

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Схемотехника

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему

УСТРОЙСТВО КОНТРОЛЯ СОСТОЯНИЯ ВОДИТЕЛЯ АВТОПОЕЗДА

БГУИР КП 1-40 02 01 018 ПЗ

Студент: гр. 050541 Тимурзиев И. Ю.

Руководитель: доцент каф. ЭВМ,
Селезнев И. Л.

Минск 2023

Учреждение образования
«Белорусский государственный университет информатики
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ

(подпись)

«_____» _____ 2023 г.

ЗАДАНИЕ
по курсовому проектированию

Студенту Тимурзиеву Идрису Юнусовичу
(фамилия, имя, отчество)

1. Тема проекта Устройство контроля состояния водителя автопоезда

2. Срок сдачи студентом законченного проекта с 14.11.2023 по 20.11.2023

3. Исходные данные к проекту:

1. Микроконтроллер – напряжение питания 7-12 В, свободно используемых 14 программируемых входов/выходов, поддерживаемые интерфейсы: I2C, UART, SPI, PWM, 1-Wire
2. Источник питания – напряжение питания 12 В, минимальный выходной ток не менее 1 А
3. Датчик температуры – напряжение питания 5 В, диапазон температуры от -50 до +100 °С погрешность измерения $\pm 0,5$ °С, цифровой выходной сигнал
4. Датчик касания – напряжение питания 5 В, время отклика не менее 100 мс, цифровой выходной сигнал
5. Устройство отображения информация – напряжение питания 5 В, ожидаемый интерфейс I2C

4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов)

Введение.

1. Обзор литературы по теме устройства контроля состояния водителя автопоезда.

2. Разработка структуры устройства контроля состояния водителя автопоезда.

3. Обоснование выбора элементов функциональной схемы устройства контроля состояния водителя автопоезда.

4. Разработка принципиальной электрической схемы устройства контроля состояния водителя автопоезда.

5. Разработка программного обеспечения устройства контроля состояния водителя автопоезда.

Заключение.

Литература.

5. Перечень графического материала (с точным указанием обязательных чертежей)

1. Структурная схема устройства (формат А3)

2. Функциональная электрическая схема устройства (формат А3)

3. Принципиальная электрическая схема устройства (формат А3)

6. Консультант по проекту (с назначением разделов проекта) И.Л. Селезнёв

7. Дата выдачи задания 02.06.2023

8. Календарный график работы над проектом на весь период проектирования (с назначением сроков исполнения и трудоемкости отдельных этапов):

разделы 1,2 к 28.09 – 20 %;

раздел 3 к 13.10 – 20 %;

разделы 4 к 25.10 – 25 %;

раздел 5 к 09.11 – 20 %;

оформление пояснительной записки и графического материала к 13.11 – 15 %;

защита курсового проекта с 22.11 по 24.11.

РУКОВОДИТЕЛЬ доцент каф. ЭВМ Селезнёв И.Л.
(подпись)

Задание принял к исполнению 02.06.2023

(дата и подпись студента)

СОДЕРЖАНИЕ

| | |
|--|----|
| Введение..... | 4 |
| 1 Обзор литературы | 5 |
| 1.1 Структура разрабатываемого устройства..... | 5 |
| 1.2 Микроконтроллеры..... | 5 |
| 1.3 Устройства отображения информации..... | 6 |
| 1.4 Датчики температуры..... | 6 |
| 1.5 Датчики касания..... | 7 |
| 1.6 Датчики пульса..... | 7 |
| 1.7 Устройства индикации | 8 |
| 1.8 Органы управления..... | 8 |
| 1.9 Устройства звукового оповещения | 8 |
| 2 Разработка структуры устройства | 10 |
| 2.1 Определение функций устройства | 10 |
| 2.2 Определение компонентов устройства..... | 10 |
| 3 Обоснование выбора элементов функциональной схемы устройства | 12 |
| 3.1 Обоснование выбора микроконтроллера | 12 |
| 3.2 Обоснование выбора устройства отображения информации..... | 13 |
| 3.3 Обоснование выбора датчика температуры..... | 14 |
| 3.4 Обоснование выбора датчика касания..... | 15 |
| 3.5 Обоснование выбора датчика пульса..... | 16 |
| 3.6 Обоснование выбора устройства индикации | 16 |
| 3.7 Обоснование выбора органов управления | 16 |
| 3.8 Обоснование выбора устройства звукового оповещения..... | 17 |
| 4 Разработка принципиальной электрической схемы устройства | 18 |
| 4.1 Расчет потребляемой мощности устройства..... | 18 |
| 4.2 Микроконтроллер Arduino Uno | 18 |
| 4.3 Светодиод | 19 |
| 4.4 Датчик температуры DS18B20 | 20 |
| 4.5 Датчик касания TTP223B | 21 |
| 4.6 Датчик пульса Pulse Sensor | 22 |
| 4.7 Дисплей LCD1602 | 23 |
| 4.8 Пассивный зуммер KY-006..... | 23 |
| 4.9 Матричная клавиатура HW-834 | 24 |
| 5 Разработка программного обеспечения устройства | 25 |
| 5.1 Алгоритм работы устройства | 25 |
| 5.2 Схема программы..... | 25 |
| 5.3 Исходный текст программы..... | 26 |
| Заключение | 29 |
| Список использованных источников | 30 |

ВВЕДЕНИЕ

Целью курсового проекта является разработка устройства контроля состояния водителя автопоезда.

Основные функции, которые должно выполнять устройство:

- измерять температуру в кабине;
- измерять пульс машиниста;
- следить за тем, чтобы машинист не уснул;
- оповещать машиниста звуковым сигналом если он заснул;
- выводить информацию о физиологическом состоянии машиниста на устройство отображения информации.

Устройство будет оснащено 3 датчиками для мониторинга состояния водителя в кабине автопоезда, устройством звукового оповещения, органами управления, а также устройством отображения информации.

Первым датчиком будет датчик касания, который будет использоваться для отслеживания состояния засыпания водителя автопоезда. Если водитель долгое время не будет реагировать с датчиком касания, то в таком случае устройство звукового оповещения начнет воспроизводить специальный звуковой сигнал.

Дополнительно в устройстве будет установлен датчик температуры, предназначенный для мониторинга температуры в кабине автопоезда. Температура будет постоянно выводиться на устройстве отображения информации.

Для мониторинга физиологического состояния водителя будет использоваться датчик пульса. Однако в отличие от температуры, которая будет постоянно измеряться в кабине автопоезда, пульс будет измеряться по необходимости самим водителем.

