**Проект “Структурная схема цифровой тропосферной станции Р-423-1”**

Рис. 1 Стенд

Куратор:

* п/п-к Романовский С.В.

Разработчики:

* ст. Шиханцов И.А. - [ivanka6342@gmail.com](mailto:ivanka6342@gmail.com), @ivanka6342, +375 (29) 9457146
* ст. Петрович В.А. - @vp1242, +375(29)8625913

**Содержание**

* Введение
* Оборудование
* Структура кода
* ИК-приемник и пульт управления
* Контроллер светодиодной матрицы MAX7219
* Особенности схемотехники
* Энергопотребление
* Переключение режимов
* Работа с памятью
* Руководство пользователя
* Полезные ссылки

**Введение**

Данная работа выполняется в рамках Военно-Научного Общества(ВНО) и предназначена принести пользу военному факультету БГУИР, в частности студентам и преподавателям кафедры связи при изучении цифровой тропосферной радиостанции Р-423-1(далее СТАНЦИЯ)

Цель работы: “оживить” стенд со структурной схемой СТАНЦИИ, визуально отображая тракты прохождения сигналов, режим работы выбирается пользователем.

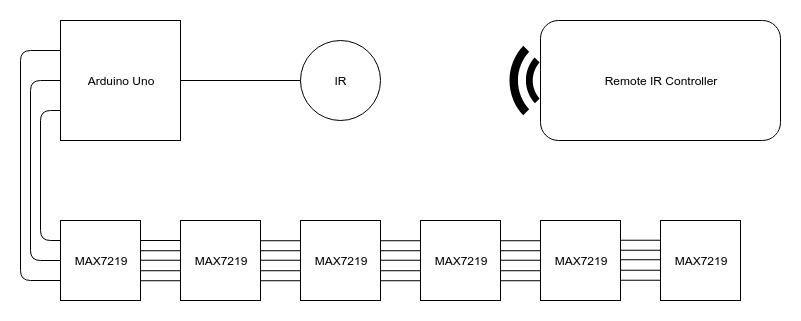
Пользователь – студент либо преподаватель, изучающий схему работы СТАНЦИИ при помощи стенда.

Разработчик – студенты, разрабатывающие и сопровождающие проект.

Требования к Разработчику:

* умение паять
* знание принципов работы микроконтроллеров(AVR) или языка С и платформы Arduino
* владение основами цифровой схемотехники

Разработчики и их контакты указаны вначале, исходный код можно найти в github-репозитории по ссылке [4].

 Рис.2 Схема подключения устройств

Также хотелось бы передать привет поколению казахов(Азамат Алиазов, АКА Баур, он же Искандер) за вложенную неаккуратность в схемотехнику и “качественное” общение, а также спасибо за действительно неплохую идею.

**Оборудование**

В качестве основного используемого обурудования были использованы:

* стенд со структурной схемой СТАНЦИИ;
* отладочная плата Arduino Uno с МК ATmega328;
* датчик инфракрасного излучения;
* пульт с инфракрасным светодиодом;
* контроллер светодиодной матрицы MAX7219;
* цветные маломощные светодиоды;
* провода;
* питающий(отладочный) кабель(>=2м).

Вспомогательное оборудование, необходимое для выполнения работы:

* дрель, сверла;
* клей(клеевый пистолет);
* паяльник, канифоль, припой;
* нож;
* ноутбук;
* и др. по необходимости.

**Структура кода**

В папке с исходными кодами есть следующие проекты:

* debug\_test/checkModulesInMemory — проверка всех модулей при отсутствии доступа к стенду(отладочный проект, юзать дома при правках ROM)
* debug\_test/checkModulesStand — проверка всех модулей на стенде(проект для теста после внесения изменений в «железе» и правок ROM)
* debug\_test/led\_index — отладочный проект для проверки матриц или отдельных светодиодов на них.
* debug\_test/IRreceiver — отладочный проект для проверки соответствия кодов нажатых клавиш пульта их кодам в исходниках
* debug\_test/IRtestCodes — отладочный проект для получения кодов нажатых клавиш. Используется при смене пульта для изменения данных кодов в основном проекте
* **writeROM** — основной проект для перезаписи ROM новыми значениями
* **military training stand/viewCompact** — основной проект, исполняемый микроконтроллером после обновления ROM

