**Аппаратная часть**

В качестве аппаратной части проекта используется программатор USB ISP version 2.0. Для возможности перепрошивки программатора и использования его в качестве генератора сигналов необходимо замкнуть перемычку согласно рисунку 1. Это позволит микроконтроллеру входить в режим программирования и выведет ножку RST на разъем программатора.

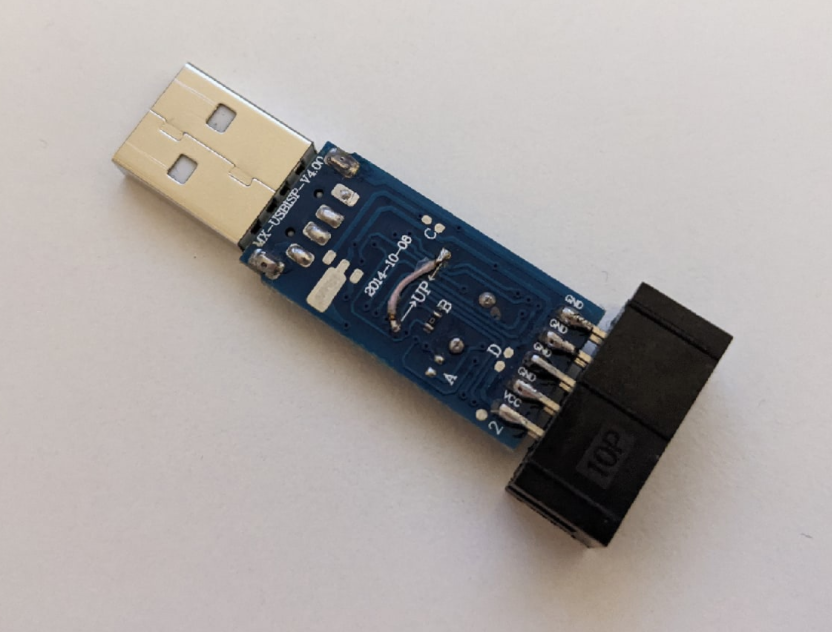


Рисунок 1 – Перемычка

Для разработки ПО была выбрана среда Microchip Studio так как на программаторе установлен микроконтроллер ATmega88.

Так как на программаторе отсутствует аппаратная поддержка USB, а также отсутствует какой-либо дополнительный микроконтроллер-конвертор, то необходимо реализовать программную эмуляцию интерфейса USB.

На текущий момент оптимальным решением будет использовать проект V-USB (<https://www.obdev.at/products/vusb/index.html>) предоставляющий возможность реализовать программную эмуляцию интерфейса USB 1.1 low-speed. Этот проект широко распространен и хорошо задокументирован.

Далее рассмотрим принцип, по которому будет осуществляться связь и передача данных с МК на ПК и наоборот.

Создаваемое нами устройство будет классифицироваться в системе как устройство класса HID. Данный тип устройств обладает весомым преимуществом – драйверы, необходимы для работы, уже предустановлены в OS Windows, таким образом, нам нет необходимости писать собственный драйвер.

Для идентификации устройства системой (при подключении), микроконтроллер должен передать компьютеру USB HID report descriptor, в котором записаны параметры подключаемого устройства, так Windows сможет определить какой именно драйвер необходимо использовать для данного устройства.

Подробная инструкция по настройке USB HID report descriptor находится по ссылке: <https://eleccelerator.com/tutorial-about-usb-hid-report-descriptors/>. Для формирования дескриптора используется специальная программа HID Descriptor Tool, размещенная по ссылке <https://www.usb.org/document-library/hid-descriptor-tool>.

Но прежде чем формировать дескриптор, необходимо определиться с данными, которые планируется передавать. Так как передача данных осуществляется пакетами фиксированного размера, то нужно придумать методику шифрования данных в пакеты. Данные нужно компоновать так, чтобы они занимали минимальный размер и при этом были хорошо структурированы. Пример такой компоновки находится в файле bits.docx.

Теперь, когда длина и количество пакетов определены, можно сформировать дескриптор. Пример дескриптора с описанием изображен на рисунке 2.

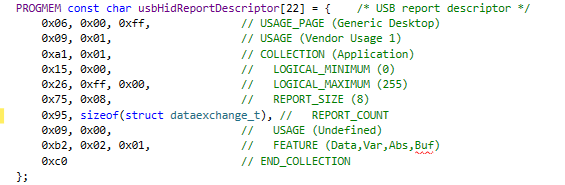


Рисунок 2 – Дескриптор

Так как связь ПК с микроконтроллером происходит за счет передачи сигналов по линиям D+ и D- порта, то данные линии необходимо подключить к соответствующим ножкам прерываний INT0 и INT1 чтобы обеспечить максимальный приоритет данных сигналов. Так как на программаторе данные линии уже соединены, можно предположить, что передача данных осуществлялась по тому же принципу.

Теперь необходимо настроить параметры файла usbconfig.h из проекта V-USB. Нас интересуют только параметры:

USB\_CFG\_IOPORTNAME – буква порта прерываний, в данном случае D

USB\_CFG\_DMINUS\_BIT – номер порта D-, в нашем случае 3

USB\_CFG\_DPLUS\_BIT – номер порта D-, в нашем случае 2

USB\_CFG\_VENDOR\_NAME – имя разработчика в формате ‘a’, ‘b’, ‘c’

USB\_CFG\_VENDOR\_NAME\_LEN – число знаков имени разработчика

USB\_CFG\_DEVICE\_NAME – название устройства в том же формате

USB\_CFG\_DEVICE\_NAME\_LEN – число знаков названия устройства

USB\_CFG\_INTERFACE\_CLASS – номер класса устройства, для HID это 3

В дальнейшем имя разработчика и название устройства будет использоваться для определения генератора импульсов среди остальных HID устройств, подключенных к компьютеру.

После написания программы, ее необходимо загрузить в МК, для этого использовался программатор AVR ISP MKII. Подключение осуществлялось к выводам, расположенным в торце корпуса генератора сигналов.

**Программная часть**

Для осуществления связи ПК с МК используется библиотека hidlabrary.h, подключаемая в проект в RAD Studio, также необходимо подключить библиотеку usbconfig.h (из нее приложение узнает название нашего устройства). Далее необходимо повторить описание структуры передаваемых и принимаемых данных и прописать шифрование, которое используется в микроконтроллере (согласно файлу bits.docx).

P.S.

Та как библиотека старая, то возникают конфликты в типах данных, так как в разных версиях RAD Studio использовались разные типы данных. Основная часть ошибок исправлена, но приложение все еще не может получить названия подключенных устройств. Проект запускается только в версии RAD Studio XE8

Шаблон проекта находится на гугл диске.