**3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНСУЛИНОВОЙ ПОМПЫ.**

На данный момент использование инсулиновой помпы является единственным доступным способом вести нормальный образ жизни для людей, страдающих сахарным диабетом первого типа, а также для тех пациентов, кому не помогают лекарства. Поэтому разработка инсулиновой помпы является важной задачей, а само программное обеспечение, осуществляющее ее работу, требует особого внимания.

В данной работе была разработана модель инсулиновой помпы на основе микроконтроллера Atmega162. Программирование микроконтроллера, реализующего синхронную работу всей системы написано на языке программирования С, одном из самых приближенных к машинному коду. Код программы написан там же, где и собрана схема прибора, в программе Proteus с компилятором WinAVR.

Для отлаженной работы устройства, необходимо описать работу всех четырех модулей:

1. Кнопки являются элементом блока управления системой.
2. Светодиоды служат индикацией рабочего режима и функционала кнопок.
3. LCD дисплей, на нем высвечиваются значения дозы инсулина, скорости двигателя и направление его движения.
4. Шаговый двигатель, осуществляющий роль дозатора.

**3.1. Работа кнопок и светодиодов.**

Вся работа кнопок прописана в прерывании TIMER0 COMP Timer. Для его инициализации подключены библиотека avr/interrupt.h.

Работа программы заключается в проверке состояния кнопок и выполнении соответствующих действий.

Всего 4 кнопки, одна из которых всегда нажата, отвечает за режим работы. Проверка состояния каждой происходит с помощью оператора сравнения if. Действия по изменению значения переменных, вроде увеличения скорости вращения шагового двигателя, происходят, если значение соответствующего пина равно нулю, тогда как вывод значений осуществляется, пока кнопка отжата. Это связано с тем, что проверка условий происходит с каждым циклом, поэтому запись функций изменения значений статическому состоянию приведет к произвольному изменению всех переменных.

Пример кода приведен на рисунке 1.

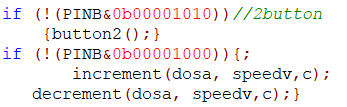


Рисунок 1. Часть кода обработки прерываний с кнопок.

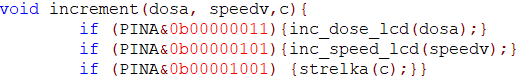


Рисунок 2. Функция инкремента

До нажатия кнопки значение ее пина равно 1, и на дисплее показы значения дозы, скорости и направления движения двигателя, описанные в соответствующих функциях (рис.2), ссылающихся на значения измененные при исполнении функции button2 при нажатии кнопки. Сама функция представлена на рисунке 3.

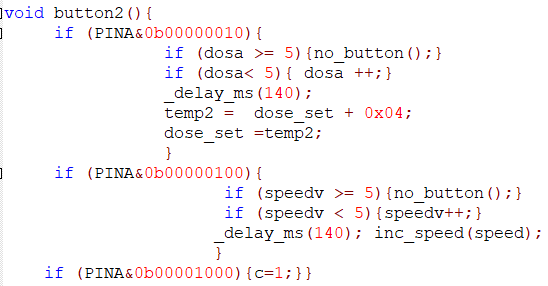


Рисунок 3. Функция button2.

Поскольку в качестве индикаторов режима выступают светодиоды, для его определения достаточно узнать, какой из них горит. Для реализации устройства было принято значение максимальной дозы равное 5 ед. инсулина. В данной функции происходит сравнение текущего значения dose\_set с максимальным и ее увеличение, при меньшем значении. Аналогично со значением скорости.

Значение PORTA, отвечающего за работу светодиодов, изменяется при нажатии четвертой кнопки (рис.4).

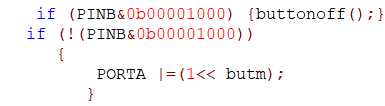


Рисунок 4. Работа четвертой кнопки.

Функция buttonoff осуществляет проверку значения переменной butm и ее увеличения на единицу. При нажатии кнопки происходит смещение единицы в значении порта влево и зажигание соответствующего диода.

