

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН



Институт промышленной автоматизации и цифровизации им. А.Буркитбаева

Кафедра «Робототехники и технических средств автоматики»

Нурланова Арайлым Нурлановна

Медицинский робот для умной сортировки
и дозировки лекарственных средств

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

Специальность 5В071600 – Приборостроение

Алматы 2020

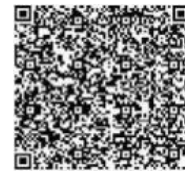
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН



Институт промышленной автоматизации и цифровизации им. А.Буркитбаева

Кафедра «Робототехники и технических средств автоматики»

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой РТиТСА
кандидат технических наук



К. А. Ожикенов
«24» мая 2020 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

На тему: «Медицинский робот для умной сортировки
и дозирования лекарственных средств»

по специальности 5В071600 – Приборостроение

Выполнила Нурланова Арайлым
Научный руководитель
Магистр технических наук



Баянбай Н.А.
«24» мая 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН



Институт промышленной автоматизации и цифровизации им. А.Буркитбаева

Кафедра «Робототехники и технических средств автоматики»

5B071600 – Приборостроение

УТВЕРЖДАЮ
Заведующий кафедрой РТиТСА
кандидат технических наук



К.А.Ожикенов
«27» января 2020 г.

ЗАДАНИЕ
на выполнение дипломного проекта

Обучающейся Нурлановой Арайлым Нурлановне
Тема: Медицинский робот для умной сортировки
и дозировки лекарственных средств

Утверждена приказом Ректора Университета №726-б от "27" января 2020 г.
Срок сдачи законченной работы «15» мая 2020 г.

Исходные данные к дипломному проекту: Arduino IDE, датчик LIDAR

Перечень подлежащих разработке в дипломном проекте вопросов: а)
построили алгоритм работы программного обеспечения;

б) разработали электрическую схему «Медицинского робота-дозатора»;

в) отстроили 3Д-модель «Медицинского умного робота-дозатора» в
программном обеспечении SolidWorks.

Перечень графического материала (с точным указанием обязательных
чертежей):

Представлены 14 слайдов презентации работы

Рекомендуемая основная литература: из 29 наименований 26

ГРАФИК

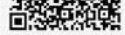
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Технологическая часть	22.01 – 15.02.2020 г.	Выполнено
Программная часть	15.03 – 20.04.2020 г.	Выполнено

Подписи
консультантов и нормоконтролера на законченный проект с указанием
относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Нормоконтролер	Ж.С. Бигалиева, магистр технических наук, лектор	24.05.2020 г.	



Научный руководитель  Баянбай Н.А.

Задание принял к исполнению

обучающийся

Нурланова А. Н

Дата «27» января 2020 г.

АҢДАТПА

Дипломдық жоба медицина саласындағы дәрі-дәрмектерді қабылдауды шешуге арналған медициналық таблетка диспенсері роботының тұжырымдамасына, дизайнына және прототипіне арналған, өйткені ол планшеттерді, сонымен қатар көптеген басқа жетілдірілген функцияларды өздігінен сұрыптау функциясына ие, және бұл робот ауруханалардан емес үйлерде қолдануға арналған қарттар. Зерттеулер көрсеткендей, медициналық қателіктердің үлкен пайызы пациенттер мен қарттардың дәрі-дәрмектеріне жауапберетін адамдар дәрі-дәрмектердің үлкен мөлшерін сұрыптау мен айналысуы керек екендігіне байланысты.

Жобаның мақсаты – бұл ауруды жою, медициналық қызметкерлерге немесе клиенттерге планшеттердің санын және қабылдау уақытын анықтауға мүмкіндік беретін робот бағдарламаланған. Бұл жұмыста барлық қажетіліктері зерттеліп, қазіргі кездегі проблема үшін заманауи шешімдер ұсынылады.

АННОТАЦИЯ

Эта дипломная работа посвящена концепции, дизайну и созданию прототипа медицинского робота дозатора для таблеток, предназначенного для решения приема препаратов в медицинской области, поскольку он обладает функцией самостоятельно сортировать таблетки, а также многими другими расширенными функциями, причем этот робот предназначен для использования в больницах или домах престарелых. Исследования показали, что большой процент медицинских ошибок происходит из-за того, что люди, отвечающие за лекарства пациентов и пожилых людей, должны иметь дело с сортировкой огромного количества лекарственных препаратов.

Целью проекта является устранить этот недуг. Данный робот программируется, что позволяет медицинским работникам или клиентам определять количество таблеток и время нужного приема. В данной работе изучены все нюансы, а также предложены современные решения такой актуальной на данное время проблемы

This thesis is dedicated to the concept, design and prototype of a medical pill dispenser robot designed to solve taking medications in the medical field, since it has the function of independently sorting tablets, as well as many other advanced functions, and this robot is designed for use in hospitals or homes the elderly. Studies have shown that a large percentage of medical errors are because people responsible for the medicines of patients and the elderly must deal with sorting a huge amount of drugs.

The aim of the project is to eliminate this ailment. This robot is programmed, which allows medical workers or clients to determine the number of tablets and the time of the necessary intake. In this work, all the nuances are studied and modern solutions are proposed for such a current problem.

СОДЕРЖАНИЕ

Введение	7
1 Обзор технологических проблем	8
1.1 Обзор существующих прототипов	10
1.2 Исследование интеллектуальных систем «Медицинских роботов-дозаторов»	12
2 Программное обеспечение	14
2.1 Разработка алгоритма работы	14
2.2 Алгоритм программного обеспечения	16
2.3 Структурная схема	17
2.4 Характеристика используемых деталей	19
3 Разработка 3Д модели	21
3.1 Конструкция модуля дозирования таблеток	21
3.2 Конструкция механизма дозирования таблеток	22
3.3 Обзор этапа сборки 3Д модели	24
3.4 Реализация 3Д модели	27
Заключение	32
Список использованной литературы	33
Приложение А	34

ВВЕДЕНИЕ

Есть много людей, которые нуждаются в постоянной помощи, например, наши пожилые люди, члены семьи, те, у кого есть особые потребности. Пожилые люди более подвержены влиянию времени приема определенного препарата, чем другие, для того чтобы предотвратить любую дисфункцию или болезнь, время является обязательным условием.

Но поскольку со старением приходят плохое зрение и плохая память, что делать, если у пациента есть слабоумие, как у Альцгеймера. Некоторые люди могут забыть принять лекарства в нужное время и могут забыть лекарства, которые они должны принимать. Для того чтобы исключить факторы всегда необходимого наблюдения, такие как медсестры или риск пропущенной дозы, мы должны были найти простое, портативное и эффективное решение. Роботы-дозаторы для лекарств уже существуют, но большинство из них либо имеют ограниченное использование, не подходят для пожилых людей, либо даже имеют большой размер, что делает его непригодным для того, чтобы взять его с собой куда-либо.

Для того чтобы сделать действительно полезный умный дот, он должен был быть легко интегрирован с новейшими широкими интеллектуальными технологиями. Хотя в то же время он был пригоден для старших и их ограниченных знаний и опыта, чтобы реализовать простоту использования. Размер и портативность также были важным фактом, который мы должны были иметь в виду.

Для того чтобы называться умным, он подключен через беспроводную сеть, что позволяет ему быть подключенным к Интернету для будущих приложений и интеграции, а также имеет широкий спектр WiFi вместо Bluetooth или любого другого полевого соединения, и стирает необходимость в каких-либо проводах или проводном соединении, которое обеспечивает мобильность в первую очередь. Через ту же сеть он подключен к мобильному телефону, который с его помощью вы можете установить временной интервал для приема дозы, а также во многом уведомляет вас, когда приходит время приема дозы. Кроме того, мы добавили зуммер со светодиодом, чтобы он выглядел как физическое предупреждение, поэтому он не оставляет вам никакого выбора, кроме как запомнить время приема таблетки и принять ее.

Целью данного исследования является создание умного робота-дозатора таблеток для медицинских напоминаний и системы мониторинга. Когда время приема таблеток установлено, точка будет напоминать клиентам или пациентам принимать таблетки, используя звук и свет. Предупреждение о том, что таблетки должны быть приняты, будет показано приложением для Android, которое находится в руках пациента. Контрастирует с обычным дозатором таблеток, которая требует, чтобы клиенты или обслуживающий персонал укладывали умного дозатора каждый день или последовательно.

1 Обзор технологических проблем

В связи с тем, что таблетки стали играть столь важную роль в повседневной жизни, в последние годы наблюдается рост числа случаев медицинской халатности, связанных с неправильным назначением пациентам лекарств, таких как случай медсестры, которая дала пациенту паралитик вместо антацида, назначенного врачом, что привело к смерти пациента. После того, как мы увидели так много таких случаев, очевидно, очень важно, чтобы правильная таблетка была принята правильным человеком в нужное время, иначе неправильный прием или вообще не прием может подвергнуть пациента нескольким опасным ситуациям, начиная от легких проблем со здоровьем и заканчивая смертью.

