Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

Система контроля здоровья человека

БГУИР КР 1-40 02 01 327 ПЗ

Студент: Черевко И. М.

Руководитель: Жук Д. С.

Минск 2022

Содержание

[ВВЕДЕНИЕ 3](#_Toc100833151)

[1 ОБЗОР ЛИТЕРАТУРЫ 4](#_Toc100833152)

1.1 Информация о платформе Arduino ………………………………………………4

1.2 Обзор аналогов ……………………………………………………………………7

[1.3 Постановка задачи 8](#_Toc100833153)

[2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ 9](#_Toc100833154)

[2.1 Модуль авторизации 9](#_Toc100833155)

[2.2 Модуль работы с приложением 9](#_Toc100833156)

[2.3 Модуль пользовательского интерфейса 9](#_Toc100833157)

[2.4 Модуль оперативного хранения данных 9](#_Toc100833158)

[2.5 Модуль чтения и записи данных 10](#_Toc100833159)

[2.6 Модуль сети 10](#_Toc100833160)

[3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ 11](#_Toc100833161)

[3.1 Описание структур приложения 11](#_Toc100833162)

[ЗАКЛЮЧЕНИЕ 12](#_Toc100833163)

[ЛИТЕРАТУРА 13](#_Toc100833164)

[ПРИЛОЖЕНИЕ 14](#_Toc100833165)

# ВВЕДЕНИЕ

В двадцать первом веке развитие медицины достигло абсолютно невообразимого ранее уровня. Человечество способно вылечить большинство из известных нам болезней, и более того, даже не допустить заболевание ими. Ещё каких то сто лет назад, такие заболевания как корь, сифилис, бешенство и оспа были смертельны для человека, служа практически смертным приговором. Однако, за последнюю сотню лет медицина шагнула настолько далеко вперед, что эти болезни не представляют практически никакой угрозы, при условии вовремя начатого лечения.

Однако, как же вовремя диагностировать заболевания? Ведь в большинстве случаев это позволяет избежать серьёзных последствий огромного количества недугов, предоставляет возможность победить болезнь до того, как она начнёт прогрессировать и серьёзно влиять на самочувствие и здоровье человека.

Ответом на этот вопрос стал технический прогресс. Жизнь человека круто изменилась с момента изобретения транзистора и появления первых компьютеров. Сегодня мы все находимся на связи благодаря мобильным телефонам, а огромное количество людей имеет возможность работать из дома используя персональные компьютеры. Было бы большой ошибкой не использовать все технические достижения человечества для решения самой серьёзной проблемы – заботе о здоровье человека и поднятия уровня медицины. Решив проблемы с физическим и ментальным здоровьем человека, мы освободим его от этих проблем и значительно увеличим производительность труда со всеми из этого вытекающими.

В результате работы над этой проблемой, медицина стала одной из самых технологичных отраслей. Теперь большую часть работы врачей выполняют всякого рода устройства, позволяющие диагностировать и лечить заболевания. Системы магнитно-резонансной томографии, электрокардиографии доступны для покупки любыми слоями общества, а своей простотой позволяют использовать их даже детям.

Некоторые устройства, такие как глюкометры, стали неотъемлемой частью жизни больных сахарным диабетом и страдающим от повышения уровня сахара в крови, позволяя им проверить уровень сахара практически в любом месте и очень быстро.

Стоит рассчитывать, что дальнейшее развитие технологий позволит человечеству ещё лучше контролировать состояние здоровья, сделав эти процедуры ещё удобнее и доступнее как можно большему количеству людей.

# ОБЗОР ЛИТЕРАТУРЫ

1.1 Информация о платформе Arduino

Arduino — это платформа для создания электроники своими руками. К печатной плате, которая является миниатюрным компьютером, можно подсоединять различные компоненты, например датчики, экраны, переключатели. Или даже другие платы со своими функциями.

«Мозг» любого конструктора Arduino — это собственно одноимённая плата. На ней есть процессор, модули памяти и порты ввода‑вывода, к которым подключаются другие компоненты.

В данном курсовом проекте будет использоваться версия платформы под названием Arduino Nano. Платформа Nano, построенная на микроконтроллере ATmega328 (Arduino Nano 3.0) или ATmega168 (Arduino Nano 2.x), имеет небольшие размеры и может использоваться для решения большого спектра задач.

Так как в данном курсовом проекте критично важна эргономичность и небольшие размеры в целом, то выбор Nano как основы проекта выглядит оправдано.

Данная версия платы имеет следующие характеристики:

Микроконтроллер: Atmel ATMega168 или ATMega328

Рабочее напряжение: 5B

Входное напряжение(рекомендуемое): 7-12B

Входное напряжение(предельное): 6-20B

Цифровые входы\выходы: 14

Аналоговые входы: 8

Постоянный ток через вход\выход: 40мА

Флеш-память: 32 Кб

ОЗУ: 2Кб

EEPROM: 1Кб

Тактовая частота: 16МГц

Размеры: 1.85х4.2 см

Данные характеристики могут вполне позволить нам разработать функционал, подходящий под наши потребности. Большое количество цифровых входов\выходов позволяют подключить множество дополнительного оборудования в виде датчиков.

