

Лабораторная работа 2.2

Операционная система реального времени FreeRTOS — работа с диспетчером задач.

Цель работы: Знакомство с OCPB FreeRTOS, работа с микроконтроллерами в ОС Linux / Windows.

Аппаратное и программное обеспечение: PC, ОС Linux / Windows с установленным компилятором GCC, GCC-AVR, программа для загрузки кода AVRDUDE (для ОС GNU/Linux Ubuntu устанавливается пакет arduino, для ОС Windows необходимо установить программу WinAVR), подключенное к USB порту устройство на базе микроконтроллера.

Вариант выполнения работы №1:

1. Работа выполняется на локальной машине с ОС Linux и установленным toolchain для микроконтроллеров AVR. Перед выполнением, при необходимости, создайте рабочий каталог с именем выполняющих работу (в терминале Linux команда **mkdir Ivanov_Petrova**).
2. Необходимо загрузить архив freertos.tar и распаковать его в рабочий каталог. Перейти в каталог freertos/FreeRTOS/Demo/lab22 и собрать «прошивку» командой **make clean && make**

Далее необходимо запрограммировать контроллер командой **make program**

После процесса программирования на устройстве должен замигать светодиод.

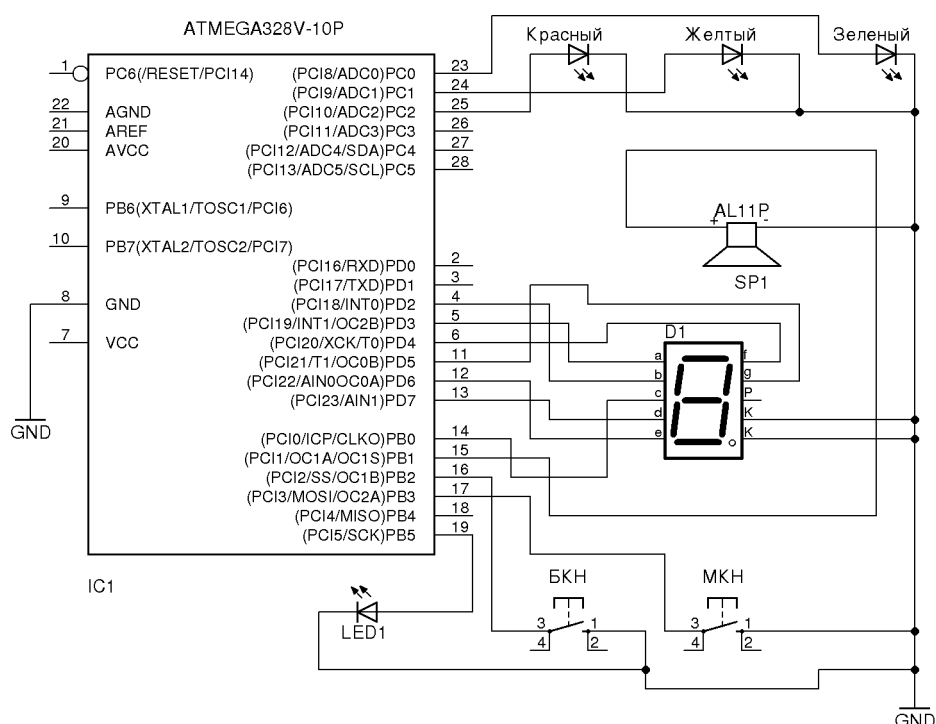


Рисунок 2.2.1 Схема подключения периферийных устройств к микроконтроллеру .

3. Необходимо составить программу, которая должна выводить на сегментный индикатор номера выполняющих работу (согласно списку группы). Номера выводятся

следующим образом: через секунду после запуска программы выводится номер первого студента (например для первого по списку выводим «0» в течении 0,5 с, затем пауза 0,5 с, затем «1» в течении 0,5 с). Далее через секундную паузу аналогично выводится номер второго выполняющего работу и так далее. Для индикации каждого номера необходимо создать отдельную задачу (API функция `xTaskCreate()`). Для временных задержек необходимо использовать способность задач изменять свое состояние (состояние Blocked, функция `vTaskDelayUntil()`).

4. Дополнить программу задачей ручного переключения (последовательного) номеров с помощью кнопки.
5. Добавить в программу задачу звуковой индикации нажатия кнопок.
6. Добавить вывод номеров в порт UART микроконтроллера. Наблюдать вывод можно в эмуляторе терминала (Linux), введя команду:

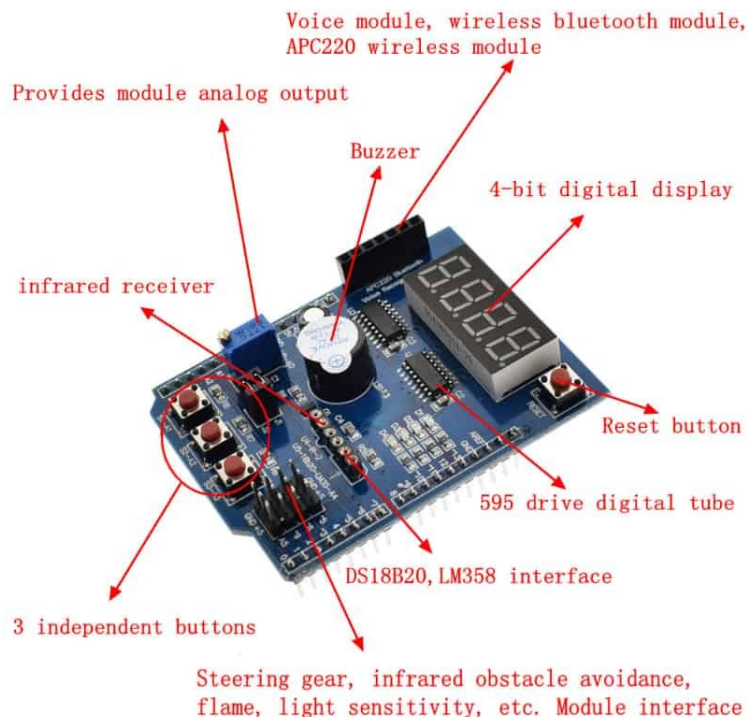
screen /dev/ttyUSB0 9600

завершить работу сессии screen можно комбинаций клавиш «Ctrl+a k y». Для Windows возможно использовать программу Hyper Terminal или аналогичную.