Органы управления необходимы для переключения режимов работы устройства. Первый режим необходим для контроля состояния сна водителя при помощи датчика касания, а также для определения температуры в кабине автопоезда. Второй режим работы необходим исключительно для измерения пульса водителя.

Устройство должно улучшить безопасность водителей автопоезда, обеспечивая мониторинг и контроль их состояния во время поездок.

1 ОБЗОР ЛИТЕРАТУРЫ

1.1 Структура разрабатываемого устройства

Разрабатываемое устройство контроля состояния водителя автопоезда предназначено для улучшения безопасности машинистов, обеспечивая мониторинг и контроль их состояния во время поездок. Для решения этой задачи в состав устройства должны входить:

- микроконтроллер;
- устройство отображения информации;
- датчик касания;
- датчик пульса;
- датчик температуры;
- устройство индикации;
- органы управления;
- устройство звукового оповещения.

1.2 Микроконтроллеры

Микроконтроллеры – это специализированные интегральные микросхемы, предназначенные для управления электронными системами и устройствами. На сегодняшний день существует огромное множество различных микроконтроллеров, различающихся по размеру, архитектуре и функциональным возможностям. В таблице 1.1 перечислены наиболее распространенные на рынке микроконтроллеры, которые в большей степени отвечают задачам проектирования устройства [3, 4, 5].

Таблица 1.1 – Характеристики микроконтроллеров

| Микроконтроллер | ATmega328PB | STM8S003F3P6 | ATmega2560 |
|---------------------------|-------------------|------------------|------------------|
| Напряжение питания | 1,8 – 5,5 В | 2,95 – 5,5 В | 4,5 – 5,5 В |
| Флэш-память | 32 Кб | 8 Кб | 256 Кб |
| Разрядность | 8 бит | 8 бит | 8 бит |
| Макс. тактовая частота | 20 МГц | 16 МГц | 16 МГц |
| RAM | 2 Кб | 1 Кб | 8 Кб |
| EEPROM | 1 Кб | 0,128 Кб | 4 Кб |
| Архитектура | AVR | STM8 | AVR |
| Количество входов/выходов | 27 | 16 | 86 |
| Рабочая температура | От -40 до +105 °С | От -40 до +85 °С | От -40 до +85 °С |

АТmega328РВ применяется в разработке средств автоматизации и промышленных устройствах, обладает низким энергопотреблением. Широко используется в Arduino-сообществе. STM8S003F3P6 применяется в встраиваемых системах промышленности и управлении периферийными устройствами. АТmega2560 широко применяется в робототехнике, обладает множеством портов ввода и вывода.

1.3 Устройства отображения информации

Устройства отображения информации в электронных устройствах выполняют функцию визуализации данных и контента. В устройстве предполагается выводить только текстовую информацию, поэтому могут быть использованы ЖК-матрицы. В таблице 1.2 перечислены наиболее распространенные на рынке ЖК-дисплеи, которые в большей степени отвечают задачам проектирования устройства [6, 7, 8].

Таблица 1.2 – Характеристики дисплеев

| Дисплей | LCD1602 | ILI9341 TFT LCD | LCD2004 |
|--------------------|-------------------------|------------------|-------------------------|
| Напряжение питания | 5 В | 3,3 – 5 В | 5 В |
| Разрешение | 2 строки по 16 символов | 240x320 пикселей | 4 строки по 20 символов |
| Интерфейс | I2C | SPI | I2C |
| Графический | Нет | Да | Нет |
| Цветной | Нет | Да | Нет |

1.4 Датчики температуры

Датчики температуры – это устройства, предназначенные для измерения температуры в различных объектах или средах. Они обнаруживают изменения температуры и преобразуют их в электрические сигналы, которые могут быть интерпретированы и использованы для мониторинга. В таблице 1.3 перечислены наиболее распространенные на рынке датчики температуры, которые в большей степени отвечают задачам проектирования устройства [9, 10, 11].

Таблица 1.3 – Характеристики датчиков температуры

| Датчик | DS18B20 | GSMN DHT11 | LM35DZ |
|--------------------|-----------|------------|-----------|
| Напряжение питания | 3 – 5,5 В | 3,3 – 5 В | 4 – 30 В |
| Потребляемый ток | До 1 мА | До 2,5 мА | До 60 мкА |

Продолжение таблицы 1.3

| | | | |
|-----------------------|--|---|--|
| Погрешность измерения | $\pm 0,5\text{ }^{\circ}\text{C}$ | $\pm 2\text{ }^{\circ}\text{C}$ | $\pm 0,5\text{ }^{\circ}\text{C}$ при $25\text{ }^{\circ}\text{C}$ |
| Диапазон температур | От -55 до $+125\text{ }^{\circ}\text{C}$ | От 0 до $+50\text{ }^{\circ}\text{C}$ | От -55 до $+150\text{ }^{\circ}\text{C}$ |

1.5 Датчики касания

Датчики касания – это устройства, предназначенные для обнаружения физического контакта или прикосновения к поверхности. Они могут использоваться в различных технических системах и устройствах для определения наличия или отсутствия касания, а также для измерения силы или давления касания в некоторых случаях. В таблице 1.4 перечислены наиболее распространенные на рынке датчики касания, которые в большей степени отвечают задачам проектирования устройства [12, 13, 14].

Таблица 1.4 – Характеристики датчиков касания

| | | | |
|--------------------|------------|-------------|-------------|
| Датчик | TTP223B | MPR121 V12 | TTP224 |
| Напряжение питания | 2 – 5,5 В | 2,5 – 3,6 В | 2,4 – 5,5 В |
| Потребляемый ток | До 8 мА | До 29мА | До 13 мА |
| Время отклика | 60 мс | – | 100 мс |
| Размер | 24 x 24 мм | 30 x 20 мм | 35 x 29 мм |

1.6 Датчики пульса

Датчики пульса предназначены для мониторинга и анализа сердечного ритма человека. Их также можно использовать для биометрической аутентификации, предоставляя доступ только при определенном считываемом пульсе.

Среди датчиков пульса можно выделить MAX30102, он позволяет определять не только пульс, но и уровень насыщения крови кислородом. Помимо этого в сенсор строен температурный датчик для измерения температуры окружающей среды. Напряжение питания датчика 5 В.