Более подробно цифровые и аналоговые входы\выходы показаны на данном рисунке:

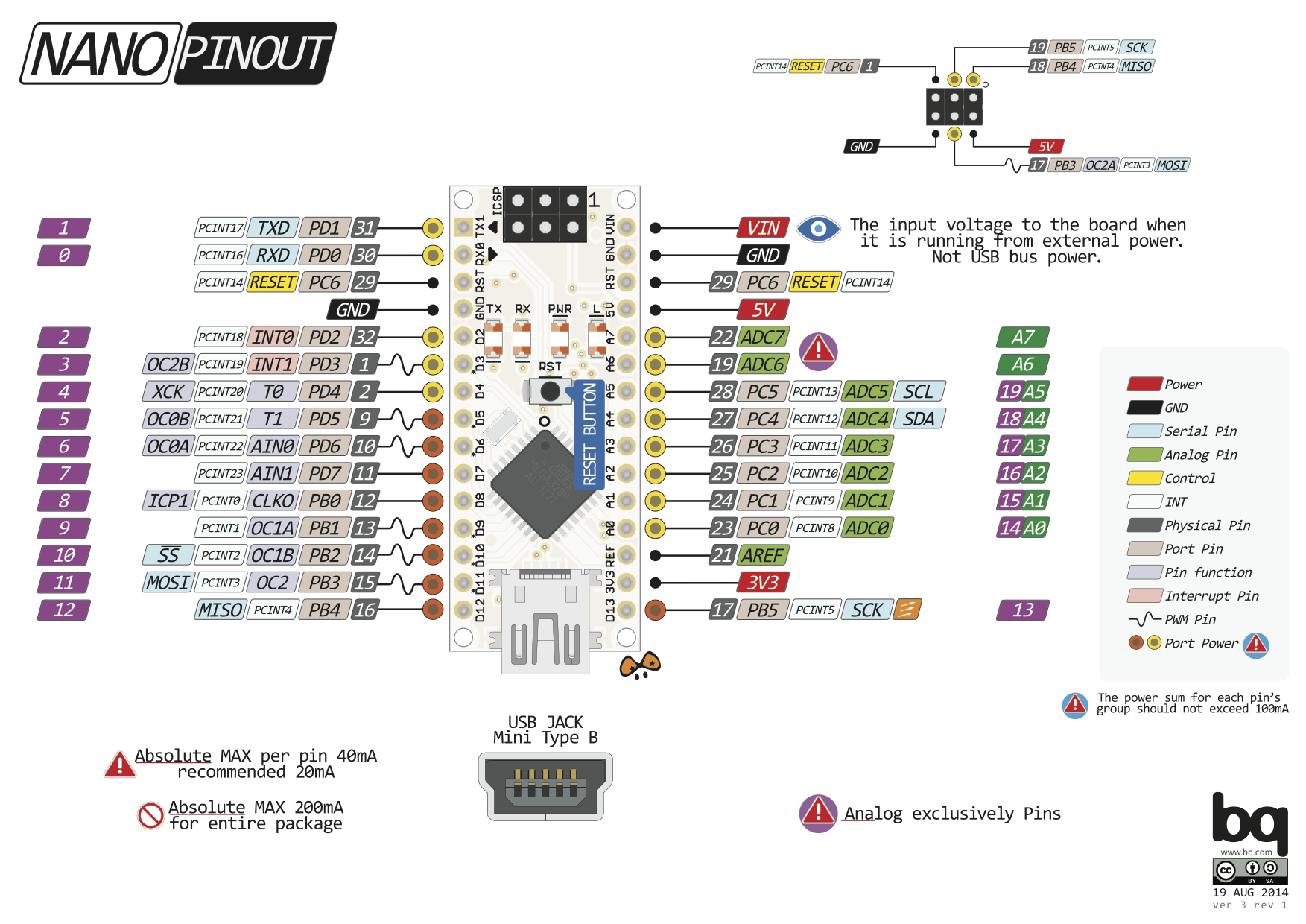


Рисунок 1.1 – Распиновка платы Arduino Nano

Так же стоит уделить внимание тем комплектующим, которые помогут в реализации данного курсового проекта:

**1.Датчик сердцебиения**

Датчик импульсов, с которым мы будем работать, представляет собой, фотоплетизмограф, который является хорошо известным медицинским устройством, используемым для мониторинга сердечного ритма.

Фотоплетизмограмма — метод регистрации кровяного потока с использованием источника инфракрасного или светового излучения и фоторезистора или фототранзистора.

Фоторезистор меняет сопротивление в зависимости от количества поглощённого света. Чем больше кровяной поток, тем меньше света поглощается в тканях организма, следовательно, больше света приходит на фоторезистор.