**3.2. LCD дисплей**

Для программирования дисплея использовалась библиотека hd44780.h и hd44780\_settings.h, в котором определены его настройки. Его работа также определяется работой кнопок. При каждом изменении дозы, скорости или направления движения двигателя изменяются их соответствующие значения на дисплее. Слова прописаны в отдельной функции word и не зависят ни от каких параметров, всегда присутствуют на экране, выведены с помощью функции lcd\_puts, числа записываются в одну и ту же переменную и выводятся на экран на одном и том же месте поверх предыдущей, со стрелкой ситуация аналогична, используемая функция – lcd\_putc.

Инициализация экрана происходит с помощью функции lcd\_init.

**3.3. Вращение двигателя**

Вращение двигателя вызывается первой кнопкой (ввод) и осуществляется за счет смены положения, каждое из которых представляет двоичное число и прописано в функции pos. В зависимости от выбранного направления эти позиции меняются по порядку и приводят шаговый двигатель в работу. Скорость работы определяется значением переменной speed, а количество оборотов величиной дозы, значением переменной dose\_set. Часть программы, осуществляющей его работу приведена на рисунке 5.

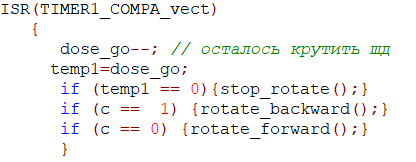


Рисунок 5. Прерывание таймера по кручению мотора.

Переменная с определяет направление движения двигателя, принимает только два значения, 0 и 1. Функция rotate\_backward осуществляет поворот двигателя против часовой стрелки, путем перебора значений положения, записываемых в соответствующий порт. Начало движения осуществляется функцией start, находящейся в блоке управления сигналов с кнопками. Она разрешает прерывание таймера и включает красный светодиод, означающий ввод инсулина. Двигатель крутится до тех пор, пока значение переменной dose\_go не уменьшится до нуля, изначально ее величина равна величине выбранной дозы. Остановка двигателя происходит за счет запрета прерывания таймера в функции stop\_rotate.

**3.4. Работа устройства**

Для того, чтобы проверить работу программы достаточно запустить симуляцию. Прибор способен отмерять как представленные стандартные дозы, так и выполнить ввод согласно предоставленному документу с их значениями. Однако, чтобы использовать этот файл, его необходимо перевести в массив, скомпилировав в другой программе, в моем случае Visual Studio C++, так как в языке С нет возможности работать с текстовыми файлами. Исходный файл в текстовом формате загружается и компилируется, результат записывается в файл с расширением .h, для дальнейшего чтения в исходной программе. Код преобразования приведен на рисунке 6.



Рисунок 6.

Исходный файл sign.txt находится в переменной stream, а итоговый mys.h в stream1.

Режим ввода доз согласно файлу сопровождается миганием зеленого и свечением красного светодиода, начинается с нажатием кнопки старт. Чтобы убедиться, что введены именно те дозы, значения текущей записывается в массив. Значения элементов массивов приведены на рисунке 7.

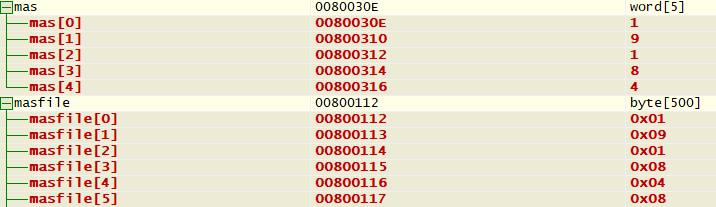


Рисунок 7.

Mas – выходной массив, misfile – исходные данные файла. Приведенные значения обоих массивов совпадают, значит шаговый двигатель работает строго по программе.

**ВЫВОДЫ ПО ГЛАВЕ**

В результате проделанной работы был написан код на языке С, реализующий работу прибора. В ходе его написания были опробованы возможности языка и выявлены недостатки, доставляющие некоторые проблемы при реализации задуманной программы. В итоге, полученная программа приводит в действие инсулиновую помпу, позволяет выполнить все задуманные функции прибора.

**Список использованных источников**

1. Анисимов А.А. Медицинские микропроцессорные системы: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2019. 83 с.
2. Управление шаговым двигателем с помощью AVR микроконтроллера ATmega8 // URL: <https://avrlab.com/node/60>.
3. Atmel AVR ATmega162 datasheet / Atmel Corporation, 2013. 324c