Также стоит отметить Pulse Sensor Kit. Это компактный и простой в использовании датчик пульса, который специально разработан для мониторинга пульса человека. В отличие от MAX30102, Pulse Sensor Kit ориентирован исключительно на мониторинг сердечного ритма и не обладает дополнительной функциональностью, такой как измерение насыщения крови кислородом или встроенный температурный датчик. Напряжение питания датчика 3,3 – 5 В.

1.7 Устройства индикации

Устройства индикации используются для визуального отображения информации или состояния устройства. В проектируемом устройстве предлагается использование светодиодной индикации для отображения процесса запуска устройства.

1.8 Органы управления

Органы управления используются для регулирования и управления различными аспектами работы устройства. Для управления микроконтроллерами используются такие управляющие устройства, как кнопки и клавиатуры разного типа.

Кнопка – это минимальный элемент управления, который может использоваться для выполнения определенных действий или функций. В контексте электроники и устройств управления, одна кнопка может быть простым механическим выключателем, который при нажатии или удержании выполняет определенное действие. Например, кнопка может использоваться для включения или выключения устройства, выбора режима работы устройства.

Матричная клавиатура – это устройство, состоящее из сетки клавиш, обычно организованных в виде матрицы. Каждая клавиша представляет собой уникальную комбинацию строки и столбца, и, когда клавиша нажимается, происходит замыкание электрической цепи между соответствующей строкой и столбцом. Эта информация затем передается контроллеру клавиатуры, который определяет, какая клавиша была нажата. Матричные клавиатуры компактнее и могут содержать большее количество клавиш на ограниченной поверхности, что особенно важно в случае ограниченного пространства для органов управления.

1.9 Устройства звукового оповещения

Устройства звукового оповещения необходимы для воспроизведения звуковых сигналов. Недорогим устройством для воспроизведения простых звуковых сигналов является зуммер. Зуммер бывает активным и пассивным. Активный зуммер обычно включает в себя встроенный источник звука, такой как колеблющаяся мембрана или пьезокерамический элемент. Пассивные зуммеры используются для воспроизведения простых звуковых тональностей или гудков. Главное отличие активного зуммера от пассивного заключается в том, что активный зуммер генерирует звук самостоятельно, пассивный же зуммер требует источника сигнала. В таблице 1.5 перечислены наиболее распространенные на рынке устройства звукового оповещения, которые в большей степени отвечают задачам проектирования устройства [16, 17].

Таблица 1.5 – Характеристики устройств звукового оповещения

| Модуль | Активный зуммер ТМВ-12А05 | Пассивный зуммер КУ-006 |
|---------------------|------------------------------|----------------------------|
| Напряжение питания | 4 – 7 В | 1,5 – 15 В |
| Уровень шума | До 85 дБ | До 85 дБ |
| Потребляемый ток | До 30 мА | До 30 мА |
| Рабочая температура | От -20 до +60 °С | От -20 до +70 °С |

2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА

2.1 Определение функций устройства

Для того, чтобы составить структуру устройства контроля состояния водителя автопоезда необходимо выделить функции, которые будет выполнять устройство, затем определить компоненты, необходимые для реализации этих функций.

Функции, которые должно выполнять устройство:

- измерять температуру в кабине;
- измерять пульс машиниста;
- следить за тем, чтобы машинист не уснул;
- оповещать машиниста звуковым сигналом если он заснул;
- выводить информацию о физиологическом состоянии машиниста на устройство отображения информации.

2.2 Определение компонентов устройства

Проанализировав функции, которое должно выполнять устройство, определены следующие компоненты, необходимые для работы устройства:

1 Микроконтроллер – это ключевой компонент в электронном устройстве, отвечающий за обработку событий в реальном времени и управление внешними устройствами.

2 Блок питания отвечает за обеспечение необходимого питания для правильной работы микроконтроллера и всех подключенных к нему компонентов, предоставляет стабильное напряжение.

3 Органы управления необходимы для регулирования и управления различными аспектами работы устройства. В рамках проектируемого устройства это необходимо для измерения пульса машиниста.

4 Устройство индикации необходимо для визуального отображения информации или состоянии устройства. В рамках проектируемого устройства это необходимо для отображения процесса запуска устройства.

5 Устройство отображения информации играет ключевую роль в визуализации данных и контента, делая информацию более доступной и понятной. В рамках проектируемого устройства это необходимо для вывода информации о пульсе машиниста и температуры в кабине автопоезда.

6 Устройство звукового оповещения необходимо для воспроизведения звуковых сигналов. В рамках проектируемого устройства необходимо оповещать водителя звуком сигналом если он заснул.

7 Датчик касания – это устройство, которое реагирует на физическое касание или прикосновение. В рамках проектируемого устройства необходимо для отслеживания состояния засыпания водителя автопоезда. Водитель должен с определенной частотой реагировать с датчиком касания, чтобы

поддерживать свою активность, в противном случае он будет считаться заснувшим.

8 Датчик пульса предназначен для мониторинга сердечного ритма человека. В рамках проектируемого устройства пульс будет использоваться как один из основных показателей физиологического состояния машиниста автопоезда.

9 Датчик температуры – это устройство, предназначенное для измерения температуры в окружающей среде или ограниченном пространстве. В рамках проектируемого устройства датчик температуры необходим для измерения температуры в кабине автопоезда.

Структурная схема представлена в приложении А.

3 ОБОСНОВАНИЕ ВЫБОРА ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА

3.1 Обоснование выбора микроконтроллера

Рассматривая микроконтроллеры, представленные в таблице 1.1, можно сделать вывод, что ATmega328PB является более доступным вариантом по сравнению с ATmega2560, так как имеет более низкую стоимость. ATmega2560 в свою очередь является более мощным и функциональным по сравнению с ATmega328PB и STM8S003F3P6. Однако в рамках проекта он является чересчур мощным и подходит для более сложных проектов, которые требуют больше вычислительной мощности, более объемной памяти и большего количества периферийных устройств. Использование ATmega2560 будет излишней тратой ресурсов.

ATmega328PB основан на архитектуре AVR и имеет тактовую частоту до 20 МГц, STM8S003F3P6 основан на архитектуре STM8 и имеет тактовую частоту до 16 МГц, что может потребовать более тщательной оптимизации кода в некоторых случаях. ATmega328PB широко используется в Arduino-совместимых платформах, что обеспечивает обширное сообщество для поддержки и множество библиотек, которые будут полезны во время разработки программного обеспечения. STM8S003F3P6, конечно, тоже имеет свои инструменты разработки, однако они менее распространены и доступны.