Фотоплетизмограмма позволяет измерять объёмный пульс крови, вызванный периодическим изменением кровяного объёма при каждом ударе сердца, частоту сердцебиения, вариабельность сердечного ритма.

Принцип действия датчика сердцебиения представлен на рисунке 1.2:

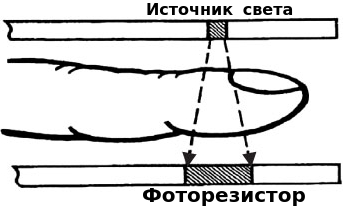


Рисунок 1.2 – Принцип действия датчика сердцебиения

Схема подключения датчика представлена на рисунке 1.3:

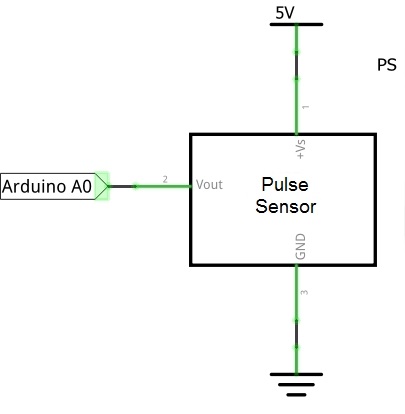


Рисунок 1.3 – Принцип подключения датчика сердцебиения

**2.Датчик температуры тела**

Используемый в нашем проекте датчик температуры MAX30205 имеет точность измерений 0.1°C в диапазоне от 37°C до 39°C. Он работает по протоколу I2C.

Датчик может работать с напряжением 5 или 3.3V. Тем не менее, его плата сконфигурирована для работы с напряжением 5V, но в своем составе датчик имеет схему понижения уровня до 3.3V поскольку часть его узлов работает с таким уровнем напряжения.

Этот датчик не является бесконтактным – то есть для его корректной работы нужен непосредственный контакт с телом человека.

**3.Датчик концентрации газа**

Датчик газа MQ-135 способен обнаруживать широкий диапазон различных газов в окружающем воздухе: NH3, NOx, алкоголь, бензол, дым и углекислый газ (CO2). Датчик MQ-135 можно купить как в виде модуля, так и в виде отдельного датчика. В нашем проекте мы будем использовать его в виде модуля для простоты подключения и замены в случае изменения функционала. Этот датчик отличается высокой скоростью восстановления, что позволяет проводить измерения когда в этом будет потребность.

Универсальность данного датчика сыграет важную роль в вариативности функционала конечного устройства, так как он позволяет измерять концентрацию в воздухе бензола, спирта и дыма, то есть теоретически возможно так же реализовать функционал алкотестера или системы оповещения о задымлении.

1.2 Анализ существующих аналогов

Сегодня устройствами медицинского назначения можно считать такие девайсы, как Apple Watch, Xiaomi Mi Band и Huawei Watch. Все они обеспечивают схожий функционал, и помимо функции измерения и показа времени позволяют узнать такие важные параметры о состоянии здоровье человека, как:

1. Слежка за физической активностью человека
2. Считать шаги
3. Считать калории
4. Измерять пульс
5. Записывать ЭКГ
6. Проверять кислород в крови
7. Следить за качеством сна
8. Контролировать уровень шума
9. Использовать экстренный вызов для обращения в службы спасения

Также довольно занятной функцией таких часов, как Apple Watch,

является контроль за качеством мытья рук. Устройство понимает когда человек занят этих процессом, и включает двадцатисекундный таймер. Это стало особенно полезно в пик COVID-19

Для коммерческих устройств расширение функционала и создание уникальных технических решений является ключевым фактором успеха устройства на рынке, однако, для данного курсового проекта для демонстрации умения применения теоретических знаний на практике будет достаточно реализовать лишь небольшое количество функций.

## 

## 1.3 Постановка задачи

После рассмотрения аналогов можно сказать, что все они обладают большим количеством функций, которые невозможно реализовать в курсовом проекте за данный период времени. Поэтому были выбраны несколько ключевых возможностей, которые будут выполнены в рамках одного семестра:

- Система должна иметь возможность отслеживать температуру тела человека.

- Система должна отслеживать пульс человека.

- Система должна замерять уровень вредных газов в воздухе.

- Вывод информации должен быть реализован на дисплей, или в виде Android приложения.

В качестве языка программирования выбран C++, так как он поддерживается платформой Arduino и средой разработки Arduino IDE , также будут использоваться библиотеки позволяющие считывать данные с датчиков. Данный список средств позволяет реализовать все задачи, выбранные для курсового проекта.

# 2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

После определения требований к функционалу разрабатываемого приложения его следует разбить на функциональные блоки. Такой подход упростит понимание проекта, позволит устранить проблемы в архитектуре, обеспечит гибкость и масштабируемость программного продукта в будущем путем добавления новых блоков.

## 2.1 Модуль инициализации

Модуль предоставляет доступ к функциям приложения. Его задача заключается в авторизации пользователей в приложении, определении имени и всех необходимых данных пользователя, зашедшего в приложение. А также первоначальной настройки пользовательского интерфейса.

