

ZX-Unikeyboard PS/2

PS/2 контроллер и необходимое ПО для опроса и построения прошивку под произвольную клавиатурную матрицу до 26 пин.

valerum@rambler.ru

Russia, 2021

andy.karpov@gmail.com

Ukraine, 2021

Эта тема на форуме:

<https://zx-pk.ru/threads/32497-kak-peredelat-praktichieski-lyubuyu-klaviaturu-v-ps-2/page8.html>

Содержание

Краткий HOW-TO по настройке произвольной клавиатуры.....	1
Этап 1. Прошивка для опроса матрицы.....	1
Этап 2. Тестирование опроса матрицы.....	2
Этап 3. Шаблон матрицы клавиатуры.....	2
Этап 4. Опрос и анализ матрицы.....	2
Этап 5. Сборка основной прошивки.....	3

Краткий HOW-TO по настройке произвольной клавиатуры.

Этап 1. Прошивка для опроса матрицы

Подключаем клавиатуру к атмеге (32, 16 или какой там захочется - главное, чтоб было как минимум четыре 8-разрядных порта) и собираем под нее `avr-keyexplorer.ino` в Arduino IDE.

Если нет поддержки нужного контроллера, ставим его, например, из модуля **MightyCore** через менеджере плат -

https://mcudude.github.io/MightyCore/package_MCUdude_MightyCore_index.json), как добавить в менеджер плат - читать здесь: <https://alexgyver.ru/lessons/naked-chip/>.

Обращаем внимание на установку частоты **8 МГц** в IDE при сборке прошивки, а также на установку соответствующих фьюзов (**LOW=0x04, HIGH=0xD7** – важно получить 8 МГц на внутреннем генераторе). В случае других частот/вариантов тактирования правим то и другое единообразно.

Если есть отклонения от схемы в подключении пинов клавиатурного разъема к атмеге - приводим в соответствие все **#define PIN*** в скетче.

Прошиваем сборку «тестовой» прошивки в атмегу своим любимым программатором и соответствующим софтом.

Этап 2. Тестирование опроса матрицы

Подключаем **UART** атмеги к ПК через любой удобный преобразователь, смотрим дебаг с соответствующего компорта в режиме **115200 8-н-1**, можно пользоваться прилагаемой программой **PortMon**, можно **встроенным монитором** ком-порта в Arduino IDE. Нажимаем все клавиши своей клавиатуры, они все должны генерировать нажатия с парой чисел в дебаговом выводе - это номера пары пинов для каждой клавиши.

Этап 3. Шаблон матрицы клавиатуры

Подготавливаем **файл клавиатурного шаблона** — это текстовый файл с именами всех имеющихся клавиш по одному в каждой строке (допускаются цифры и буквы латиницы, а также знаки «-» и «_»; прочие символы в именах могут вызвать ошибки компиляции прошивки на следующих этапах).

Для анализа рекомендуется внести **ВСЕ** клавиши, а не только предполагаемые к использованию, иначе может некорректно пройти анализ столбцов и строк и останутся непривязанные клавиши. Лучше вносить их не в алфавитном порядке, а в соответствии с расположением на клавиатуре (например, слева направо и сверху вниз).

Можно отредактировать имеющийся шаблон из двух приложенных (Asus X200 – 82 клавиши без кейпада, Asus X551 – 102 клавиши с кейпадом, но без выделенного Insert).

Этап 4. Опрос и анализ матрицы

Запускаем **UniKeyboardConf_en.exe** (или **.._ru.exe**), выбираем COM-порт, куда подключен UART для атмеги, в режиме 115200 8-н-1 (остальные параметры не важны).

При необходимости можно отредактировать файл клавиатурного шаблона — Edit Keyboard template (из программы вызывается potepad для его редактирования), сохраняем.

Загружаем файл шаблона (Open Keyboard Template), удостоверяемся, что считано нужное количество клавиш. Запускаем считывание (**Start reading keys**), нажимаем все клавиши в соответствии с подсказками; если произошла ошибка — можно заново начать считывание кнопкой «Start reading keys».

По завершении считывания всех клавиш из списка будет предложено **начать анализ данных** («**Would you like to analyze...**»), после завершения анализа будет дан комментарий по результату, построен и предложен к сохранению файл **customkey.h**. Его следует сохранить в каталог с прошивкой **avr_kbd_ps2/includes/**.

В результате анализа будут получены два непересекающихся множества линий клавиатуры — продолжных и поперечных. Те, что в меньшем количестве, будут далее считаться столбцами, остальные — строками. В файле **customkey.h** будут включены все определения имен клавиш с префиксом **KEY_**, массив соответствующих им линий и два массива с номерами линий-столбцов (**cols**) и линий-строк (**rows**).

Внимание! Если после анализа выдается предупреждение о том, что остались неприсвоенные строки/столбцы так как в паре строка/столбец оказалась всего одна клавиша, нужно проанализировать, что это за клавиши и не было ли ошибок ввода. Если указанные клавиши не планируется использовать на Спеке, то можно не обращать внимания на это предупреждение — в файл **customkey.h** они не будут включены.

Этап 5. Сборка основной прошивки

Ревизируем исходный код основной прошивки **avr_kbd_ps2** на предмет соответствия имен клавиш в кейсах внутри функции **kbd_event** тому, что сформировано в **customkey.h**, добиваемся отсутствия ошибок компиляции, если нужно – корректируем обработку нужных клавиш. Собираем прошивку, загружаем ее в Atmega.