Учитывая всё вышесказанное, самым оптимальным вариантом будет микроконтроллер ATmega328PB, изображенный на рис. 3.1.

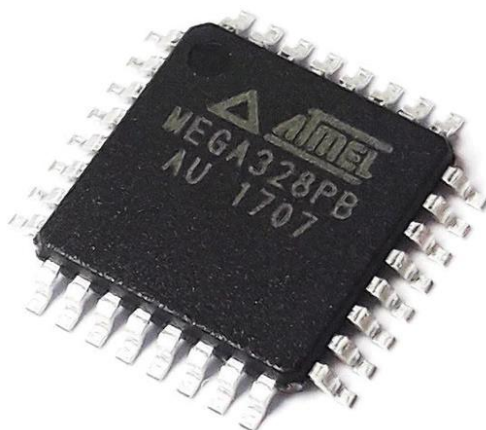


Рисунок 3.1 – Внешний вид ATmega328PB

Как и говорилось ранее, ATmega328PB широко используется в Arduino-совместимых платформах. Среди таких платформ можно выделить Arduino Nano и Arduino Uno, так как они оба используют ATmega328PB. Arduino Nano имеет меньший размер, но Arduino Uno имеет больше входов и выходов, поэтому для разработки устройства выбрана платформа Arduino Uno, изображенная на рис 3.2. Подробные характеристики платы в таблице 3.1.

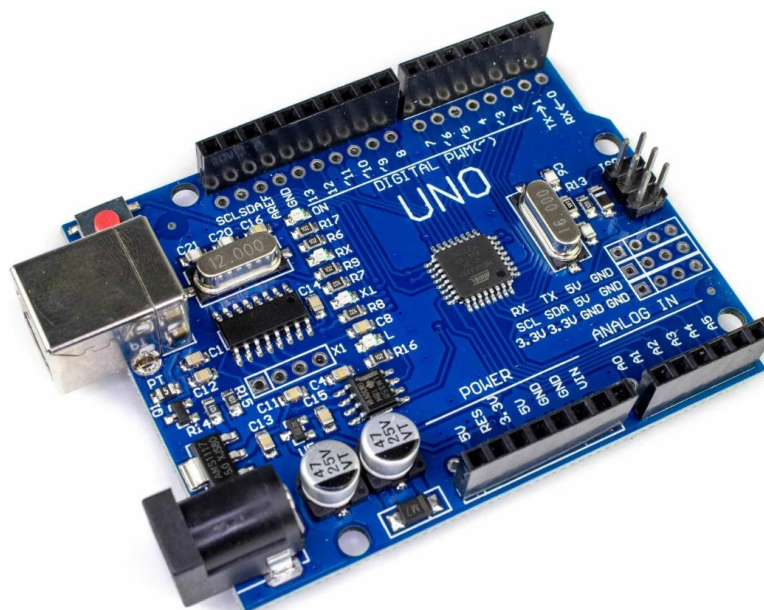


Рисунок 3.2 – Внешний вид Arduino Uno

Таблица 3.1 – Характеристики микроконтроллера Arduino Uno

| | |
|-----------------------|----------------------------------|
| Рабочее напряжение | 5 В |
| Входное напряжение | От 7 до 12 В |
| Цифровые входы/выходы | 14 (6 из которых могут быть ШИМ) |
| Аналоговые входы | 6 |
| Ток на выводе I/O | 40 мА |
| Ток на 3,3 В выводе | 50 мА |
| Тактовая частота | 16 МГц |
| Таймеры | 3 (1 x 16-битный, 2 x 8-битных) |

3.2 Обоснование выбора устройства отображения информации

Рассматривая устройства отображения информации, представленные в таблице 1.2, можно сделать вывод, что LCD1602 является более доступным вариантом по сравнению с ILI9341 TFT LCD, так как имеет более низкую стоимость. LCD1602 – это текстовый LCD-дисплей с двумя строками по 16 символов в каждой. Он предназначен для отображения текстовой информации. ILI9341 TFT LCD – это цветной TFT-дисплей с поддержкой графики и высоким разрешением, он может отображать как тест, так и графику, что делает его более универсальным. Однако в рамках проекта нет необходимости в сенсорном экране или же в поддержке графики, достаточно

текстовой информации. Использование ILI9341 TFT LCD будет излишней тратой ресурсов.

LCD1602 и LCD2004 различаются только количеством отображаемых символов. LCD1602 позволяет отображать 2 строки по 16 символов, LCD2004 позволяет же 4 строки по 20 символов. Для отображения пульса и температуры в кабине автопоезда будет более чем достаточно возможностей LCD1602, изображенного на рис. 3.3

3.3 Обоснование выбора датчика температуры

Рассматривая датчики температуры, представленные в таблице 1.3, можно сделать вывод, что наименее точным датчиком температуры является GSMN DHT11 с погрешностью измерения в ± 2 °C. При этом GSMN DHT11 обладает наименьшим диапазоном температур от 0 до +50 °C. Хотя датчик и позволяет помимо температуры, определять влажность, она не является необходимым показателем для работы устройства. В использовании GSMN DHT11 нет необходимости из-за его неконкурентоспособности на фоне других датчиков представленных в таблице.

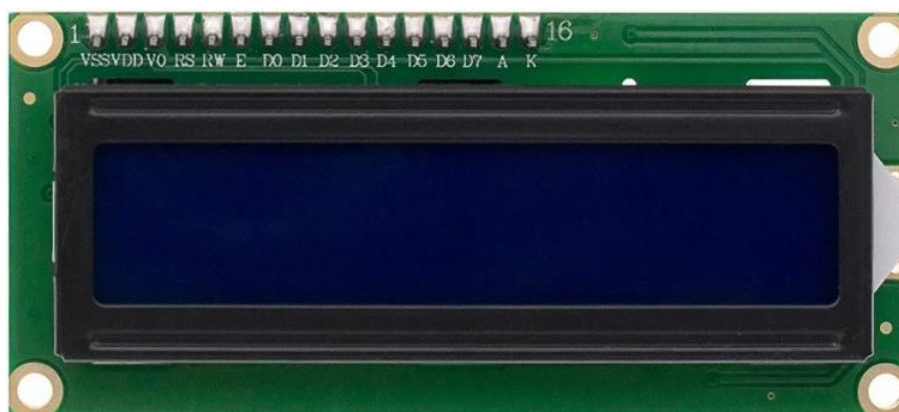


Рисунок 3.3 – Внешний вид дисплея LCD1602

Сравнивая датчики температуры DS18B20 и LM35DZ можно сказать, что у LM35DZ меньше максимальный потребляемый ток, но при этом более высокая погрешность измерения. Погрешность измерения у DS18B20 составляет $\pm 0,5$ °C, LM35DZ в свою же очередь может поддерживать такой уровень погрешности только при 25 °C. У LM35DZ чуть выше максимальный порог в диапазоне температур: от -55 до +150 °C. У DS18B20 максимальный порог в диапазоне чуть меньше: от -55 до +125 °C. Однако в рамках проекта разница между +125 °C и +150 °C не имеет никакого значения, потому датчик температуры необходим для измерения температуры в кабине автопоезда, где важна погрешность измерений.