## 2.2 Модуль работы с приложением

Модуль работы с приложением необходим для взаимодействия пользователя с приложением и является неотъемлемой его частью. Тут осуществляется весь необходимый функционал и возможности:

- Просмотр активных клиентов

- Просмотр полученных сообщений

- Отправка сообщений

- Просмотр параметров подключения

## 2.3 Модуль пользовательского интерфейса

Модуль пользовательского интерфейса предназначен для взаимодействия пользователя с приложением, основываясь на представлении и визуализации данных.

Для разрабатываемого проекта создается простейший пользовательский интерфейс с помощью библиотеки ncurses. Данный пользовательский интерфейс представляет собой набор пунктов меню, с которыми может взаимодействовать пользователь.

## 2.4 Модуль оперативного хранения данных

Модуль оперативного хранения данных необходим для организации временного хранения информационных ресурсов по мере работы с приложением. Информация о пользователе, который в данный момент работает с приложением, полученные и отправленные сообщения должны фиксироваться для непосредственной работы.

## 2.5 Модуль чтения и записи данных

Модуль чтения и записи данных необходим для взаимодействия с данными, которые хранит приложение, что очень важно при работе с проектом.

Данный модуль должен отвечать за считывание информации от пользователя при авторизации и оправки сообщений, её записи в используемую область памяти для дальнейшей работы, а также считывание и запись всей другой необходимой информации по мере работы с приложением.

## 2.6 Модуль сети

Модуль сети предназначен для подключения пользователей друг к другу и необходим для их взаимодействия в реальном времени.

Данный модуль должен отвечать за передачу сообщений по технологии peer-to-peer между пользователями и отслеживать активность клиентов в приложении.

# 3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

Данная глава будет представлять собой ключевой раздел, дающий понимание работы моего приложения, его структуру, с точки зрения описания отдельных функций и модулей, обработки данных, с приведением структурной диаграммы, вынесенной в приложение «А».

# 3.1 Описание структур приложения

Основными структурами программы являются User и sockaddr\_in.

Структура sockaddr\_in описывает сокет для работы с протоколами IP. Значение поля sin\_family всегда равно AF\_INET. Поле sin\_port содержит номер порта, который намерен занять процесс. Поле sin\_addr содержит IP адрес, к которому будет привязан сокет.

В структуре User находятся 3 поля, а именно:

- name – имя пользователя.

- isActive – поле для определения активен ли пользователь или отключен.

- структура sockaddr\_in address, в которой хранится адрес пользователя для взаимодействия с ним.

**3.2 Описание модулей и блоков программы**

Все вышеописанные структуры являются полностью бесполезными, без средств работы с ними, так называемыми «модулями», или функциями про-

граммы, которые содержат блоки кода. Рассматривать их следует от самого простого к самому сложному:

3.2.1 createSocket

Данная функция создает сокет, если открытие прошло неуспешно, то программа выдает ошибку и завершает работу.

3.2.2 setAddress

Данная функция получает на вход дескриптор сокета, к которому требуется привязать структуру структуру sockaddr\_in, при возникновении ошибки программа выводит ошибку на экран и завершает работу.

3.2.3 функции parseName, parseConnectAddress, parsePort

Данные функции я объединил в одну группу, так как они выполняют схожие задачи, а именно получают аргументы командной строки, такие как имя пользователя, IP адрес и порт. После их обработки и проверки на валидность сохраняют в переменные хранения для дальнейшего использования.

3.2.4 interface\_init

Инициализирует пользовательский интерфейс с выводом на экран первоначальной необходимой информацией.

3.2.5 updateInfoBox

Обновляет информацию о пользователе в верхнем разделе интерфейса.

3.2.6 updateUserBox

Функция проходит по массиву пользователей и обновляет информацию о подключенных пользователях к действительной сессии чата.

3.2.7 addUser

Получает на вход адрес и имя нового пользователя и добавляет его в массив ко всем существующим пользователям.

3.2.8 connectToUser

Функция, с помощью которой можно подключиться к активной сессии чата. Принимает адрес и имя нового пользователя, посылает пакет с информацией остальным пользователям на дачу разрешения для подключения, при его получении вызывает функцию addUser.

3.2.9 sendToSocket

Функция для передачи информации через сокет. На вход принимает аргументы такие, как дескриптор сокета, строку для отправки, ее размер и структуру sockaddr\_in. Сама отправка данных осуществляется с помощью встроенной функции sendto.

3.2.10 socketRead

Функция для чтения данных из сокета. На вход принимает аргументы такие, как дескриптор сокета, строку для заполнения полученной информацией и структуру sockaddr\_in. Само получение данных осуществляется с помощью встроенной функции recvfrom, которая возвращает количество полученных байт. Данная функция возвращает количество полученных байт.