Другие случаи неправильного приема таблеток пациентами вызваны самими пациентами, особенно в пожилом возрасте. По мере того как люди стареют, человеческий организм имеет тенденцию к сбоям, и количество таблеток, которые средний человек должен принимать по достижении определенного возраста, значительно увеличивается, так как, согласно исследованию 2008 года, опубликованному в журнале *American Medical Association*, более 40 процентов людей в возрасте 65 лет и старше принимают пять лекарств в день. Обычно это широкий спектр различных таблеток, которые пожилой человек должен принимать в разное время. Отслеживание приема нужной таблетки в нужный момент каждый день может стать сложным опытом для пожилых людей, поскольку это не так просто, как могло бы быть для молодого человека. Этот факт легко объясняется, когда мы понимаем, что многие способности, такие как зрение, память или логические способности, имеют тенденцию уменьшаться пропорционально возрасту, как только люди вступают в старость, что затрудняет им запоминание того, какую таблетку принимать в нужное время, запоминание того, чтобы принять их или путать одну таблетку с другой, поскольку человек может быть не в состоянии отличить одну от другой из-за их уменьшенного зрения, а также сходства форм и цветов таблеток. Эта проблема, скорее всего, будет причиной для беспокойства людей, окружающих принимающего таблетки, поскольку не прием таблетки в нужное время может вызвать серьезные проблемы (такие как отторжение органа у пациента с трансплантацией органа или сердечный приступ у пациентов, страдающих тяжелыми заболеваниями сердца).

С другой стороны, некоторые проблемы, связанные с большим количеством таблеток, которые в настоящее время назначаются пациентам, встречаются в больницах или домах престарелых. В этих местах одной из главных задач является выдача своему пациенту правильных таблеток. Старение человека, сортировка и выдача таблеток каждому из пациентов иногда могут иметь высокую вероятность ошибки, когда пациент или резидент получает одну или несколько неправильных таблеток.

Наконец, существуют ситуации, когда прием неправильного количества таблеток является делом неопытности пациента и / или его невнимательности. Независимо от причины, было доказано, что существует значительный риск того, что люди в конечном итоге проглотят неправильное лекарство или дозу.

Болезнь Альцгеймера - это наиболее широко признанный тип деменции, общий термин для обозначения неудачной памяти и других интеллектуальных способностей, достаточно подлинных, чтобы вмешиваться в повседневную жизнь. Болезнь Альцгеймера составляет от 60 до 80 процентов случаев деменции. Болезнь Альцгеймера-это не обычный процесс созревания. Наиболее известным фактором риска является возрастной рост, и преобладающая часть людей с болезнью Альцгеймера-это люди в возрасте 65 лет и старше. В любом случае, болезнь Альцгеймера-это не только болезнь старшинства. Около 1 500 000 человек моложе 65 лет имеют более молодую начальную инфекцию Альцгеймера (иначе называемую ранней начинающейся болезнью Альцгеймера).

Построение "пропущенной дозы" является неотъемлемой частью многочисленных оценок приверженности к лечению. Тем не менее, лишь немногие исследования показали, как пациенты или клиницисты концептуализируют пропущенные дозы или степень согласованности или жесткости между клиницистами и пациентами. Вообще говоря, пациенты выявили более строгое значение пропущенных измерений, чем клиницисты. Пятьдесят пять процентов пациентов заподозрили, что отсрочка приема таблеток на 6 ч после утвержденного времени приема представляет собой пропущенное измерение, по корреляции, только один врач согласился с этой оценкой. Более 33% пациентов считали, что наилучшая возможная реакция на пропущенную дозу - это пропустить ее до конца, но только около 12% клиницистов согласились с этим.

Основная цель этой модели будет состоять в том, чтобы решить вышеупомянутые проблемы путем разработки и создания инструмента, который позволит владельцу отслеживать каждую таблетку, чтобы проглотить ее легким и простым способом, не требующим никакого обучения или сложного обучения с их стороны, чтобы управлять устройством. Это устройство будет интеллектуальным дозатором таблеток.

Дозатор таблеток будет разработан для предотвращения ошибок в домашних больницах и домах престарелых, где каждый пациент должен ежедневно принимать много таблеток, каждый пациент, владеющий устройством, не только резко уменьшит вероятность возникновения ошибок, но и хорошо оптимизирует и ускорит работу сиделок/медсестер, позволяя устройству заботиться об управлении таблетками для них и освобождая время, обычно выделяемое на это.

Это устройство предназначено для регистрации названия таблетки, количества таблеток и часов, в течение которых каждая таблетка фактически

принимается, в сравнении с тем временем, когда она должна была быть принята. В настоящее время наблюдается растущее осознание того, что количество таблеток, назначаемых пожилым людям, говорит о том, что такое большое количество таблеток может иметь негативные последствия для здоровья пациента. Журналы дотов помогут собрать данные по этому вопросу.

Существует необходимость обеспечить беспроводное подключение устройства, чтобы управление устройством (определение часов, когда должна быть принята конкретная таблетка, количество таблеток в каждом отделении и т. д.), а также возможность выдачи предупреждений родственникам владельца или медсестрам, если это необходимо (например, пациент не принимает таблетки).

По сути, устройство должно быть беспроводным электронным устройством, имеющим особое внимание, чтобы сделать его очень точным, поскольку ошибка может оказаться фатальной.

1.1 Обзор существующих прототипов

В работе был предложен медицинский робот для таблеток на основе микроконтроллера MCS-51; этот робот для таблеток может посылать лекарство с помощью шагового двигателя в запланированное время, но не было никаких условий для записи времени, когда пациент действительно принимал лекарство. Помимо вышеперечисленных недостатков, другие недостатки этих предыдущих систем заключаются в следующем:

Члены семьи или пациенты должны заполнить лекарство в дозаторе для таблеток вручную; это дополнительная ответственность для членов семьи пожилых людей или даже пациентов.

Ручное заполнение лекарства в дозаторе для таблеток может привести к тому, что лекарство легко увлажнится.

В работе был предложен интеллектуальный умный дозатор для лекарственных средств (IPB). IPB основан на системе медицинского мешка для таблеток, и IPB посылает сигнал медицинскому мешку из дозатора в соответствующее время. Если пациент не заберет сумку с лекарствами, IPB уведомит об этом сиделок по скайпу. Система IPB улучшает интерактивность между пациентом и опекунами, но она хорошо работает только при наличии подключения к интернету.

В работе был предложен электронный медицинский робот-дозатор для таблеток под названием MedTracker; в MedTracker записывается время, в которое каждая крышка была открыта или закрыта, и передается на ПК по Bluetooth - каналу. Однако MedTracker не предоставляет никаких функций напоминания или подтверждения.

В работе разработана безопасность лекарств и, чтобы избежать путаницы при приеме таблеток среди пожилых людей, в этой статье был

предложен умный таблеточный робот-дозатор с функциями напоминания и подтверждения. Предложенный робот-дозатор для таблеток может снизить ответственность члена семьи за обеспечение правильного и своевременного приема лекарств. Потому что предлагаемый дот содержит звуковой сигнал оповещения для пользователя за определенное время и часы реального времени дают непрерывное время в качестве выходного сигнала.

В работе предложенные мульти дисциплинарные подходы к достижению эффективных и надежных систем для будущего рабочего мониторинга электронного здравоохранения (2014), технологическое слияние интернета вещей, беспроводной сети телесного пространства и облачных вычислений имеют жизненно важное значение в нынешнем

электронном здравоохранении, которые улучшают качество медицинской помощи, в основном ориентированный на пациента мониторинг играет определенную роль в электронных медицинских услугах, которые включают сбор медицинских данных, агрегирование и анализ данных., передача данных и анализ данных здесь весь жизненный цикл мониторинга и компонент основных услуг имеют Дискус, а также проблемы проектирования при разработке схемы мониторинга качества и пациента, ориентированной на пациента, наряду с потенциальным решением. В работе предложил интеллектуальный дизайн и внедрение pillboxdDesign (2014), внедрение pillbox было предложено путем учета проблем пожилых людей, чтобы обеспечить полную безопасность лекарств. Медицинский робот-дозатор с таблетками

будет напоминать пациенту о времени, когда можно контролировать злоупотребление этим препаратом.

В дипломной работе предложена автономная беспроводная система мониторинга окружающей среды с использованием энергии почвы. Система мониторинга может использовать самоподдерживающуюся беспроводную среду с помощью возобновляемых источников энергии, которые могут быть полезны в отдаленных местах, где проблема питания широко распространена, в этом система продемонстрировала, что будет использовать энергию почвы с углеродом, цинковыми электродами.

В проекте также предложена встроенная платформа для веб-мониторинга и управления умным домом, представляющая собой недорогую встроенную платформу для веб-мониторинга и контроля, а также платформу, состоящую из распределенной сети зондирования и управления.

Для получения дополнительной информации о Smart здравоохранения с помощью ВГД можно найти в работах.

1.2 Исследование интеллектуальных систем «Медицинских роботов-дозаторов»

Медицина не всегда была точной наукой, которую мы знаем сегодня, на самом деле, на протяжении всей истории подавляющее большинство болезней и недугов приводили бы к вероятной смерти. К счастью, за последнее столетие медицина пережила большое развитие, что привело к увеличению продолжительности жизни и возможности излечения огромного количества болезней и расстройств здоровья.