Учитывая всё вышесказанное, самым оптимальным вариантом будет датчик температуры DS18B20, изображенный на рис. 3.4.

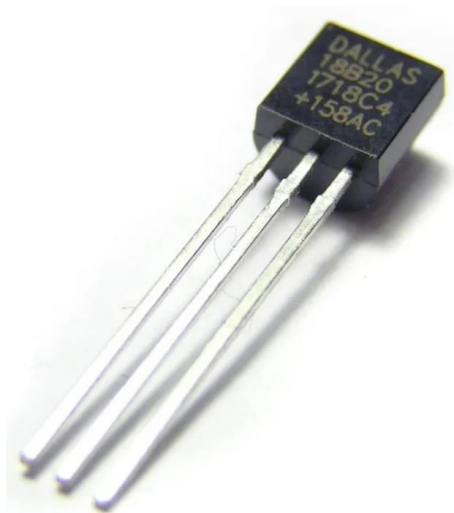


Рисунок 3.4 – Внешний вид DS18B20

3.4 Обоснование выбора датчика касания

Рассматривая датчики касания, представленные в таблице 1.4, можно сделать вывод, что самым компактным датчиком является ТТР223В. При этом у ТТР223В меньше максимальный потребляемый ток, а МРР121 v12 может потреблять больше всего тока. У ТТР224 время отклика хуже, чем у ТТР223В. Все датчики, описанные в таблице, являются емкостными сенсорными датчиками. Это означает, что у них схожий принцип работы, по этой причине сравнивать их нужно исключительно по данным описанным в таблице 1.4.

Учитывая всё вышесказанное, самым оптимальным вариантом будет датчик касания ТТР223В, изображенный на рис. 3.5.



Рисунок 3.5 – Внешний вид ТТР223В

3.5 Обоснование выбора датчика пульса

В разделе 1.6 представлены датчики пульса MAX30102 и Pulse Sensor. Преимущество MAX30102 в том, что он позволяет определять не только пульс, но и уровень насыщения крови кислородом. Помимо этого в сенсор строен температурный датчик для измерения температуры окружающей среды. Pulse Sensor – это компактный и простой в использовании датчик пульса, который специально разработан для мониторинга пульса человека. В отличие от MAX30102, Pulse Sensor ориентирован же исключительно на мониторинг сердечного ритма. Pulse Sensor был создан специально для Arduino, поэтому имеет удобную библиотеку, которая облегчит разработку программного обеспечения. MAX30102 хоть и может дополнительно измерять температуру окружающей среды, однако в этом нет необходимости, потому что в устройстве уже определен датчик температуры DS18B20, который сможет более точно определять температуру в кабине автопоезда.

Учитывая всё вышесказанное, самым оптимальным вариантом будет датчик пульса Pulse Sensor, изображенный на рис. 3.6.



Рисунок 3.6 – Внешний вид Pulse Sensor

3.6 Обоснование выбора устройства индикации

Для устройства индикации достаточно использовать четыре светодиода красного цвета. Необходимости в использовании RGB светодиодов нет, так как устройство индикации необходимо для демонстрации успешного запуска основного устройства, а для этого более чем достаточно обычных красных светодиодов.

3.7 Обоснование выбора органов управления

Рассматривая органы управления, представленные в разделе 1.8, можно сделать вывод, что наиболее оптимальным вариантом будет матричная

клавиатура HW-834. Матричная клавиатура – это устройство, состоящее из сетки клавиш, обычно организованных в виде матрицы. Конечно, как орган управления можно использовать и одну механическую кнопку, однако в таком случае орган управления ограничен в регулировании различных аспектов работы основного устройства. Матричная клавиатура изображена на рис 3.7.

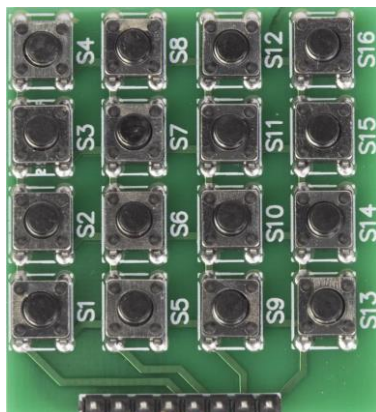


Рисунок 3.7 – Внешний вид матричной клавиатуры

3.8 Обоснование выбора устройства звукового оповещения

Рассматривая устройства звукового оповещения, представленные в таблице 1.5, можно сделать вывод, что пассивный зуммер является наиболее подходящим вариантом. Пассивный зуммер более гибкий по сравнению с активным зуммером и позволяет создавать различные звуки за счет управления длительностью и частотой звуковых сигналов. У активного зуммера частота звукового сигнала не может быть изменена программно и он способен воспроизводить только predetermined звук. Пассивный зуммер КУ-006 изображен на рис. 3.8.

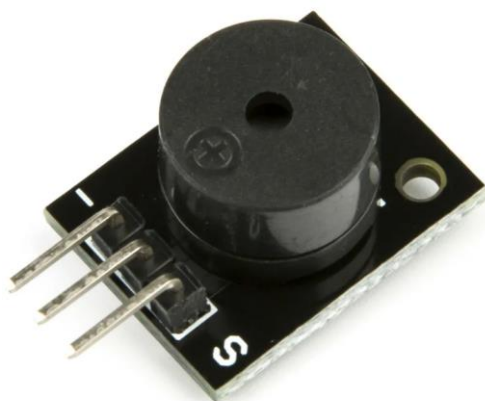


Рисунок 3.8 – Внешний вид КУ-006

4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА

4.1 Расчет потребляемой мощности устройства

Потребляемая мощность разрабатываемого устройства равна сумме мощностей, потребляемых его элементами. Расчет мощности элементов схемы устройства и самого устройства представлены в таблице 4.1.

