET, IGBT).
- P-ch p-channel (FET, IGBT);
- enh.- режим обогащения(FET, IGBT);
- dep.- режим истощения (FET, IGBT);
- пример ответа для BJT: "NPN";
- пример ответа для FET (MOSFET): «MOSFET n-ch enh».

HINT

- отображает подсказки по специальным функциям;
- применяется для диодов, BJT, FET и IGBT;
- доступные подсказки:
- NPN, возможно NPN BJT (диод);
- PNP, возможно PNP BJT (диод);
- резистор в базе-эмиттер R_BE (диод, BJT);
- BJT + встроенный flyback диод на той же подложке, создающий 2-й "паразитный" BJT (BJT);
- D_FB встроенный / flyback диод (BJT, FET, IGBT);

- SYM симметричный сток и исток (FET);
- пример ответа для BJT: «D_FB R_BE»;
- пример ответа для FET (MOSFET): «D_FB».
- D_CLAMP BJT с барьером Шоттки (BJT)
требуется включения обнаружения транзистора Шоттки.

MHINT

- возвращает подсказки по измерениям;
- относится к BJT;
- подсказки доступны для:
- h_FE_e - измерение h_FE выполняется с помощью схемы с общим эмиттером (BJT);
- h_FE_c - измерение h_FE выполняется по схеме с общим коллектором (BJT)
- пример ответа для BJT: "h_FE_e".

PIN

- определяет цоколёвку компонента;
- используемые идентификаторы:
- резистор x = подключен, - = не подключен;
- конденсатор x = подключен, - = не подключен;
- диод A = анод, C = катод, - = не подключен;
- BJT B = база, C = коллектор, E = эмиттер;
- FET G = затвор, S = исток, D = сток, x = сток/ исток;
- IGBT G = затвор, C = коллектор, E = эмиттер;
- SCR G =затвор, A = анод, C = катод;
- TRIAC G = затвор, 2 = MT2, 1 = MT1;
- PUT G =затвор, A = анод, C = катод;
- UJT E = эмиттер, 2 = B2, 1 = B1.
- формат ответа:
<идентификатор зонда № 1> <идентификатор зонда № 2> <идентификатор зонда № 3>;
- пример ответа для резистора: "xx-";
- пример ответа для диода: «C-A»;
- пример ответа для BJT: «EBC».

R

- определяет значение сопротивления;
- применяется к резистору (содержит индуктивность);
- пример ответа: «122R».

C

- определяет значение емкости;
- применяется к конденсатору;
- пример ответов: "98nF" "462uF".

L

- определяет значение индуктивности;
- применяется к резистору (содержит индуктивность);
- пример ответа: «115uH».

ESR

- определяет значение ESR (эквивалентное последовательное сопротивление);
- требует включения измерения ESR;
- применяется к конденсатору;
- пример ответа: "0.21R".

I_I

- определяет значение I_{leak} (эквивалентный ток утечки собственного разряда);
- применяется к конденсатору;
- пример ответа: «3.25uA».

V_F

- определяет значение V_F (прямое напряжение);
- применяется к диоду и PUT;
- также применяется к встроенному диоду MOSFET и обратноточному диоду BJT или IGBT;
- пример ответа: «654mV».

V_{F2}

- определяет значение V_F слаботокового измерения (прямое напряжение);
- применяется к диоду;
- пример ответа: «387mV».

C_D

- определяет значение C_D (емкость диода);
- применяется к диоду;
- пример ответа: "8pF".

I_R

- определяет значение I_R (обратный ток);
- применяется к диоду;
- пример ответа: «4.89uA».

R_{BE}

- определяет значение R_{BE} (резистор база-эмиттер);
- применяется к диоду и BJT;
- пример ответов: "38.2R" "5171R".

h_{FE}

- определяет значение h_{FE} (усиление постоянного тока);
- соответствует BJT;
- пример ответа: «234».

h_{FE_r}

- определяет обратное значение h_{FE} (коллектор и эмиттер перевернуты);
- соответствует BJT;
- пример ответа: «23».