Таблетки являются одним из столпов медицины, будучи использованы в качестве прямого лечения какой-то болезни, чтобы гарантировать, что некоторые медицинские условия остаются стабильными.

Наша интеллектуальная система напоминания о лекарствах предназначена для того, чтобы помочь пожилым людям заботиться о себе и принимать лекарства в нужное время и в нужном количестве. Было замечено, что люди в целом пренебрегают своим здоровьем и отдают предпочтение другим вещам, а не принимают свои лекарства. Это также является причиной того, что они забывают вовремя принимать свои рецепты. Многие организации, занимающиеся поддержанием здоровья, практикующие врачи и медицинские исследователи осознали, что более широкое использование напоминаний о пациентах может значительно повысить эффективность лечения хронических заболеваний и качество медицинских услуг для нуждающихся в них пациентов.

Однако многие пациенты и особенно пожилые люди не принимают свои лекарства в нужном количестве. Они либо принимают передозировку лекарств, думая, что это поможет им быстрее выздороветь, либо боятся, что врач прописал большее количество, чем требуется, и принимают их в недостаточной дозировке. Первое приводит к нескольким неудачным последствиям для здоровья, в то время как второе задерживает лечение пациента и в некоторых случаях позволяет болезни распространяться дальше, требуя дальнейшего лечения.

Кроме того, некоторые пациенты настолько заняты своими повседневными делами, что просто забывают принимать лекарства. Это особенно верно для пожилых пациентов, которым приходится принимать более одного лекарства более одного раза в день. Установка будильников - это утомительная задача, которую пациенты слишком ленивы, чтобы устанавливать снова и снова. Если спросить о том, в какое время люди должны принимать свои лекарства, многие забывают ответить правильное время или вспомнить, принимали ли они уже лекарство в течение дня. Пожилые люди специально сталкиваются с этой проблемой из-за ухудшения памяти и в тяжелых случаях забывают, что они уже приняли свой рецепт и повторно принимают одно и то же лекарство 2 или 3 раза за один и тот же срок. Это может быть не вредно для более легких лекарств, но для некоторых сильных и

концентрированных лекарств это может иметь дальнейшие вредные последствия для организма. Именно здесь может помочь наша система напоминаний о лекарствах. Наша система принимает данные рецепта от пользователя, такие как продолжительность рецепта, названия лекарств, время их приема и количество каждого лекарства, которое должно быть принято. После того, как все эти данные будут введены, наша система напомнит пользователю в назначенное время, какое лекарство должно быть принято в виде мобильного уведомления и физического напоминания. Пациенты могут оставить прием лекарств только в нашем приложении. всякий раз, когда время для лекарства истекает, они будут уведомлены, и они должны только принимать свои рецепты в течение этого времени, и никакого другого времени. При правильном применении это значительно уменьшит передозировку лекарств из-за забывчивости, а также напомнит пациентам о необходимости принимать свои лекарства.

2 Программное обеспечение

2.1 Разработка алгоритма работы

Цель предлагаемой модели состоит в том, чтобы достичь тех же или даже лучших результатов, что и многие бренды в этой области, достичь максимального качества с наименьшими возможными затратами, а также с помощью нашей цели, которая заключается в том, чтобы предупредить пациента и его опекунов о регулярных дозах, которые он должен принимать.

Возьмите периодическое время и храните его для каждой таблетки. Тревожить пользователя можно многими способами через само устройство или мобильное приложение.

В случае любой пропущенной дозы он уведомляет об этом родственников пациента.

Нормальный режим предназначен для того, чтобы быть рабочим режимом по умолчанию, когда старшие используют его. Во время этого режима дот остается в режиме ожидания до тех пор, пока пользователь не примет таблетку/дозу.

Как только таблетка будет готова, дот будет излучать акустические и визуальные уведомления через внутренний динамик и светодиоды. Эти эффекты будут длиться в течение короткого промежутка времени, который будет определен позже (около 10 сек), а затем издавать звуковой сигнал каждые 10 секунд. светодиоды будут гореть до тех пор, пока старший не вступит в контакт с устройством и не примет таблетку. Пользователь должен будет нажать кнопку, а затем дот автоматически выдаст таблетку, отключив все визуальные и акустические уведомления после нажатия кнопки и выдачи таблетки.

Если какое-то время (которое будет определено на более поздних этапах проектирования) проходит без нажатия пользователем кнопки, то дот будет выдавать беспроводное уведомление, которое предупредит ответственную медсестру/сиделку о том, что есть таблетка, ожидающая принятия пользователем этим роботом-дозатором.

Режим управления будет задействован, когда дот управляется управлением мобильного приложения. Подключение будет осуществляться по беспроводной сети.

Программа управления может:

Перепрограммировать ДОТ с новым расписанием таблеток.

Активировать последовательность заправки, для которой каждый из отсеков дот станет доступным, а программа укажет, какой тип таблеток соответствует в каждом отсеке, чтобы сиделка / медсестра пополнила его.

Запустить тестовую функцию, в течение которой медсестра / сиделка будет проверять правильность работы дот.

изучении каждой из моделей, какой метод был необходим, чтобы сделать возможным для дота давать только одну таблетку и не более пациентам без использования передовой технологии, которая могла бы значительно повысить как производство, так и конечную цену дота. Был изобретен чисто механический механизм, использующий только пластиковые 3d-печатные детали, которые могли быть реализованы только в квадратной форме, а не в циркуляторе.

2.2 Алгоритм программного обеспечения

Блок-схема представляет собой логику дота при работе. На Рисунке 2.2 показана блок-схема для имплантации дозатора smartpillbox.

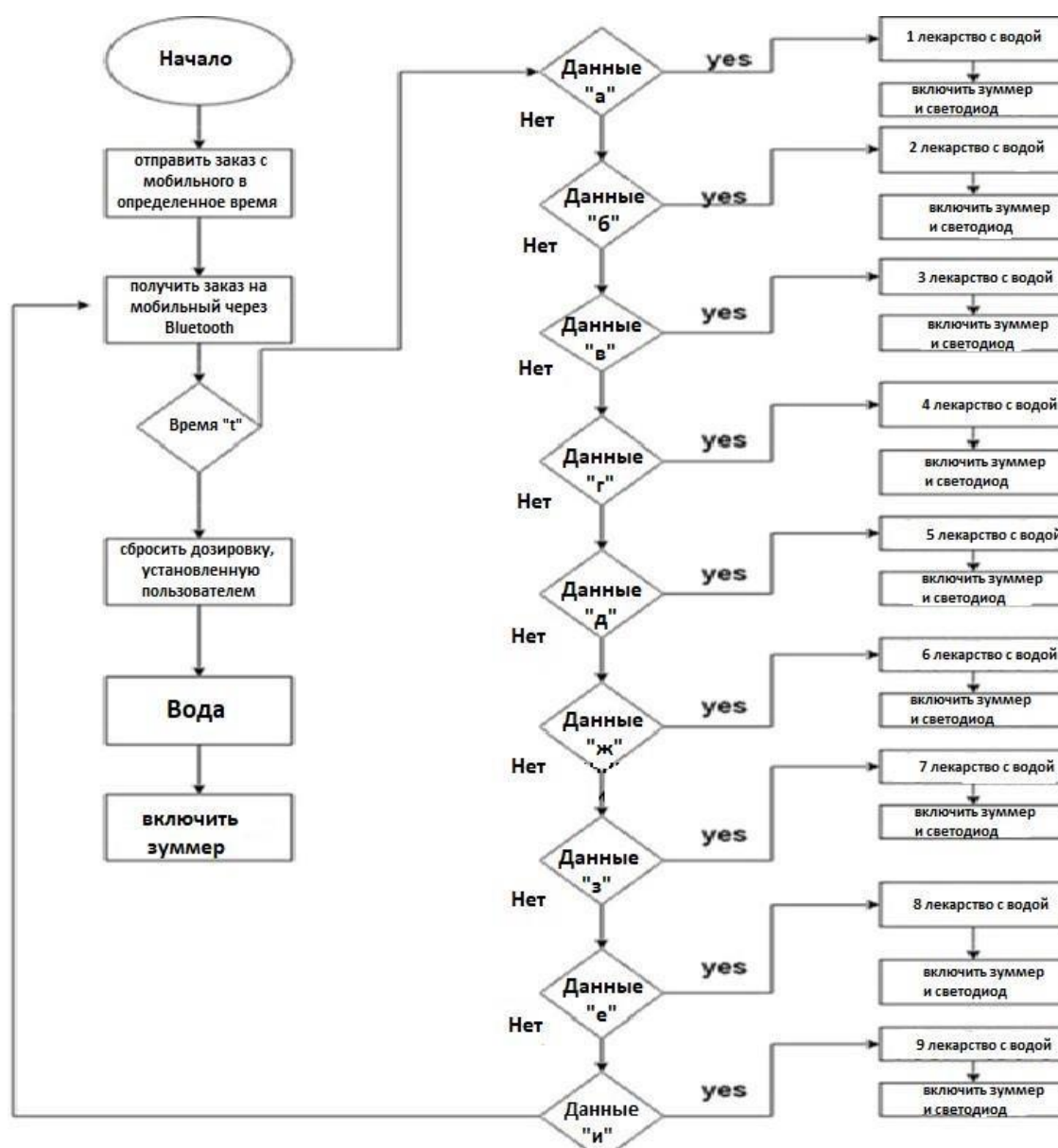


Рисунок 2.2- Блок схема для предлагаемой системы

2.3 Структурная схема

Простая структурная схема была создана для использования в качестве шаблона при проектировании различных частей (как физических, так и программных) дота, эту схему можно увидеть на рисунке 2.2 Плата Arduino станет “мозгом” дота, управляющим каждой из различных областей и будет питаться непосредственно от адаптера (зарядное устройство 12 В). Блок уведомлений будет отвечать за излучение звуковых и световых сигналов для пациента, чтобы он знал, что таблетка должна быть принята, и беспроводных сигналов для тех, кто ухаживает. В отдельном блоке отсеки для таблеток и механизм дозирования таблеток также будут управляться отдельно Arduino. Наконец, внешнее зарядное устройство будет управлять зарядкой аккумулятора непосредственно через Arduino.