Таблица 4.1 – Расчет мощности элементов схемы устройства

| Компонент | U, В | I, мА | P, мВт |
|--------------------------------|------|-------|--------|
| Микроконтроллер Arduino Uno | 5 | 24 | 120 |
| Дисплей LCD1602 | 5 | 120 | 600 |
| Датчик температуры DS18B20 | 5 | 1 | 5 |
| Датчик касания TTP223B | 5 | 6 | 30 |
| Датчик пульса Pulse Sensor Kit | 5 | 4 | 20 |
| Красный светодиод (4 шт) | 5 | 20 | 400 |
| Матричная клавиатура HW-834 | 5 | 10 | 50 |
| Пассивный зуммер KY-006 | 5 | 22 | 110 |
| Суммарная мощность, мВт | | | 1335 |

Потребляемая мощность одного элемента схемы устройства рассчитывается по формуле:

$$P = U \cdot I. \quad (4.1)$$

Таким образом, потребляемая мощность устройства равна:

$$P = 5 \cdot 24 + 5 \cdot 120 + \dots + 5 \cdot 22 = 1335 \text{ мВт} = 1,335 \text{ Вт}. \quad (4.2)$$

Учитывая поправочный коэффициент в 20%, то максимальная потребляемая мощность составит 1,602 Вт.

Рассчитаем потребляемый ток устройства:

$$I = \frac{P}{U} = \frac{1,602}{5} = 0,3204 \approx 0,32 \text{ А}. \quad (4.3)$$

4.2 Микроконтроллер Arduino Uno

Микроконтроллер Arduino Uno имеет 14 цифровых и 6 аналоговых входов и выходов, а также поддерживает следующие интерфейсы:

1 Serial (UART) – встроенный серийный интерфейс UART, который используется для связи с компьютером и другими устройствами. Этот интерфейс обычно используется через USB-порт на плате.

2 SPI – это быстрый серийный интерфейс, который позволяет устройствам общаться друг с другом. Arduino Uno предоставляет поддержку SPI через специальные выводы на цифровых контактах 10-13.

3 I2C – это двухпроводной серийный интерфейс, используемый для связи между различными периферийными устройствами и микроконтроллером. На Arduino Uno эти контакты находятся на аналоговых входах A4 (SDA) и A5 (SCL).

4 PWM – ШИМ используется для управления аналоговыми устройствами, такими как светодиоды и сервоприводы, с помощью цифровых сигналов. Arduino Uno предоставляет ШИМ на нескольких цифровых выводах.

В таблице 4.2 представлены подробные характеристики микроконтроллера Arduino Uno.

Таблица 4.2 – Характеристики микроконтроллера

| | |
|-----------------------|----------------------------------|
| Рабочее напряжение | 5 В |
| Входное напряжение | От 7 до 12 В |
| Цифровые входы/выходы | 14 (6 из которых могут быть ШИМ) |
| Аналоговые входы | 6 |
| Ток на выводе I/O | 40 мА |
| Ток на 3,3 В выводе | 50 мА |
| Тактовая частота | 16 МГц |
| Таймеры | 3 (1 x 16-битный, 2 x 8-битных) |

Датчик температуры DS18B20 подключается к цифровому входу D2, датчик касания TTP223B подключается к цифровому входу D3, датчик пульса Pulse Sensor подключается к аналоговому входу A0, дисплей LCD1602 подключается к аналоговым входам A4, A5, пассивный зуммер KY-006 подключается к цифровому выходу D5, матричная клавиатура HW-834 подключается к цифровым выходам D7 и D8 и к цифровому входу D6, светодиод подключается к цифровому выходу D12. Обозначение микроконтроллера на принципиальной схеме представлено на рис 4.1.

4.3 Светодиод

В проекте используется четыре светодиода красного цвета как устройства индикации. Для ограничения тока светодиода используется резистор номиналом, рассчитываемым по следующей формуле:

$$R = \frac{U_{\Pi} - U_{\text{Д}}}{I_{\text{ПР}}}, \quad (4.4)$$

где U_{Π} – напряжение питания, $U_{\text{Д}}$ – прямое напряжение светодиода, $I_{\text{ПР}}$ – ток светодиода.

| | | |
|-------|----------------|------|
| IOREF | Arduino Uno | SCL |
| 5V | | SDA |
| RESET | | AREF |
| 3.3V | | GND |
| 5V | | D13 |
| GND | | D12 |
| GND | | D11 |
| VIN | | D10 |
| A0 | | D9 |
| A1 | | D8 |
| A2 | | D7 |
| A3 | | D6 |
| A4 | | D5 |
| A5 | | D4 |
| | | D3 |
| | | D2 |
| | | D1 |
| | | D0 |
| | DD7 | |

Рисунок 4.1 – Обозначение микроконтроллера Arduino Uno

Рассчитываем номинал резистора для ограничения тока светодиода:

$$R = \frac{5 - 1}{0,02} = 200 \text{ Ом}. \quad (4.5)$$

Следовательно, чтобы светодиод не перегорел он должен быть подключен через резистор с сопротивлением не менее 200 Ом. Однако если взять слишком больше сопротивление, то светодиод будет гореть очень тускло, поэтому светодиод будет подключаться через резистор с сопротивлением 220 Ом. На рис. 4.2 представлена схема подключения светодиода.

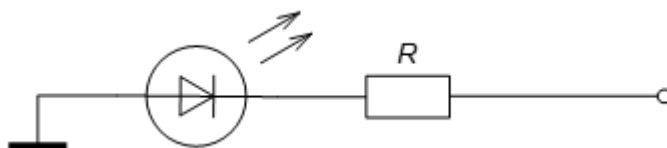


Рисунок 4.2 – Схема подключения светодиода

4.4 Датчик температуры DS18B20

Датчик температуры подключается к цифровому входу D2, питается датчик от напряжения 5 В. Используется резистор с сопротивлением 4,7 кОм

между линией данных и питанием, чтобы обеспечить стабильное чтение данных. В таблице 4.4 представлены подробные характеристики датчика температуры DS18B20.

Таблица 4.4 – Характеристики датчика температуры

| | |
|-------------------------------|---|
| Тип датчика | Цифровой |
| Потребляемый ток | До 1 мА |
| Напряжение питания | От 3,0 до 5,5 В |
| Диапазон измерения температур | От -55 до +125 °С |
| Поддержка паразитного питания | Питание осуществляется через линию данных, что позволяет использовать только 2 провода. |

Обозначение датчика температуры на принципиальной схеме представлено на рис 4.3.