I_C

- определяет тестовый ток I_C для измерения hFE;
- требует вывода тестового тока для включения измерения hFE;
- измерение hFE для схемы с общим эмиттером;
- соответствует BJT;
- пример ответа: «3245uA».

I_E

- определяет тестовый ток I_E для измерения h_{FE} ;
- требует вывода тестового тока для включения измерения h_{FE} ;
- измерение h_{FE} для схемы с общим коллектором;
- соответствует BJT;
- пример ответа: «3245uA».

V_{BE}

- определяет значение V_{BE} (напряжение база-эмиттер);
- соответствует BJT;
- пример ответа: «657mV».

I_{CEO}

- определяет значение I_{CEO} (ток коллектор-эмиттер, открытая база);
- соответствует BJT;
- пример ответа: «460.0uA».

V_{th}

- определяет значение V_{th} (пороговое напряжение);
- применяется к FET (MOSFET) и IGBT;
- пример ответа: «2959mV».

C_{GS}

- определяет значение C_{GS} (емкость затвора);
- применяется к FET (MOSFET);
- пример ответа: «3200pF».

R_{DS}

- определяет значение $R_{DS_{on}}$ (сопротивление сток-исток);
- применяется к FET (MOSFET);
- пример ответа: «1.20R».

I_{DSS}

- определяет значение I_{DSS} (ток истока-стока, нулевое смещение / закороченный затвор);
- применяется к FET (режим истощения);
- пример ответа: «6430uA».

C_{GE}

- определяет значение C_{GE} (емкость затвор-эмиттер);
- соответствует IGBT;
- пример ответа: "724pF".

V_{GT}

- определяет значение V_{GT} (напряжение триггера затвора);
- применяется к SCR и TRIAC;
- пример ответа: «865mV».

V_T

- определяет значение V_T (напряжение смещения);
- применяется к PUT;
- пример ответа: «699mV».

R_BB

- определяет значение R_BB (сопротивление между базами);
- требует включения UJT для определения;
- применяется к UJT;
- пример ответа: «4758R».

V_GS_off

- возвращает значение V_GS (off) (напряжение отсечки);
- относится к FET (режим истощения);
- пример ответа: «-3072мВ».

V_Z

- возвращает значение V_Z (Zener или внешнее напряжение);
- требуется проверка ZENER во время обычного тестирования для включения;
- относится к Zener Diode или внешнему напряжению (опция оборудования);
- пример ответа: «6750 мВ»

ПОЛЕЗНЫЕ РЕСУРСЫ

- Немецкий форум:
<https://www.mikrocontroller.net/topic/248078>
- Английский форум:
[https://www.eevblog.com/forum/testgear/\\$20-1cr-esr-transistor-checker-Project/](https://www.eevblog.com/forum/testgear/$20-1cr-esr-transistor-checker-Project/)
- Российский форум:
<https://vrtp.ru/index.php?showtopic=16451&st=0>
<https://vrtp.ru/index.php?showtopic=25020&st=0>
- Информация о различных клонах, документация, схема и прошивки:
(indman@eevblog)
<https://yadi.sk/d/yW8xa5NJgUo5z>
- Руководство по компиляции, файлы для установки Winavr (indman@eevblog):
https://drive.google.com/file/d/1-IJA8uTcsCA_6SYHEuMydjfS2vNgmwdH/view?usp=sharing
- Актуальное k-firmware
<https://github.com/kubi48/TransistorTester-source>
- Актуальная документация для k-firmware
<https://github.com/kubi48/TransistorTester-documentation>

ССЫЛКИ НА ПЕРВОИСТОЧНИКИ

- [1] AVR-Transistortester, Markus Frejek, Embedded Projects Journal, 2011-11
- [2] <http://www.mikrocontroller.net/topic/131804> thread of Markus Frejek, Forum, 2009
- [3] <http://www.mikrocontroller.net/articles/AVR-Transistortester> Online documentation of the Transistortester, Online Article, 2009-2011
- [4] http://www.mikrocontroller.net/articles/AVR_Transistortester Short description of the TransistorTester, Karl-Heinz Kübbeler, Online Article, 2012