4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

4.1 Выделение ключевых процедур

Ключевыми процедурами приложения можно назвать функцию connectToUser, sendToSocket, socketRead, addMessage, createPacket.

Все процедуры будут описаны в вышеперечисленном порядке.

4.1.1 Подключение к пользователю connectToUser

Функция получает на вход такие параметры, как дескриптор сокета, имя пользователя и структуру sockaddr\_in. Создается пакет на разрешение с помощью функции createConnectRequestPacket для подключения и отправляется другим пользователям с помощью функции sendToSocket, после этого ждет подтверждения. После этого другие пользователи должны отправить пакет с разрешением. При его получении с помощью функции socketRead проверяется действительно ли это пакет с разрешением, а не какой-то другой. Если все проверки прошли успешно, то вызывается функция addUser, которая добавляет нового пользователя в массив. После данных манипуляций, новый пользователь должен получить список уже активных пользователей, после чего функция завершает свою работу.

4.1.2 Пересылка сообщений в другой сокет sendToSocket

С помощью этой функции происходит передача информации между пользователями, поэтому она в списке основных, хоть и небольшая по размеру. Принимает на вход дескриптор сокета, структуру sockaddr\_in, строку для отправки и ее размер. Внутри используется функция sendto для пересылки сообщений в другой сокет, она является встроенной.

4.1.3 Чтение сообщения из сокета socketRead

С помощью этой функции получается информация из сокета. Принимает на вход дескриптор сокета, структуру sockaddr\_in, ее размер и строку для записи. Непосредственно чтение происходит с помощью встроенной функции recvfrom, которая возвращает количество полученных байт и записанных в строку. Данная функция socketRead возвращает данное количество прочитанных байт для дальнейшего его использования.

4.1.4 Выводит сообщения в основной блок интерфейса чата addMessage

Данная функция принимает на вход сообщение, которое необходимо отобразить. Она выделяет память под новое сообщение, после чего копирует его в массив и вызывает функцию updateMessageBox, которая в свою очередь обновляет список сообщений.

4.1.5 Создание пакета для передачи информации в сокет createPacket

Данная функция используется для передачи информации в сокет и последующей ее обработки. В первый байт строки записывается число, которое подразумевает определенный тип пакета с разной несущей в себе информацией и последующей обработкой.

**5 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ**

В данной программе присутствуют проверки на ошибки ввода флагов пользователем. На рисунках 5.1 и 5.2 показаны верные вводы для запуска приложения, а на рисунках 5.3, 5.4 и 5.5 показаны выводы ошибок на неправильный ввод.