7. Добавить задачу, обеспечивающую ручное переключение номеров с помощью команд, передаваемых в порт UART микроконтроллера.

Вариант выполнения работы №2:

1. Работа выполняется на локальной машине с ОС Linux / Windows и установленным toolchain для микроконтроллеров AVR. В качестве аппаратной части используется плата Arduino UNO с подключенным multi-function shield (см рисунок 2.2.2) Перед выполнением, при необходимости, создайте рабочий каталог с именем выполняющих работу (в терминале Linux команда **mkdir Ivanov_Petrova**).



- Необходимо загрузить архив freertos.tar и распаковать его в рабочий каталог. Перейти в каталог freertos/FreeRTOS/Demo/lab22 и собрать «прошивку» командой (в терминале Linux):
make clean && make
- Исходя из принципиальных схем подключения индикатора, кнопок, светодиодов и звукового излучателя (рисунки 2.2.3 — 2.2.4) выполнить пункты 3 — 7 варианта №1 данной работы.

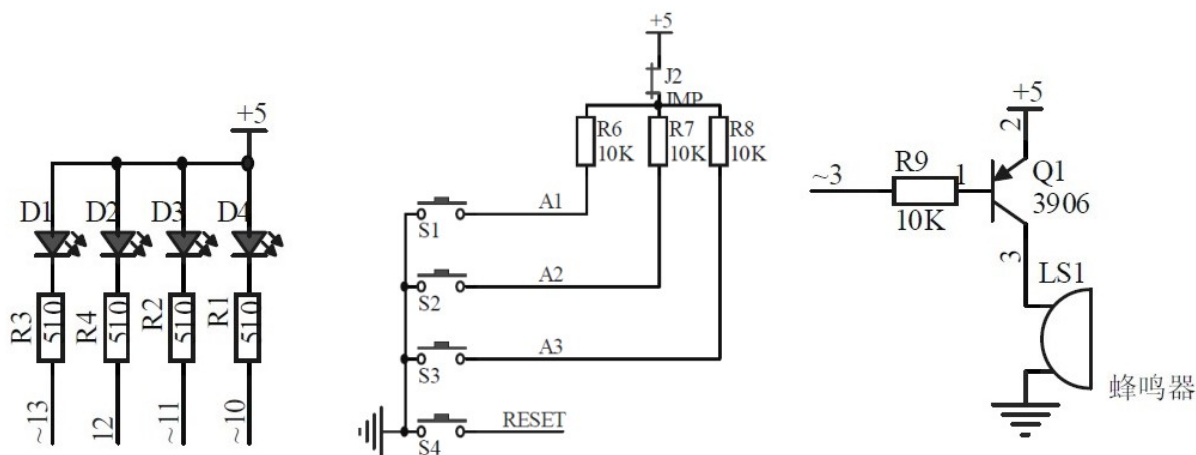


Рисунок 2.2.3 Multi-function shield - принципиальная схема подключения светодиодов, кнопок и звукового излучателя

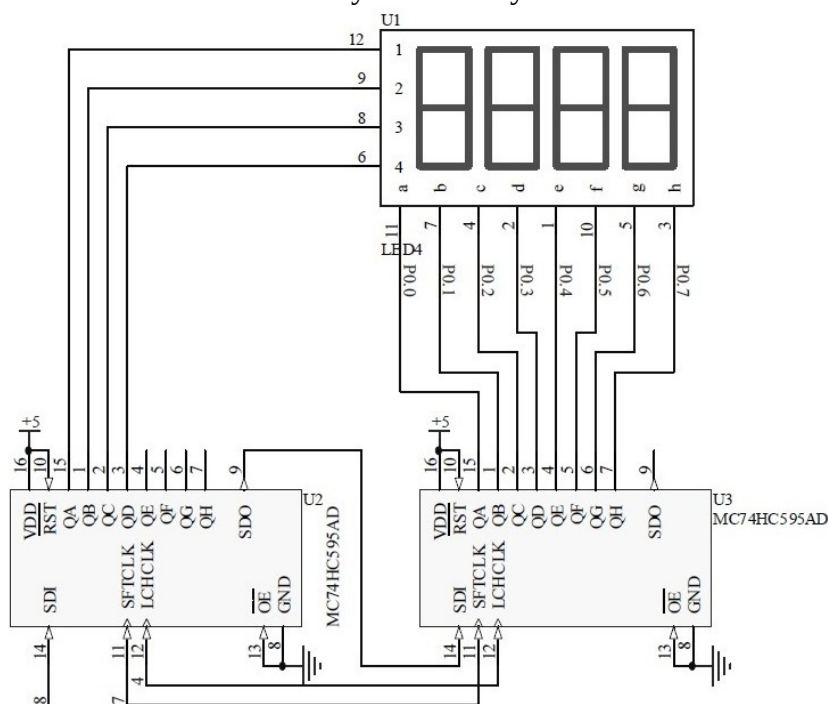


Рисунок 2.2.4 Multi-function shield - принципиальная схема подключения сегментного индикатора

Источники информации:

<http://microsin.ru/content/view/1304/44/>

www.freertos.org

<http://www.google.ru>