1) ArduinoNANO (Рисунок 2.3): ArduinoNano-это небольшая, полная и макетная плата с учетом ATmega328P.

Он имеет в значительной степени аналогичную полезность ArduinoDuemilanove, однако в альтернативном пакете. Он нуждается в постоянном токе только в управляющем разъеме и работает с USB-соединением Mini-B, а не стандартным. ArduinoNano является открытым исходным кодом.

ArduinoNano может управляться с помощью USB-ассоциации, нерегулируемого внешнего источника питания (6V 30) или направленного внешнего источника питания 5V (pin 27). Источник, естественно, выбирается для самого поразительного источника v. ATmega328P на ArduinoNano поставляется с загрузчиком, который позволяет передавать новый код без использования внешнего инженера-программиста оборудования. Он передает использование первой конвенции STK500. Вы также можете обойти загрузчик и запрограммировать контроллер через ICSP (In-CircuitSerialheader, использующий ArduinoISP или аналогичный).

2) ESP 8266 NodeMCU 12EDev комплект: NodeMCU-это фаза для IoT, которая является организацией с открытым исходным кодом, которая, кроме того, предохраняет прошивку, которая продолжает работать на ESP8266 WiFiSoC от EspressifSystems и gear, которая опирается на модуль ESP-12. Так что мы здесь подразумеваем Дэва. сам блок не является прошивкой, которая продолжает работать на нем. В презрении к тому, как чип использует Luascriptingvernacular, все же мы полагали, что было бы более утешительно использовать Центр Arduino для чипа с конкретной истинной целью программирования его через условие Arduino.

Ядро Arduino поставляется некоторыми изобретательными вентиляторами ESP8266, которые можно найти на названии GitHub под

названием "ArduinocoreforESP8266 WiFichip", которые могут быть предварительно представлены в ArduinoIDE через исполнительный совет или, с другой стороны, различными разнообразными способами.

Ядро-это курс действий программных библиотек и частей, необходимых ArduinoIDE для накопления исходного кода ArduinoC/Cpp вплоть до машинного жаргона ObjectiveMCU, который является чипом ESP8266 в нашем случае.

3) зуммер: пьезоэлектрические сигналы, или пьезозвонки, как их иногда называют, были разработаны японскими производителями и вписаны в широкую группу предметов в 1970-1980-е гг.

Этот прогресс в первую очередь принес свои плоды благодаря полезным усилиям. В 1951 году они создали научно-исследовательский комитет по применению титаната бария, который дал возможность этим организациям реализовать несколько пьезоэлектрических разработок и разработок. Пьезоэлектрические материалы являются либо нормально доступными, либо синтетическими. Пьезокерамика - это класс синтетического материала, который позирует пьезоэлектрическому удару и широко используется для изготовления пластины, сердцевины пьезозвонка. В тот момент, когда они подвергаются воздействию обменного электрического поля, они расширяются или упаковываются, в соответствии с повторением флага, таким образом создавая звук.

4) приложение RoboRemo: простая форма интерфейса в соответствии с вашим заданием требует: включить защелки, ползунки, светодиоды и так далее. Используйте акселерометр для управления руководством радиоуправляемый авто, или для наклона автомата. Используйте графики, чтобы показать постоянную информацию от датчиков. Вы можете напрямую взаимодействовать с платой Arduino, используя OTGlink (если ваш гаджет поддерживает OTG), или вы можете использовать удаленный модуль и ассоциировать его с Bluetooth или WIFI. Клиент беспроводной протокол TCP, сервер TCP и UDP, мобильного интернета TCP клиент, сервер TCP и UDP.

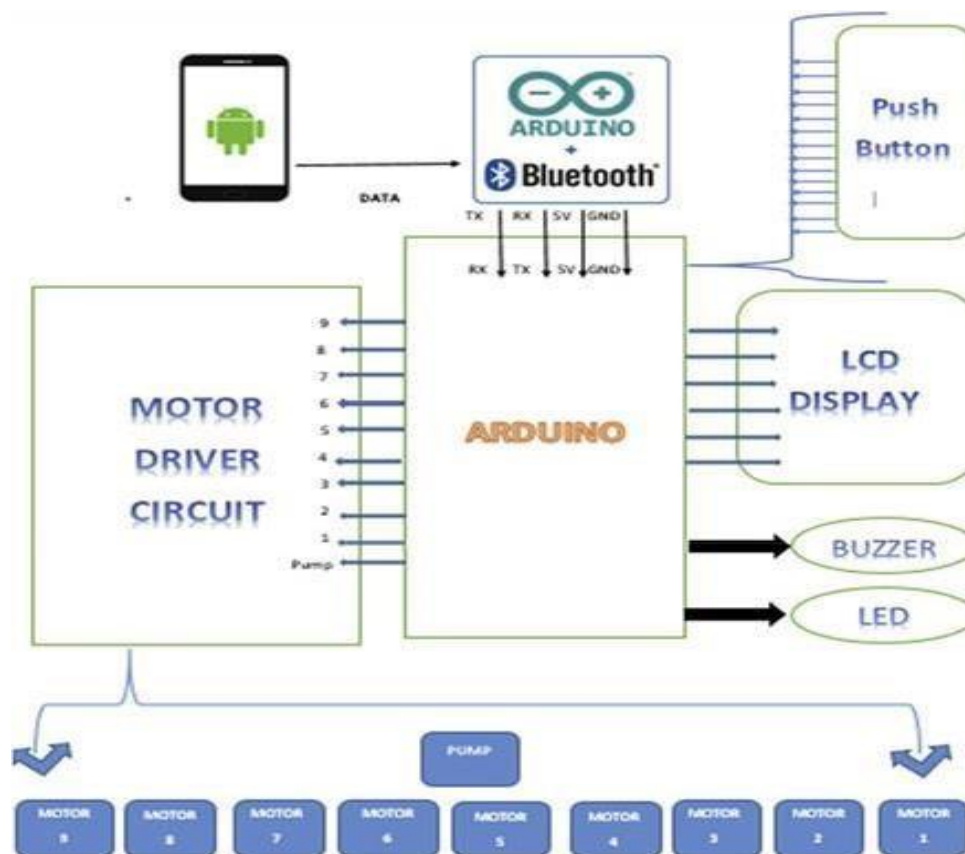


Рисунок 2.3 - Структурная схема предлагаемой системы

2.4 Характеристика используемых деталей

Структура этого проекта разделена на две подкатегории, первая- это пакет ESP, который состоит из всех классов, подклассов и объектов, необходимых для того, чтобы Wi-Fi установил сервер и стал готовым к подключению. Второй предназначен для ArduinoNANO, который представляет собой классы, настроенные в программном обеспечении для запуска последовательных портов, используемых для связи, на рисунке ниже показан сервер двух пакетов и закрывает ассоциацию. Модуль — WI-FI- это суть всего; он дает нам возможность управлять системной частью устройства.

1) SoftwareSerial: как следует из его названия, это класс, который позволяет нам определить любые два цифровых контакта как передатчик и приемник. Мы должны были использовать это на стадии отладки и тестирования, было более эффективно использовать более одной последовательной связи, пока скорость передачи данных низкая. Сначала мы используем метод `begin ()`, а затем определяем скорость передачи данных в

ботах для запуска последовательной связи UART на указанных выводах. Это позволило нам контролировать всю систему через последовательный монитор, а также видеть, как быстро идет передача, определило стабильность и возможности системы. Проверка последовательной передачи осуществлялась с помощью метода `available ()`, метод `read ()` и `write ()` предназначались для передачи и приема на устройство ESP и от него.