1. viewCompact.ino — main-файл, запускает опрос режима и его запуск, а также прерывание при нажатии сброса режима с запуском watchdog timer`a
2. ir\_receiver.h — код обработки нажатий клавиш и обновления режимов
3. modules.h — переменные, содержащие адрес модулей в ROM
4. path.h — переменные трактов(массивы), содержащие модули, тракты сохраняются в flash-память
5. stand.h — подключение трактов и алгоритм их включения

**Жирным шрифтом** выделены проекты которые должны быть «залиты» в память МК в обязательном порядке для работы стенда и любые правки вносятся в них. Остальные проекты являются тестовыми/отладочными

**ИК-приемник и пульт управления**

Для приема сигнала в проекте используется ИК-диапазон и способный его воспринимать ИК-датчик типа vs1838b(рис. 3), имеющий два контакта питания и один сигнальный. Он выведен с левой лицевой стороны стенда. При возникновении неполадок связанных с датчиком, он может быть заменен на любой подобный.

Для работы с датчиком наиболее удобно использовать библиотеку <IRremote.h>. В ней от пользователя скрыто использование аппаратного прерывания на используемом порту для записи пришедшего на ИК-датчик значения. Данную библиотеку можно найти по ссылкам [1] и [2]. обучающие материалы можно найти на просторах Интернета.



Рис. 3 ИК-датчик vs1838b Рис. 4 Рис. 5

В качестве пульта изначально выступал мини-пульт для музыкального проигрывателя(рис. 4), но ввиду малой дистанции действия(до ~2 м), недолговечности службы(отказ после множественных нажатий при тестах и работе), был заменен на ТВ-пульт(рис. 5). Было замечено, что мощность ТВ пульта выше, поэтому дальность его действия увеличивается до нескольких метров.

При смене пульта необходимо заменить коды в основном проекте (файл military training stand/viewCompact/ir\_receiver.h). Предварительно узнать коды можно с помощью отладочного проекта IrtestCodes и проверить работоспособность в проекте IRreceiver

**Контроллер светодиодной матрицы MAX7219**

Во многих проектах для визуализации используются светодиодные матрицы. Для управления ими используются контроллеры светодиодных матриц и семисегментных индикаторов MAX7219. Контроллер представляет собой чип, размещенный на своей отладочной плате. Благодаря этому матрицы удобно объединять в каскад(Рис. 6).