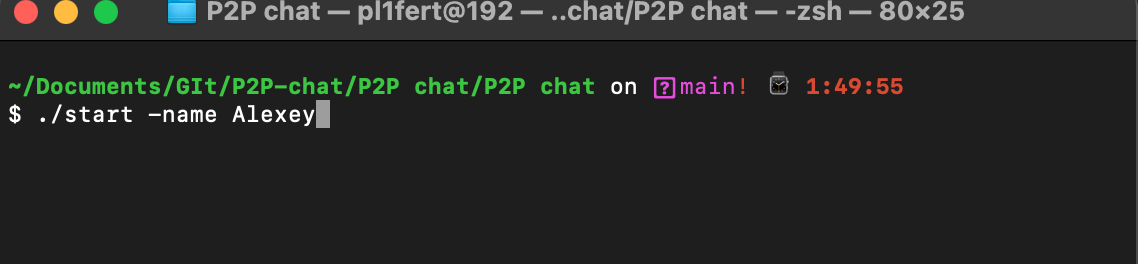
****

Рисунок 5.1 – Правильный первоначальный запуск

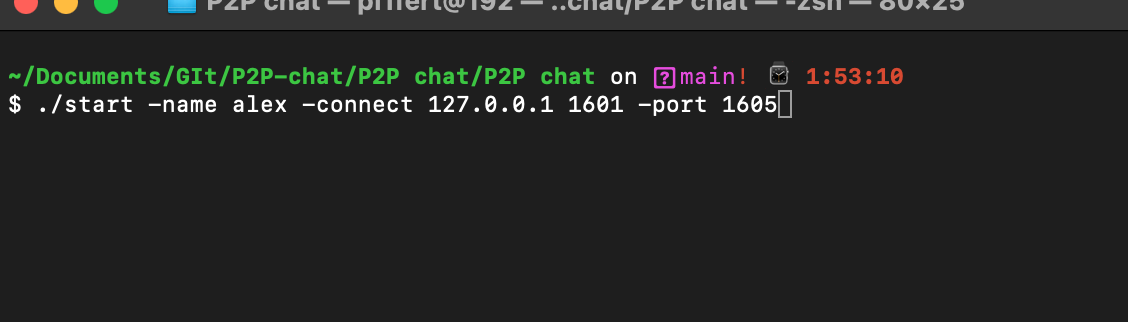
****

Рисунок 5.2 – Правильный запуск для подключения

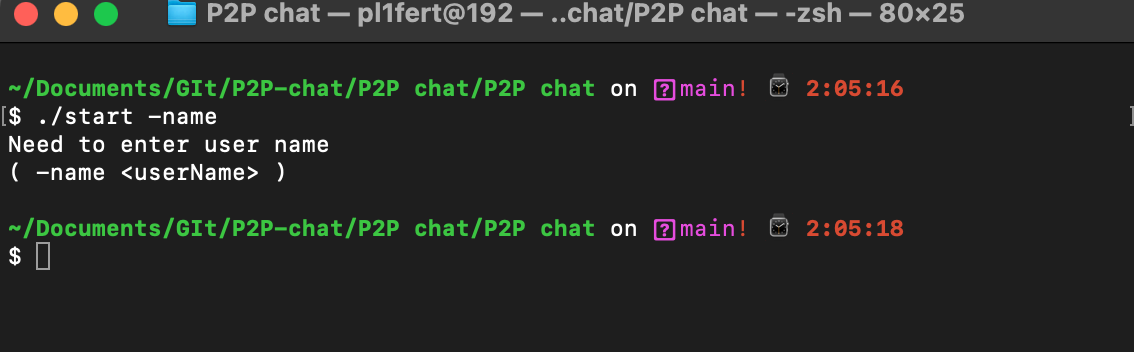
****

Рисунок 5.3 – Вывод ошибки при отсутствии имени

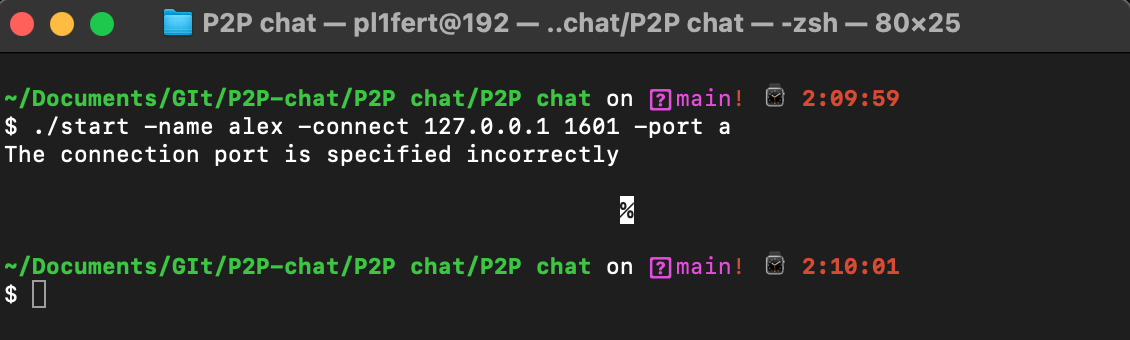
****

Рисунок 5.4 – Вывод ошибки при неправильном вводе порта

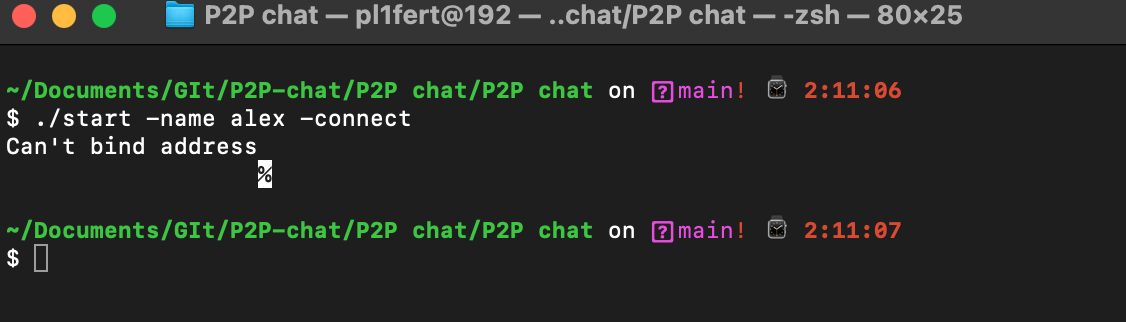
****

Рисунок 5.5 – Вывод ошибки при отсутствии адреса для подключения

**6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Для сборки данного проекта понадобятся утилиты git, gcc и make.

Установка данного приложения:

- https://github.com/Pl1Fert/P2P-chat

- cd P2P-chat

- make

Для его запуска потребуется прописать некоторые флаги, а именно:

-name <никнейм> - ваш никнейм в чате.

-connect <IP адрес сервера> <порт сервера> - используются для подключения к другому пользователю, вводить нужно его адрес и его порт.

-port <порт пользователя> - вводится номер порта для дальнейшего использования.