Рисунок 2.4- Arduino nano, ESP 8266 Node MCU 12E

2) таймер один: таймер один - это класс, используемый для управления внутренним таймером Arduino, чтобы установить периодическое время для каждой дозы, мы должны были подсчитать наименьшую шкалу времени, которая находится в секундах. Сначала мы должны инициализировать таймер с помощью метода `initialize ()` для подсчета единицы измерения или, что более удобно, как часто таймер инициирует прерывание для обработки ISR (`interruptServiceRoutine`). Затем присоедините к нему прерывание с помощью метода `attachInterrupt ()`, который определяет ISR, который должен быть выполнен. Затем `timerisr ()` содержит команду, которая должна быть выполнена, которая изначально управляет четырьмя отдельными счетчиками (см. Рисунок 2.5).

3)



Рисунок 2.5 - Пьезоизлучатель

3 Разработка 3Д модели

Различные модули должны быть определены, как указано в разделе "структурная схема" в концептуальном проекте. Для того чтобы приступить к окончательному проектированию дота, необходимо в обязательном порядке расположить каждый из модулей в определенной зоне. Эти модули будут представлять собой механизм заправки таблеток, механизм дозирования таблеток и электронный модуль, где будет располагаться большая часть схем, таких как Arduino, переключатели и т. д. Также на диаграмме должно быть отмечено расположение лотка для таблеток, где каждая из таблеток выйдет из дозатора для таблеток.

Рисунок 3.1 представляет собой минималистское поперечное сечение дота, размещающее модули внутри корпуса дота.

3.1 Конструкция модуля дозаправки таблеток

Этот модуль будет отвечать за то, чтобы позволить человеку, ответственному за заправку механизма, адекватно заправить ДОТ с минимальным шансом на ошибки в этом процессе.

Отверстие будет расположено в верхней части дота. Заправка будет производиться с помощью программного обеспечения управления ДОТом. Ответственному лицу будет предложено вставить определенный тип таблеток в умный дозатор с таблетками и нажать кнопку в приложении, когда это действие будет завершено и отсек для этой таблетки будет полностью заполнен. Как только пользователь нажмет кнопку в приложении, программа попросит пополнить другой тип таблеток и нажмет кнопку а, и действие будет повторяться до тех пор, пока все отделения для таблеток не будут заполнены.



Рисунок 3.1- Схема модуля блока

Хранение таблеток будет состоять из застежек-молний, где все различные отделения будут содержаться в виде полых столбиков, которые будут теми, кто держит таблетки в них. Для того чтобы таблетки могли попасть в каждый отсек для таблеток, серво, установленный под крышкой, которая будет удерживать молнии, будет выравнивать по команде, посылаемой компьютером, каждый из отсеков с входным отверстием, заставляя таблетки падать непосредственно в выбранный отсек. Как только используемый отсек будет заполнен, сервопривод повернет трубку так, чтобы выровнять следующий отсек с входным отверстием (см. Рисунок 3.2).

3.2 Конструкция механизма дозирования таблеток

Этот модуль будет принимать меры каждый раз, когда необходимо раздать таблетку, поскольку именно он будет отвечать за то, чтобы таблетка упала в лоток для таблеток в тот момент, когда Адриано посылает сигнал.

Модуль должен быть способен выбрать правильный отсек из разных отсеков хранилища таблеток, и очень важно, чтобы он получил только одну таблетку от компартмента, так как получение более одной таблетки может оказаться фатальным.

Механизм для модуля дозирования таблеток будет использовать два управляемых Arduino сервопривода, один из которых расположен под крышкой и прикреплен к цилиндру хранения таблеток (такой же сервопривод, как и тот, который используется в механизме дозирования таблеток), а другой, который будет управлять промежуточным механизмом, состоит из трех различных стадий: Стадия по умолчанию, стадия загрузки таблеток и стадия выпуска таблеток. После этапа выпуска таблеток система будет сброшена на стадию по умолчанию. Система проходит через три этапа каждый раз, когда таблетка должна быть предоставлена пользователю.

Детали, участвующие в механизме дозирования.

Части были обозначены от 1 до 4 на следующих диаграммах для более легкого понимания:

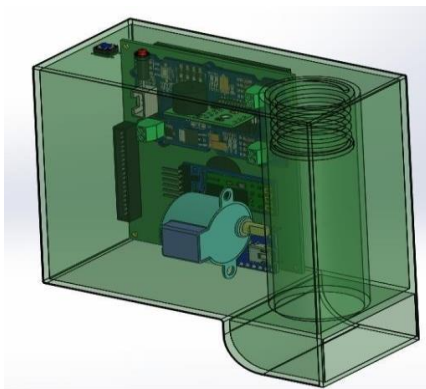


Рисунок 3.2 - Иллюстрация с видом на хранилище таблеток

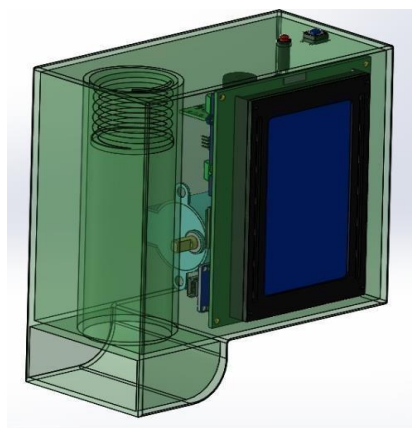


Рисунок 3.3 - Вид на камеру с таблетками

Часть 4 электронная таблетка трубы. Часть 1 хранение таблеток

Эта часть состоит из цилиндра с полыми туннелями, которые охватывают всю длину трубы, распределенную радиально вдоль трубы. В центре детали имеется круглое отверстие, через которое будут проходить различные кабели для цепей, когда дот будет собран. В этих отсеках будут храниться таблетки до тех пор, пока их не нужно будет раздать. Каждый отсек будет содержать только таблетки одного и того же типа, следовательно, может быть столько же типов таблеток, сколько пригодных для использования отсеков минус один.



Рисунок 3.4 - Механизм заправки таблетки

Часть 4 электронная таблетка трубы. Часть 1 хранение таблеток. Эта часть состоит из цилиндра с полыми туннелями, которые охватывают всю длину трубы, распределенную радиально вдоль трубы. В центре детали имеется круглое отверстие, через которое будут проходить различные кабели для цепей, когда дот будет собран. В этих отсеках будут храниться таблетки до тех пор, пока их не нужно будет раздать. Каждый отсек будет содержать только таблетки одного и того же типа, следовательно, может быть столько же типов таблеток, сколько пригодных для использования отсеков минус один.

Часть 3 таблетки камеры. Камера для таблеток состоит из цилиндра с круглым отверстием в центре и другим проходом рядом с отверстием. Этот проход достаточно широк, чтобы держать таблетку в вертикальном положении.

Высота также совпадает с моделью таблетки, которая будет использоваться в этом ящике для таблеток, поэтому эта часть может содержать только одну таблетку, стоящую в вертикальном положении в своем пространстве для хранения (пространство для хранения будет боковым проходом части, которая отныне также будет называться камерой для таблеток). Деталь можно посмотреть на фигуре 3.5.

3.3 Обзор этапа сборки 3Д модели

Эти четыре части будут представлены в минималистском виде, как показано на рисунок 3.5 Во время этого объяснения расположение различных проходов будет отмечено красным цветом в каждой из частей (эти проходы будут теми, которые таблетка должна пройти, чтобы добраться до выхода), как это видно на рисунок 3.6

Стадия 1 стадия по умолчанию

Эта стадия-стадия, на которой дот находится по умолчанию, когда никакая таблетка не нуждается в дозировании, исходное положение, с которого начинается механизм дозирования таблеток.

Отсеки для таблеток в хранилище для таблеток (часть 1) расположены таким образом, чтобы они не были выровнены с отверстием устройства для хранения таблеток.

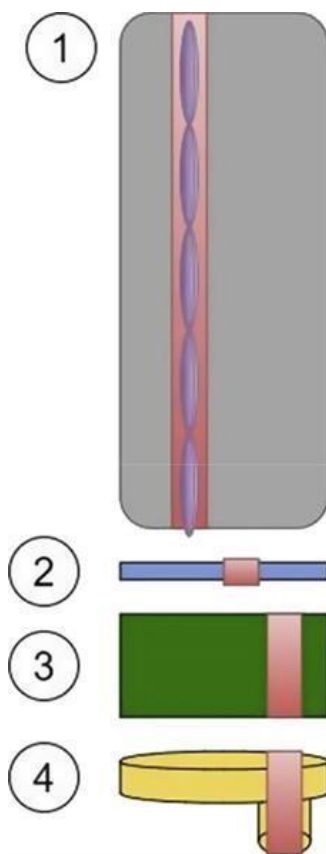


Рисунок 3.5 - Заправка таблетки в поперечном сечении

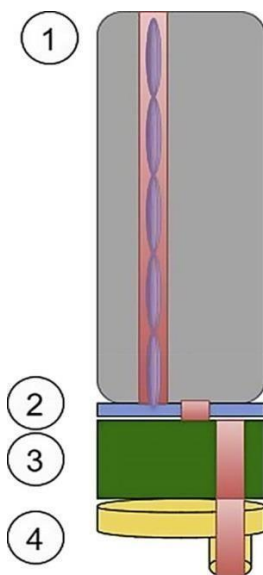


Рисунок 3.6 - Положение дота внутренних частей на этапе дефолта.