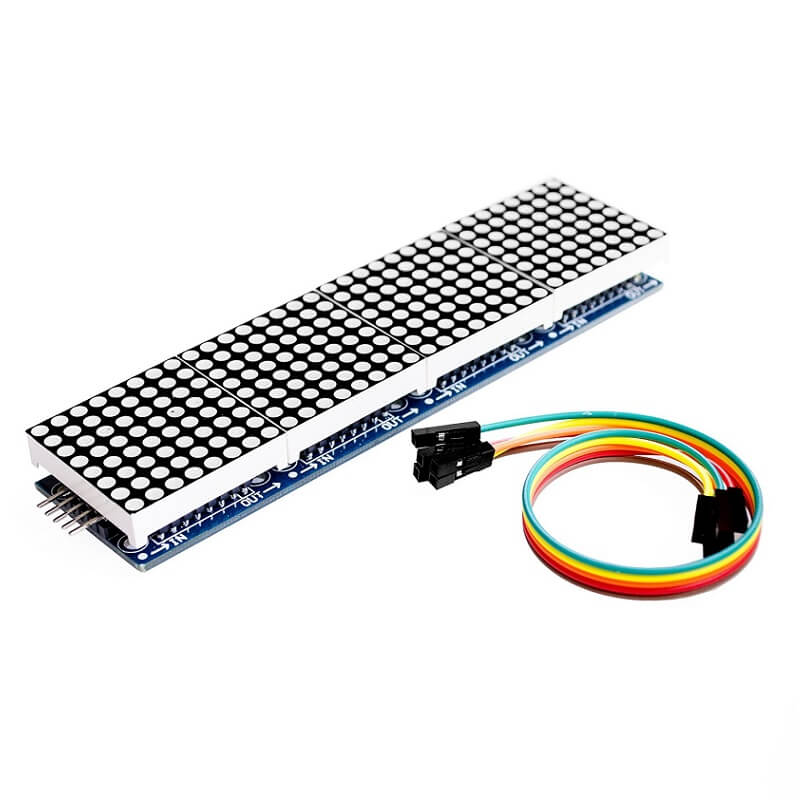


Рис.6

В данном проекте используется та же идея, но вместо миниатюрной матрицы выступает самодельная матрица из светодиодов.

Дла работы со светодиодными матрицами можно использовать библиотеки <MaxMatrix>(используется в проекте), <MD\_MAX72XX>, <LedControl> и другие. Все из них пригодны для использования, но в каждой разный принцип каскадирования матриц, что необходимо учитывать при индексации светодиодов в модулях, поэтому желательно использовать <MaxMatrix>(изначальную).

**Особенности схемотехники**

На задней стороне стенда можно увидеть внушительное количество проводов. Вся площадь поделена на зоны матриц, в каждой из которых установлен контроллер с платой, от которой провода идут на сетку. Часть сетки это условные “столбцы”(+), часть - “строки”(-). Действуют как показано на рис.7.

В конкретно нашей реализации в качестве столбцов(+) выступают гнезда 3,4,6,10,11,13,15,16; а в качестве строк(-): 1,2,5,7,8,9,12,14. Гнезда считались начиная от ключа контроллера(точка в углу).

Учитывая то, что светодиодные матрицы бывают с общим катодом/анодом, что в проекте используются разные светодиоды(по электрическим характеристикам), в случае проблем придется импровизировать(возможно, включать светодиоды обратной полярностью) и наблюдать прочую магию вне Хогвартса.

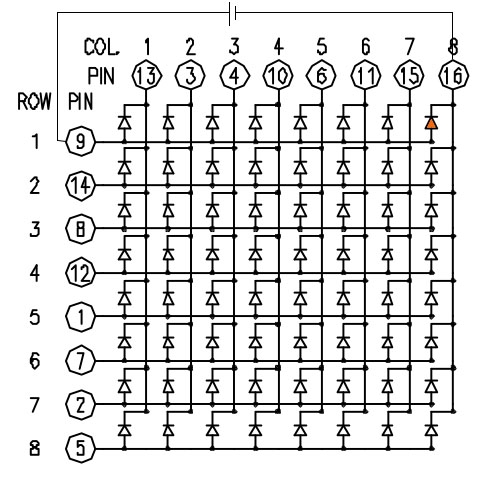


Рис. 7 Схема светодиодной матрицы

**Энергопотребление**

В проекте применяются различные светодиоды различной мощности. Ток одного светодиода не должен превышать 20-30 мА. Выходной ток матрицы ~300 мА. Таким образом, по расчетам СТЕНД может потреблять более 1А, однако пркатика показывает, что питающий ток не превышает 0,5А. Это зависит от количества светодиодов на тракт и выставленной программно(+)/аппаратно яркости. Точные данные по электрическим характеристикам контроллеров матриц можно посмотреть в даташите(ссылка [3]) или измерить мультиметром.

**Переключение режимов**

Для того, чтобы иметь возможность во время отображения тракта переключить режим(например, при неправильном выборе режима), было написано стандартное программное прерывание(ISR), работающее по таймеру. Он проверяет нажатие кнопки смены режима(Power). Если она была нажата, запускается watchdog timer(WDT), который делает reset МК. Весь проект запускается заново и пользователь может выбрать новый режим

Данный подход считается стандартным и в данной ситуации наиболее оптимальным, так как режимы не следует считывать и переключать во время выполнения предыдущего режима, иначе будет наблюдаться вложенность выполнения кода(трактов), следовательно будет забиваться память и стек вызовов.