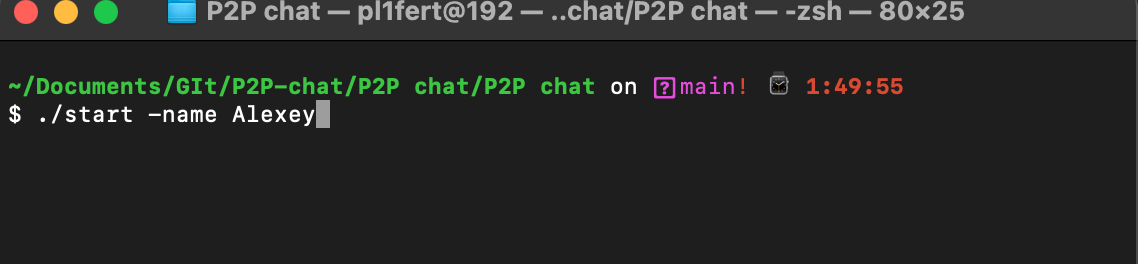
****

Рисунок 6.1 – Первоначальный запуск программы

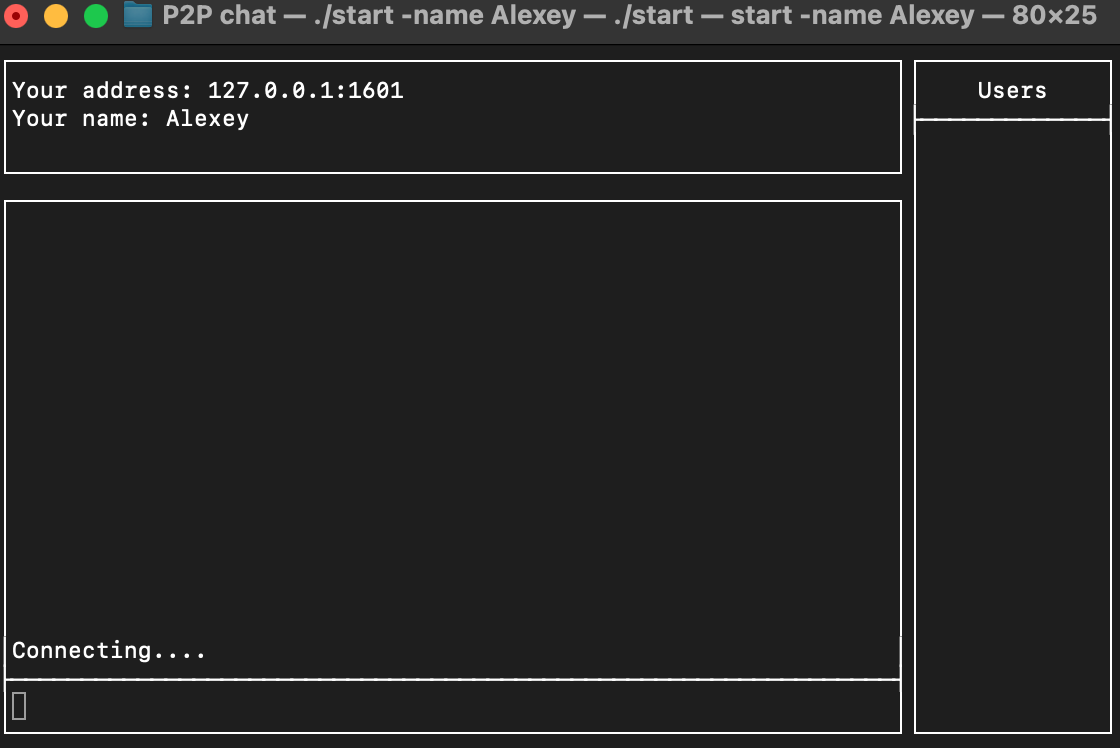
****

Рисунок 6.2 – Интерфейс программы

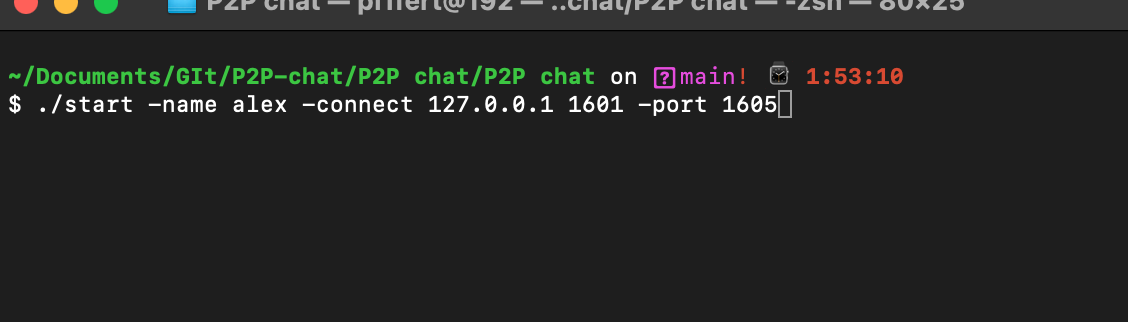
****

Рисунок 6.3 – Запуск программы для подключения

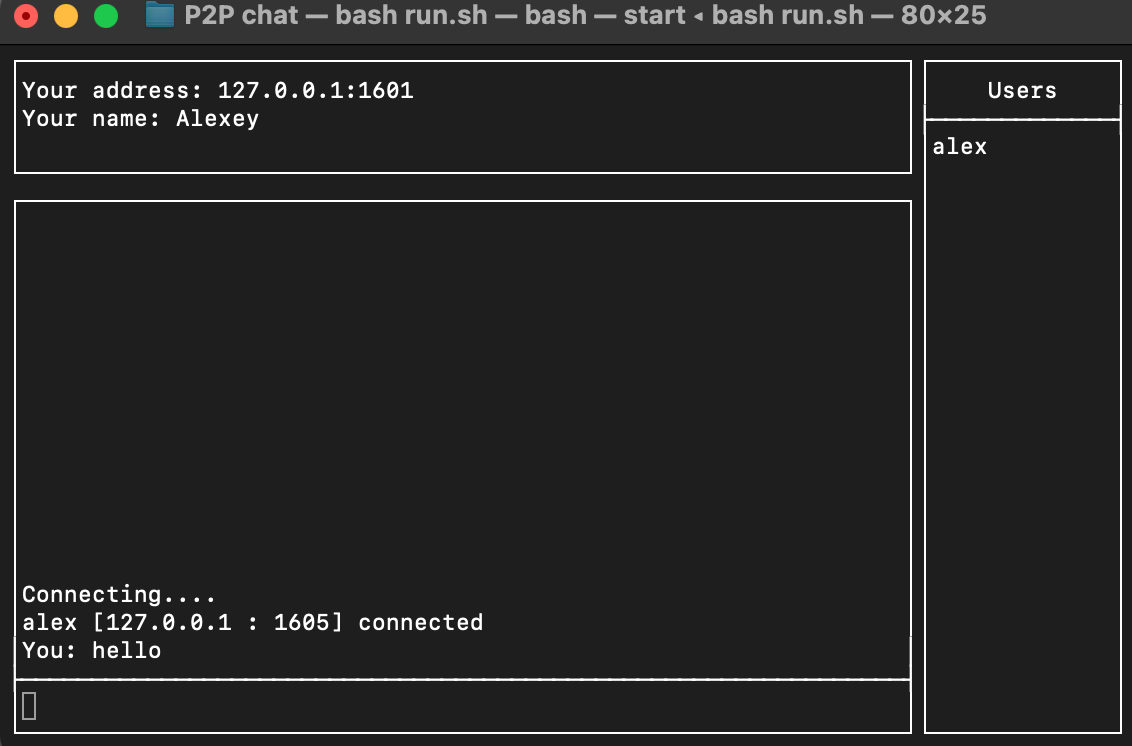
****

Рисунок 6.4 – Интерфейс программы с собеседниками

В данном окне приложения видны несколько секций, а именно: верхняя секция с указанием вашего имени и IP адреса, правая часть со списком активных пользователей в чате, средняя и самая главная часть этого приложения – это область чата с историей сообщений, и последняя и самая нижняя – поле для ввода вашего сообщения.

# ЗАКЛЮЧЕНИЕ

В результате работы над данным курсовым проектом было разработано работоспособное приложение со своим набором функций и графическим интерфейсом. Данный курсовой проект был разработан в соответствии с поставленными задачами, весь функционал был реализован в полном объеме.

Для создания программного продукта была подробно исследована технология peer-to-peer и принцип работы сетевого чата. В ходе разработки были углублены знания языка программирования C, а также получен опыт работы с библиотекой ncurses и с графическим интерфейсом.