Этап 3. Этап выпуска таблеток

Когда дот достигает этой стадии, таблеточная камера (часть 3) будет двигаться, выравниваясь с трубой таблетки и, следовательно, позволяя таблетке, хранящейся в камере, падать через трубу, это положение показано на Рисунок 3.7. Затем устройство для хранения таблеток (часть 1) вращается,

выравнивая пустой отсек устройства для хранения таблеток с отверстием люка, предотвращая выравнивание любых других отсеков над люком. Конечное положение тогда показано на Рисунок 3.8.

Дот состоит в общей сложности из 9 пластиковых 3D-печатных деталей, которые соответствуют как внешнему корпусу, так и внутренним механизмам дот. Планы для различных частей будут показаны здесь в уменьшенном масштабе. Фактические полноразмерные планы с измерениями каждой из частей будут приложены в приложении. Окончательное оборудование после этого показанона Рисунок 3.7 и 3.8.

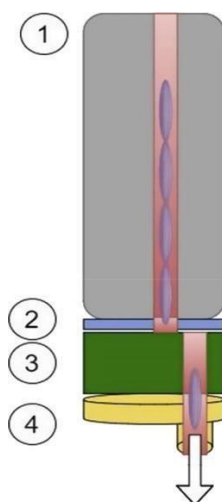


Рисунок 3.7 - Положение дота внутренних частей на этапе загрузки таблетки

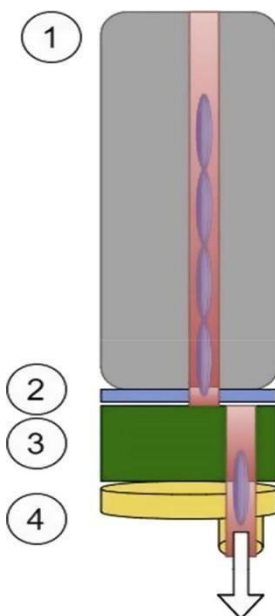


Рисунок 3.8 - Положение дота внутренних частей на первом этапе этапа выпуска таблетки

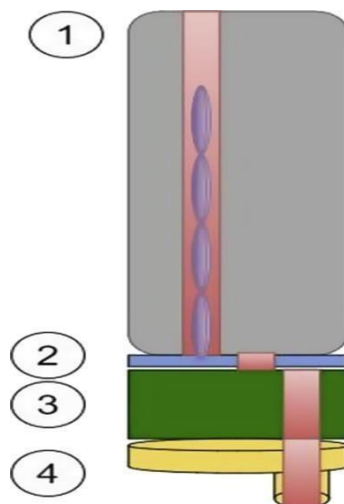


Рисунок 3.9 - Положение внутренних частей ДОТа на последнем этапе этапа выпуска таблетки

3.4 Реализация 3Д модели

Любые программные продукты с точки зрения проектировщика по назначению можно условно классифицировать:

1) Универсальные (типа AutoCAD, MathCAD) – для решения широкого круга задач в машиностроении, приборостроении, архитектуре, строительстве и т. д.;

2) Специализированные (типа ElectronicsWorkbench, PCAD, OrCAD применяют только для проектирования электротехнической и радиоэлектронной аппаратуры, САПР ТП – для решения задач технологической подготовки производства, ИНСВАР – для проектирования сварочной технологической оснастки);

3) Утилитарного назначения (языки программирования, программы-конверторы, просмотрщики и т. п.).

Для выполнения данной дипломной работы выбираем следующий программный продукт – SolidWorks 2006. Эта программа относится к универсальным. Критериями её отбора являются:

1) Возможность эффективного решения поставленной задачи;

2) Возможность поддержки форматов стандартных внешних файлов обмена;

3) Возможность разработки приложений для данной системы на языках высокого уровня, т. е. открытость системы;

4) Простота освоения; 5) Высокая степень распространенности.

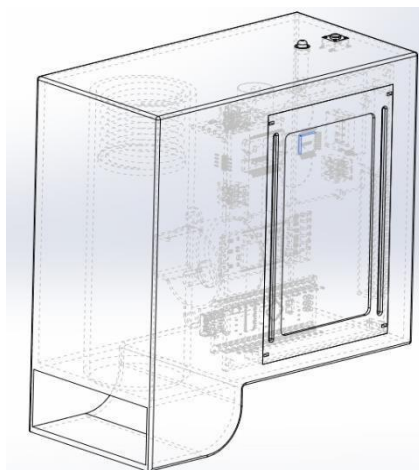


Рисунок 3.10- Масштабный план для части таблеточной камеры

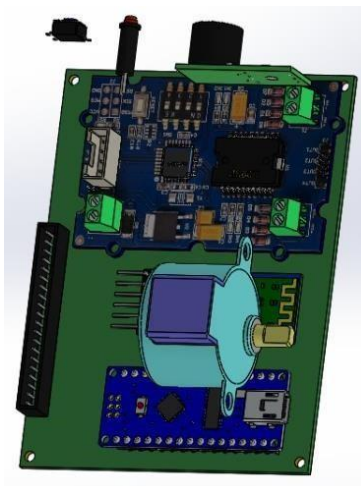


Рисунок 3.11 - Масштабный план для части электроники

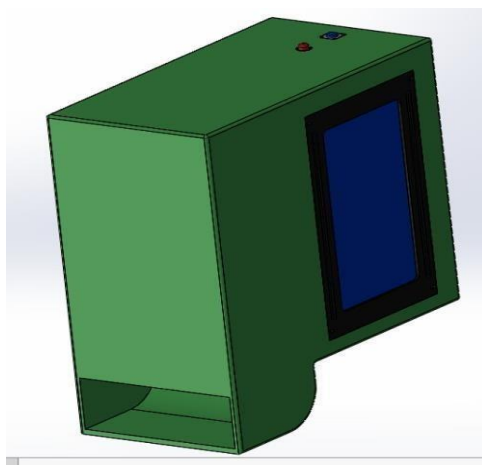


Рисунок 3.12- Соединение электроники с Медицинским роботом-дозатором

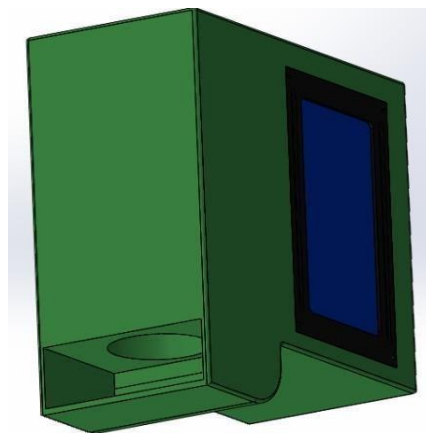


Рисунок 3.13 - Соединение электроники с Медицинским роботом-дозатором

Мобильное приложение разработано для полного контроля над smartpillbox удаленно с помощью приложения Android. Сначала, когда откроете увидите это.

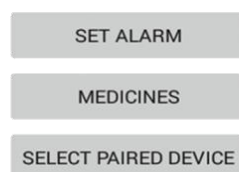


Рисунок 3.14 - Главный экран мобильного приложения

Затем после подключения между мобильным приложением и smartpillbox нажимается средняя кнопка "лекарства", чтобы написать лекарства, которые принимает пациент.

После этого переходит к последней кнопке "SETALARM", чтобы установить будильник на нужное нам время, есть много мелодий, и если человек забудет установить будильник и нажмет кнопку goout, устройство выдаст такое предупреждающее сообщение.

Подключиться с помощью телефона: открывается мобильное приложение через Roboremo.

Изменить время каждой таблетки и ввести ее в противоположное пространство.

Ждать каждый раз, когда появится уведомление, связанная с таблеткой.

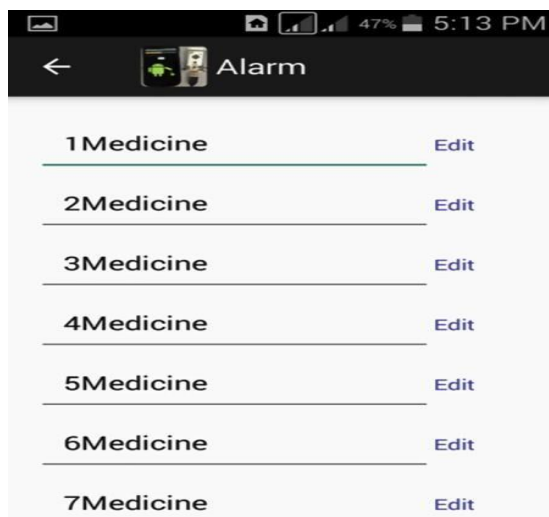


Рисунок 3.15 – Установите конфигурацию

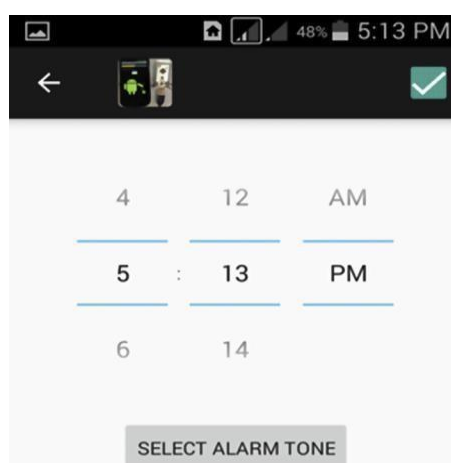


Рисунок 3.16 - Установка будильника

Во-первых, протестирована ESP-соединение с удаленным приложением, и оно было успешным, поскольку было общение с приложением, используя определенные шаблоны строк или коды. Также тестировался мониторинг через последовательный монитор, чтобы проверить, что нет никакой задержки в передаче или каких-либо потерянных символов, тот же метод тестирования был применен для проверки связи между ESP и ArduinoNANO, результаты показали, что задержка была только на 100 мс. эта небольшая задержка никак не повлияла на время приема доз, графики ниже и интерфейс, показанный ниже, гарантируют (см. Рисунок 3.17- 3.18).