Решения: 1) считывать и переключать режимы после прохождения тракта – циклическое выполнение алгоритма. 2) делать то же самое но в любой момент времени по прерыванию. Поначалу применялся способ 1, но было решено, что 2 лучше

**Работа с памятью**

Физически светодиоды на стенде могут перепаиваться, меняться местами, могут быть добавлены. Поэтому нет смысла их группировать аппаратно, кроме как в матричные зоны для удобства(их 6 штук).

Задача упорядочивания(индексация) должна проводиться в коде. Поэтому каждый блок и линия на СТЕНДЕ названы МОДУЛЕМ. МОДУЛЬ имеет следующий вид:

B10000100, 47,5, 41,4, 41,2, 47,4, // block\_1\_D\_00\_39

В первом числе(B10000100) в бинарном виде первыми двумя битами закодирован вид МОДУЛЯ(10 – блок, 01 – линия), что позволяет применять к МОДУЛЯМ разный подход(например задержки), остальные 6 бит хранят количество светодиодов в модуле для считывания правильного количества из ROM. Далее перечисляются координаты(x,y) всех светодиодов СТЕНДА. Эти данные описаны в прокте **writeROM** и должны быть предварительно записаны в ROM(для быстрого вызова и экономии памяти). В библиотеке матрицы объединяются в один последовательный каскад(например, две матрицы 8х8 = один объект 16х8).

**military training stand/viewCompact**/modules.h – это файл с переменными(RAM), содержащими адреса всех МОДУЛЕЙ(ROM).

**military training stand/viewCompact/**path.h – файл хранящий тракты в виде массивов МОДУЛЕЙ. При прошивке микроконтроллера тракты записываются в flash-память, тем самым не занимая место в RAM(см. PROGMEM, <avr/pgmspace.h>). При редактировании можно менять местами модули и добавлять/удалять их, однако не стоит забывать, что в начале массива стоит число модулей в тракте(на 1 меньше размера самого массива).

Светодиоды на блоках режимов работы не относятся к МОДУЛЯМ и записаны просто в начале ROM, откуда в том же порядке считываются.

\*\* Подход с записью части данных в ROM, части в flash и испольнением остального кода в RAM вызван ограниченностью ресурсов МК ATmega328(1 КБ ROM, 2 КБ RAM и 32 КБ flash памяти). Так, изначально был написан проект military training stand/view(ныне недействителен), где каждый светодиод являлся структурой, они объединялись в структуры МОДУЛЕЙ, а те в массивы трактов. Простое управление, построенное на структурах и указателях, не занимающее ROM и flash не компилировалось под данный контроллер, т.к. все хранилось в RAM и ело 3,7 КБ. Было решено перенести статические, не изменяющиеся данные в ROM(данные МОДУЛЕЙ), а что не вместилось(тракты) в flash. Вообще, все можно было поместить в flash, однако есть особенность flash – он считывается страницами, что замедляет работу с ним и сильнее грузит контроллер. Поэтому туда записаны крупные данные трактов.

Было замечено, что стоит экономить на каждом байте данных, потому стали применяться типы uint8\_t и приведения, где это возможно, а модули содержат свое побитное описание.

**Руководство пользователя**

Для работы обучающего стенда необходимо запитать его. Для этого достаточно взять блок питания на >0,5А, кабель USB Type B – USB Type A(как для принтеров) и подключить к Arduino.

Управлять стендом при помощи пульта. В текущей версии используются 4 кнопки:

* “вверх”/“вниз”(над/под “ок”) - выбор предыдущего/следующего режима
* “OK” - выбор режима для отображения
* “Power”(вкл/выкл) – сброс текущего режима и выбор нового

**Полезные ссылки**

1)https://www.arduinolibraries.info/libraries/i-rremote

2)https://github.com/z3t0/Arduino-IRremote

3)http://pdf1.alldatasheet.com/datasheet-pdf/view/73745/MAXIM/MAX7219.html

4)<https://github.com/ivanka6342/ArduinoProjects/tree/master/military%20training%20stand>