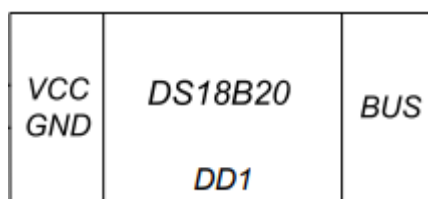


Рисунок 4.3 – Обозначение температуры DS18B20

Паразитное питание хоть и позволяет использовать только два провода (линия данных и земля), вместо трех в стандартной конфигурации (питание, земля и линия данных), но также имеет много недостатков.

Паразитное питание означает, что датчик получает энергию непосредственно от линии данных. Это может привести к нестабильности питания и негативно сказаться на точности измерений.

При низких температурах эффективность паразитного питания может снижаться, что делает его менее надежным для использования в холодной среде. Также время преобразования температуры может увеличиваться, что снижает скорость получения данных.

Таким образом при подключении датчика температуры не используется паразитное питание для обеспечения стабильной и надежной работы датчика.

4.5 Датчик касания ТТР223В

Датчик касания подключается к аналоговому входу А1, питается датчик от напряжения 5 В. В таблице 4.5 представлены подробные характеристики датчика касания ТТР223В.

Таблица 4.5 – Характеристики датчика касания

| Тип датчика | Цифровой |
|---------------------|------------------|
| Потребляемый ток | До 8 мА |
| Напряжение питания | От 2,0 до 5,5 В |
| Время отклика | 60 мс |
| Рабочая температура | От -20 до +70 °С |

Обозначение датчика касания ТТР223В на принципиальной схеме представлено на рис 4.4.



Рисунок 4.4 – Обозначение датчика касания ТТР223В

4.6 Датчик пульса Pulse Sensor

Датчик пульса подключается к аналоговому входу А1, питается датчик от напряжения 5 В. В таблице 4.6 представлены подробные характеристики датчика пульса Pulse Sensor.

Таблица 4.6 – Характеристики датчика пульса

| Тип датчика | Аналоговый |
|---------------------|------------------|
| Потребляемый ток | До 5 мА |
| Напряжение питания | От 3,3 до 5 В |
| Рабочая температура | От -40 до +85 °С |

Обозначение датчика пульса Pulse Sensor на принципиальной схеме представлено на рис 4.5.

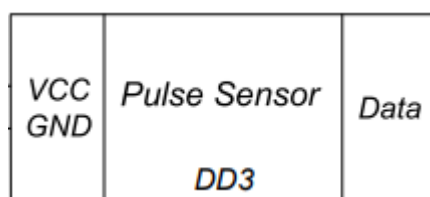


Рисунок 4.5 – Обозначение датчика пульса Pulse Sensor

4.7 Дисплей LCD1602

Дисплей подключается к аналоговым входам А4 и А5, питается дисплей от напряжения 5 В. В таблице 4.6 представлены подробные характеристики дисплея LCD1602.

Таблица 4.7 – Характеристики дисплея

| | |
|--------------------|-------------------------|
| Потребляемый ток | До 200 мА |
| Напряжение питания | 5 В |
| Интерфейс | I2C |
| Размер экрана | 16 символов на 2 строки |
| Размер символа | 5 x 8 пикселей |

Обозначение дисплея LCD1602 на принципиальной схеме представлено на рис 4.7.

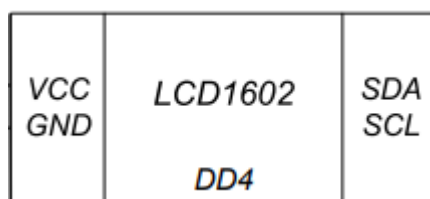


Рисунок 4.6 – Обозначение дисплея LCD1602

4.8 Пассивный зуммер KY-006

Пассивный зуммер подключается к цифровому входу D5, питается пассивный зуммер от напряжения 5 В. Вывод D5 поддерживает ШИМ, это означает, что можно контролировать тон и громкость звука, генерируемого зуммером, изменяя ширину импульса сигнала. Имея возможность точно контролировать длительность и частоту каждого импульса, можно создавать различные звуковые сигналы или мелодии.

В таблице 4.8 представлены подробные характеристики пассивного зуммера KY-006.

Таблица 4.8 – Характеристики пассивного зуммера

| | |
|---------------------|------------------|
| Потребляемый ток | До 30 мА |
| Напряжение питания | 1,5 – 15 В |
| Уровень шума | До 85 дБ |
| Рабочая температура | От -20 до +70 °C |

Обозначение пассивного зуммера KY-006 на принципиальной схеме представлено на рис 4.7.

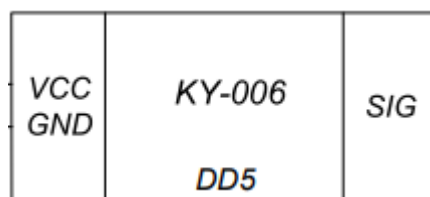


Рисунок 4.7 – Обозначение пассивного зуммера KY-006

4.9 Матричная клавиатура HW-834

Матричная клавиатура HW-834 подключается через 8-контактный разъем, однако так как нам нет необходимости в использовании всех 16 кнопок на матричной клавиатуре, будет подключено только 3 контакта. C1 и C2 подключается к цифровым выходам D7 и D8, а R1 к цифровому входу D6.

Обозначение матричной клавиатуры HW-834 на принципиальной схеме представлено на рис 4.8.

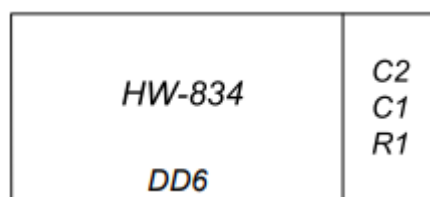


Рисунок 4.8 – Обозначение матричной клавиатуры HW-834

5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ УСТРОЙСТВА

5.1 Алгоритм работы устройства

Программа управления устройством контроля состояния водителя автопоезда должно выполнять следующие действия:

1 Инициализация – инициализация серийной связи между платой и другими устройствами USB для диагностики. Настройка выводов для светодиода, пассивного зуммера, датчика касания и остальных модулей, использующиеся в устройстве. Инициализация таймеров для асинхронного обновления, а также для контроля состояния сна водителя автопоезда. Создание объектов для управления матричной клавиатурой, дисплеем и остальными датчиками. Мигание светодиода с определенной частотой для демонстрации работоспособности устройства.