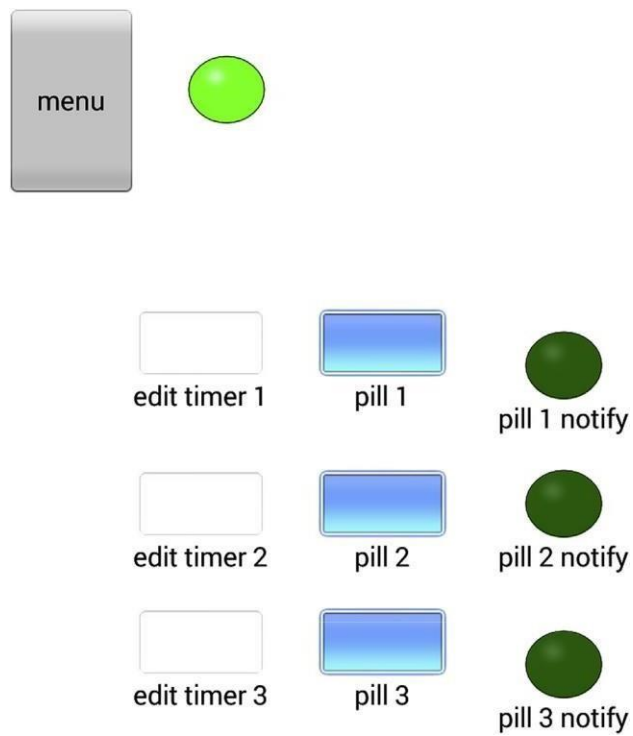


Рисунок 3.17 – Пользовательский интерфейс

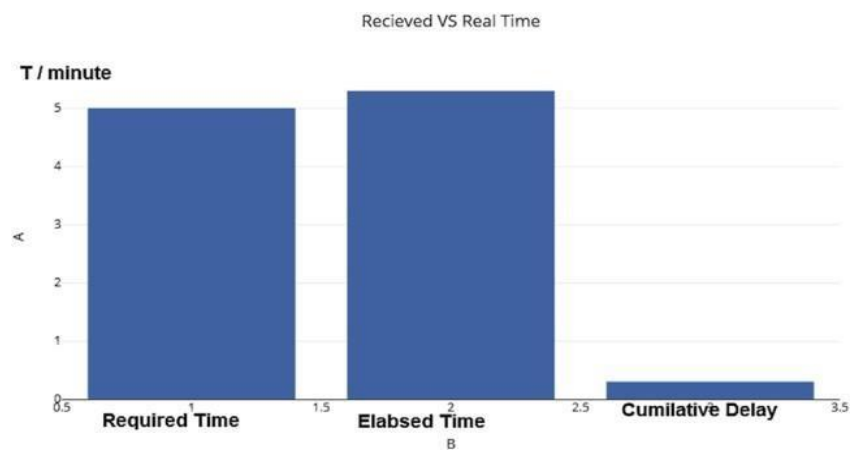


Рисунок 3.18 - Требуемое и затраченное время

ЗАКЛЮЧЕНИЕ

В этой работе предлагается и актуализируется умный дозатор. Результаты показали насколько важна, существующая на данный момент, проблема неправильного приема лекарств, а также ее регулярность. Он продуктивно контролирует время приема лекарств пациентами и пожилыми людьми. Это дополнительно уменьшает риск того, что пациент пропускает и откладывает прием лекарств. Удаленный пользовательский интерфейс соединяется с программным обеспечением RoboRemo, так что родительские цифры могут помочь пациенту. Который добавляет больше функциональности, применяя больше удобства использования через локальную сеть или используя интернет, если он доступен. Контурный дизайн очень подходит для пакетов лекарств. В дальнейшем полагаем, что внедрение этой системы будет незаменимым помощником для всех людей, нуждающимися в ней. Таким образом, пропуск и отсрочка принятия лекарств могут быть полностью устранены. Так же был рассмотрен и решён вопрос мобильности данного проекта. Были выполнены следующие задачи:

- построили алгоритм работы программного обеспечения;
- разработали электрическую схему «Медицинского робота-дозатора»;
- построили 3Д-модель «Медицинского робота-дозатора» в программном обеспечении SolidWorks.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Chang Hong-Ti, Jhu-Yu Chen. (2014). The intelligent pill box e design and implementation.
- 2 Clifton D., Wong D., Clifton L., Wilson S, Way R. (2013). A large-scale clinical validation of an integrated monitoring system in the Emergency Department.
- 3 Huang S.C, Chang H.Y. (2006). The intelligent pill box-design and implementation.
- 4 JR. R. (2009). Rosati RJ. Evaluation of remote monitoring in home health care. In: 2009 international conference on eHealth, telemedicine, and social medicine;.
- 5 Lisby M, Nielson L.P, Mainz J. (2005). Lisby M, Nielsen LP, Mainz J. Errors in the medication process: frequency, type, and potential clinical consequences.
- 6 PJ., V. (2006). An individual based framework for a study on medical error. Sankar AP, Nevedal DC, Neufeld S. (б.д.). What is a missed dose?
- 7 Implications for construct validity and patient adherence.
- 8 Sawand A, Djahel S, Zhang Z, Na F. (2014). Multidisciplinary approaches to achieving efficient and trustworthy eHealth monitoring systems., (стр. 187).
- 9 Rosati RJ. Evaluation of remote monitoring in home health care. In: 2009 international conference on eHealth, telemedicine, and social medicine; 2009. p. 25e7.
- 10 M3DITRACK3R. A design of an automated patient tracking and medicine Dispensing Mobile robot for senior citizens published in computer, communications, and control technology (I4CT). 2014. International Conference on Date 2-4 Sept. 2014 at Langkawi.
- 10 A smartphone-enhanced pill-dispenser providing patient identification and intake recognition C. Crema, A. Depari, A. Flammini, M. Lavarini, E. Sisinni published in medical measurements and applications (MeMeA). 2015. IEEE International Symposium on Date 7-9 May 2015 at Torino, Italy.
- 11 Kulkarni Alok, Sathe Sampada. Healthcare applications of the internet of things:A review. (IJCSIT) Int J Comput Sci Inf Technol 2014;5(No 5): 6229e32.
- 13 Pang Z. Technologies and architectures of the internet-of-things (IoT) for health and well-being. Ph.D. dissertation. Stockholm, Sweden: Dept. Electron.Syst., School Inf. Commun.Technol., Royal Inst. Technology (KTH); 2013.
- 12 MacLaughlin Eric J, et al. Assessing medication adherence in the elderly. Drugs Aging 2005;22(3):231e55.

ПРИЛОЖЕНИЕ А

Обзор внутренней и наружной части 3д модели умного дозатора лекарственных препаратов



ПРИЛОЖЕНИЕ Б

Код для активации медицинского робота

```

//Medicine Reminder using Arduino Uno
// Reminds to take medicine at 8am, 2pm, 8pm
/* The circuit:
  LCD RS pin to digital pin 12
  LCD Enable pin to digital pin 11
  LCD D4 pin to digital pin 5
  LCD D5 pin to digital pin 4
  LCD D6 pin to digital pin 3
  LCD D7 pin to digital pin 2
  LCD R/W pin to ground
  LCD VSS pin to ground
  LCD VCC pin to 5V
  10K resistor:
  ends to +5V and ground
  wiper to LCD VO pin (pin 3)*/

#include <LiquidCrystal.h>
#include <Wire.h>
#include <RTClib.h>
#include <EEPROM.h>

intpushVal = 0;
intval;
int val2;
intaddr = 0;

RTC_DS3231 rtc;

constint rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;           // lcd pins
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

#define getWellsoon 0
#define HELP_SCREEN 1
#define TIME_SCREEN 2

//boolpushPressed;           //flag to keep track of push button state
intpushpressed = 0;
constintledPin = LED_BUILTIN;           // buzzer and led pin
intledState = LOW;
int Signal = 0;