Работа была разделена на такие этапы, как анализ существующих аналогов, литературных источников, постановка требований к проектируемому программному продукту, системное и функциональное проектирование, конструирование программного продукта, разработка программных модулей и тестирование проекта. После последовательного выполнения вышеперечисленных этапов разработки было получено исправно работающее приложение.

В дальнейшем планируется усовершенствование текущего функционала приложения, путем улучшения графического интерфейса, добавления новых функций и модулей, а также добавления возможности работы под разными системами.

# ЛИТЕРАТУРА

1. Лафоре, Р. Объектно-ориентированное программирование в C++ / Р. Лафоре. – СПб. : Питер, 2004.
2. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон; пер. с англ. – СПб. : ДМК, 2004.
3. Лав Р. Системное программирование на Linux/ 2-е издание 2014.
4. Керниган Б. Язык программирования С/ 4-е издание М.:Питер, 2004. – 923 с.
5. Рочкинд М. Программирование для UNIX, 2-е изд. СПб, БХВ-Петербург, 2005.
6. У. Ричард Стивенс, UNIX: разработка сетевых приложений/3-e издание 2007.

ПРИЛОЖЕНИЕ **А**

(обязательное)

**ПРИЛОЖЕНИЕ Б**

(обязательное)

**ПРИЛОЖЕНИЕ В**

(обязательное)

**ПРИЛОЖЕНИЕ Г**

(обязательное)