```

```

int buzz = 13;
int push1state, push2state, push3state, stopinState = 0;    //
int push1Flag, push2Flag, Push3Flag = false;               // push button flags
int push1pin = 9;
int push2pin = 8;
int push3pin = 7;
intstopPin = A0;
int screens = 0;      // screen to show
intmaxScreen = 2;     // screen count
boolisScreenChanged = true;

longpreviousMillis = 0;
long interval = 500;    // buzzing interval
unsigned long currentMillis;

longpreviousMillisLCD = 0; // for LCD screen update
longintervalLCD = 2000;    // Screen cycling interval
unsigned long currentMillisLCD;

// Set Reminder Change Time
int buzz8amHH = 8;      // HH - hours    ##Set these for reminder time in 24hr
Format
int buzz8amMM = 00;     // MM - Minute
int buzz8amSS = 00;     // SS - Seconds

int buzz2pmHH = 14;     // HH - hours
int buzz2pmMM = 00;     // MM - Minute
int buzz2pmSS = 00;     // SS - Seconds

int buzz8pmHH = 20;     // HH - hours
int buzz8pmMM = 00;     // MM - Minute
int buzz8pmSS = 00;     // SS - Seconds

intnowHr, nowMin, nowSec;           // to show current mm,hh,ss

// All messeges
voidgwsMessege(){          // print get well soon messege
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Stay Healthy :)");    // Give some cheers
lcd.setCursor(0, 1);

```

```

lcd.print("Get Well Soon :)"); // wish
}

```

```

voidhelpScreen() {           // function to display 1st screen in LCD
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Press Buttons");
lcd.setCursor(0, 1);
lcd.print("for Reminder...!");

}

```

```

voidtimeScreen() {           // function to display Date and time in LCD screen
DateTime now = rtc.now();      // take rtc time and print in display
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Time:");
lcd.setCursor(6, 0);
lcd.print(nowHr = now.hour(), DEC);
lcd.print(":");
lcd.print(nowMin = now.minute(), DEC);
lcd.print(":");
lcd.print(nowSec = now.second(), DEC);
lcd.setCursor(0, 1);
lcd.print("Date: ");
lcd.print(now.day(), DEC);
lcd.print("/");
lcd.print(now.month(), DEC);
lcd.print("/");
lcd.print(now.year(), DEC);
}

```

```

void setup() {

Serial.begin(9600);           // start serial debugging
if (! rtc.begin()) {          // check if rtc is connected
Serial.println("Couldn't find RTC");
while (1);
}
if (rtc.lostPower()) {
Serial.println("RTC lost power, lets set the time!");
}
}

```

```

// rtc.adjust(DateTime(F(_DATE), F(_TIME))); // uncomment this to set the
current time and then comment in next upload when u set the time
rtc.adjust(DateTime(2019, 1, 10, 7, 59, 30)); // manual time set

lcd.begin(16, 2);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Welcome To"); // print a messege at startup
lcd.setCursor(0, 1);
lcd.print("Circuit Digest");
delay(1000);
pinMode(push1pin, INPUT); // define push button pins type
pinMode(push2pin, INPUT);
pinMode(push3pin, INPUT);
pinMode(stopPin, INPUT);
pinMode(ledPin, OUTPUT);
delay(200);
Serial.println(EEPROM.read(addr));
val2 = EEPROM.read(addr); // read previosuly saved value of push
button to start from where it was left previously
switch (val2) {
case 1:
Serial.println("Set for 1/day");
push1state = 1;
push2state = 0;
push3state = 0;
pushVal = 1;
break;
case 2:
Serial.println("Set for 2/day");
push1state = 0;
push2state = 1;
push3state = 0;
pushVal = 2;
break;
case 3:
Serial.println("Set for 3/day");
push1state = 0;
push2state = 0;
push3state = 1;

```

```

pushVal = 3;

break;
}

}

void loop() {
push1();           //call to set once/day
push2();           //call to set twice/day
push3();           //call to set thrice/day
if (pushVal == 1) {           // if push button 1 pressed then remind at
8am
at8am();           //function to start uzzing at 8am
}
else if (pushVal == 2) {           // if push button 2 pressed then remind at
8am and 8pm
at8am();
at8pm();           //function to start uzzing at 8mm
}
else if (pushVal == 3) {           // if push button 3 pressed then remind at
8am and 8pm
at8am();
at2pm();           //function to start uzzing at 8mm
at8pm();
}

currentMillisLCD = millis();           // start millis for LCD screen switching
at defined interval of time
push1state = digitalRead(push1pin);           // start reading all push button pins
push2state = digitalRead(push2pin);
push3state = digitalRead(push3pin);
stopinState = digitalRead(stopPin);

stopPins();           // call to stop buzzing
changeScreen();           // screen cycle function

}

// push buttons
void push1() {           // function to set reminder once/day
if (push1state == 1) {

```

```

    push1state = 0;
    push2state = 0;
    push3state = 0;
    //  pushPressed = true;
    EEPROM.write(addr, 1);
    Serial.print("Push1 Written : "); Serial.println(EEPROM.read(addr)); // for
    debugging
    pushVal = 1;                                //save the state of push button-1
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Reminder set ");
    lcd.setCursor(0, 1);
    lcd.print("for Once/day !");
    delay(1200);
    lcd.clear();
    }
}

```

```

void push2() {                                //function to set reminder twice/day
if (push2state == 1) {
    push2state = 0;
    push1state = 0;
    push3state = 0;
    //  pushPressed = true;
    EEPROM.write(addr, 2);
    Serial.print("Push2 Written : "); Serial.println(EEPROM.read(addr));
    pushVal = 2;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Reminder set ");
    lcd.setCursor(0, 1);
    lcd.print("for Twice/day !");
    delay(1200);
    lcd.clear();
    }
}

```

```

void push3() {                                //function to set reminder thrice/day
if (push3state == 1) {
    push3state = 0;
    push1state = 0;
    push2state = 0;

```

```

//  pushPressed = true;
EEPROM.write(addr, 3);
Serial.print("Push3 Written : "); Serial.println(EEPROM.read(addr));
pushVal = 3;
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Reminder set ");
lcd.setCursor(0, 1);
lcd.print("for Thrice/day !");
delay(1200);
lcd.clear();
}
}

void stopPins() {           //function to stop buzzing when user pushes stop push
button
if (stopinState == 1) {
//  stopinState = 0;
//  pushPressed = true;
pushpressed = 1;
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Take Medicine ");
lcd.setCursor(0, 1);
lcd.print("with Warm Water");
delay(1200);
lcd.clear();
}
}

void startBuzz() {         // function to start buzzing when time reaches to
defined interval

// if (pushPressed == false) {
if (pushpressed == 0) {
Serial.println("pushpressed is false in blink");
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
previousMillis = currentMillis;      // save the last time you blinked the LED
Serial.println("Start Buzzing");
if (ledState == LOW) {              // if the LED is off turn it on and vice-versa:
ledState = HIGH;

```

```

    } else {
    ledState = LOW;
        }
    digitalWrite(ledPin, ledState);
    }
    }
    else if (pushpressed == 1) {
    Serial.println("pushpressed is true");
    ledState = LOW;
    digitalWrite(ledPin, ledState);
    }
    }

void at8am() {                                // function to start buzzing at 8am
DateTime now = rtc.now();
if (int(now.hour()) >= buzz8amHH) {
if (int(now.minute()) >= buzz8amMM) {
if (int(now.second()) > buzz8amSS) {
    //////////////////////////////////////

startBuzz();
    //////////////////////////////////////
    }
    }
    }
}

void at2pm() {                                // function to start buzzing at 2pm
DateTime now = rtc.now();
if (int(now.hour()) >= buzz2pmHH) {
if (int(now.minute()) >= buzz2pmMM) {
if (int(now.second()) > buzz2pmSS) {

    //////////////////////////////////////
startBuzz();
    //////////////////////////////////////
    }
    }
    }
}

void at8pm() {                                // function to start buzzing at 8pm

```

```

DateTime now = rtc.now();
if (int(now.hour()) >= buzz8pmHH) {
if (int(now.minute()) >= buzz8pmMM) {
if (int(now.second()) > buzz8pmSS) {

        ///////////////////////////////////////////////////
startBuzz();
        ///////////////////////////////////////////////////
    }
    }
    }
}

//Screen Cycling
voidchangeScreen() {           //function for Screen Cycling
    // Start switching screen every defined intervalLCD
if (currentMillisLCD - previousMillisLCD>intervalLCD)           // save the last time
you changed the display
    {
previousMillisLCD = currentMillisLCD;
screens++;
if (screens >maxScreen) {
screens = 0; // all screens over -> start from 1st
    }
isScreenChanged = true;
    }

    // Start displaying current screen
if (isScreenChanged) // only update the screen if the screen is changed.
    {
isScreenChanged = false; // reset for next iteration
switch (screens)
    {
casegetWellsoon:
gwsMessege();           // get well soon message
break;
case HELP_SCREEN:
helpScreen();           // instruction screen
break;
case TIME_SCREEN:
timeScreen();           // to print date and time
break;

```

```
default:
    //NOT SET.
break;
}
}
```