2 Смена режимов работы – в программе должно быть предусмотрено 2 режима работы: режим контроля сна и температуры, а также режим измерения пульса. Переключение режима работы устройства должно происходить за счет нажатия кнопок на матричной клавиатуре.

3 Контроль состояния сна водителя – программа должна использовать датчик касания для отслеживания активности водителя. Если водитель неактивен в течение 15 секунд, то в таком случае программа последовательно воспроизведет несколько звуковых сигналов с различной частотой до того момента пока водитель не переключит режим работы устройства либо же не начнет взаимодействовать с датчиком касания. Время, оставшееся до воспроизведения звукового сигнала, будет отображаться на первой строке дисплея.

4 Измерение пульса водителя – программа должна измерять пульса водителя в том случае, если она работает во 2 режиме работы (режим измерения пульса). При обнаружении начала удара сердца программа обновит и отобразит текущее значение пульса на первой строке дисплея.

5 Измерение температуры в кабине автопоезда – программа должна раз в секунду измерять температуру в кабине автопоезда с помощью датчика температуры и обновлять показания температуры на второй строке дисплея в том случае, если программа работает в 1 режиме работы (режим контроля состояния сна и температуры).

5.2 Схема программы

Рассмотрим схему программы управления устройством контроля состояния водителя автопоезда, представленную в приложении Г.

Блоки 2 – 3 отвечает за инициализацию входных и выходных значений, а также переменных и констант. В этих же блоках происходит инициализация серийной связи между платой и другими устройствами USB для диагностики,

настройка выводов для светодиода, пассивного зуммера и остальных модулей, используемых в устройстве. Инициализация таймеров для асинхронного обновления, а также для контроля состояния сна водителя автопоезда. Создание объектов для управления матричной клавиатурой, дисплеем и остальными датчиками.

Блок 4 – это начало бесконечного цикла в котором происходит основная работа всего устройства.

Блоки 5 – 8 отвечают получение символа нажатой клавиши на матричной клавиатуре и её дальнейшей обработки. В случае если символ нажатой клавиши равен 0, то режим измерения пульса выключается, в ином случае наоборот включается.

В 9 блоке проверка на режим работы устройства, если включен режим измерения пульса, то устройство будет измерять пульс, в противном случае оно будет измерять температуру в кабине автопоезда и контролировать состояние сна водителя.

Блоки 10 – 13 отвечают за получение пульса и вывода его на дисплей, задержка между вычислением составляет 20 мс. Пульс будет выведен только в том случае если было обнаружено начало удара сердца.

Блоки 14 – 19 отвечают за контроль состояния сна водителя автопоезда. Считывается состояние датчика касания, если датчик был нажат, то таймер сбрасывается. В 17 блоке проверяется, прошло ли 15 секунд с того момента, как водитель не взаимодействовал с датчиком касания. Если датчик касания был неактивен более 15 секунд, то в таком случае воспроизводится звуковое оповещение. Если же водитель взаимодействует с датчиком с периодичностью менее 15 секунд, то в таком случае на дисплей выводится оставшееся время до срабатывания звукового оповещения.

Блоки 20 – 23 отвечают за измерение температуры в кабине автопоезда. Каждую секунду считываются данные с датчика температуры, после чего температура выводится на вторую строку дисплея. Далее отправляется запрос на сбор данных с датчика температуры, чтобы при следующем запросе данные о температуре были готовы к чтению.

5.3 Исходный текст программы

Исходный текст программы работы устройства написан с использованием языка программирования Arduino Wiring и приведен в приложении Д.

Строки 013 – 063 отвечают за инициализацию переменных и констант, создание объектов для управления матричной клавиатурой, дисплеем и остальными датчиками. Описанные строки относятся к 2 блоку схемы данных.

В строках 067 – 093 описана функция `void setup()`, отвечающая за инициализацию серийной связи между платой и другими устройствами USB для диагностики, а также за настройку выводов для светодиода, пассивного зуммера и остальных модулей, используемых в устройстве. В этой же

функции происходит инициализация таймеров для асинхронного обновления, а также для контроля состояния сна водителя автопоезда. Описанные строки относятся к 3 блоку схемы данных.

В строках 098 – 105 описана функция `void loop()`, являющаяся основным циклом программы. Если включен режим измерения пульса, то устройство будет измерять пульс, в противном случае оно будет измерять температуру в кабине автопоезда и контролировать состояние сна водителя. Описанные строки относятся к 4, 9 и 23 блокам схемы данных.

В строках 110 – 122 описана функция `void readKeyboard()`, отвечающая за обработку ввода с матричной клавиатуры. В случае если символ нажатой клавиши равен 0, то режим измерения пульса выключается, в ином случае наоборот включается. Описанные строки относятся к 5 – 8 блокам схемы данных.

В строках 127 – 139 описана функция `void checkPulseState()`, отвечающая за регулярное измерение и отображение пульса. Функция получает пульс и выводит его на дисплей, задержка между вычислением пульса составляет 20 мс. Пульс будет выведен только в том случае если было обнаружено начало удара сердца. Описанные строки относятся к 10 – 13 блокам схемы данных.

В строках 144 – 167 описана функция `void checkSleepState()`, отвечающая за контроль состояния сна. Считывается состояние датчика касания, если датчик был нажат, то таймер проверки засыпания сбрасывается. Далее проверяется, прошло ли 15 секунд с того момента, как водитель не взаимодействовал с датчиком касания. Если датчик касания был неактивен более 15 секунд, то в таком случае воспроизводится звуковое оповещение. Если же водитель взаимодействует с датчиком с периодичностью менее 15 секунд, то в таком случае на дисплей выводится оставшееся время до срабатывания звукового оповещения. Описанные строки относятся к 14 – 19 блокам схемы данных.

В строках 172 – 186 описана функция `void checkTemperature()`, отвечающая за регулярное измерение и отображение температуры. Каждую секунду считываются данные с датчика температуры, после чего температура выводится на вторую строку дисплея. Далее отправляется запрос на сбор данных с датчика температуры, чтобы при следующем запросе данные о температуре были готовы к чтению. Описанные строки относятся к 20 – 23 блокам схемы данных.

В строках 191 – 207 описана функция `void warning()`, отвечающая за звуковое и световое уведомление. Воспроизводится звук частотой 300, 600 и 900 Гц и мигают светодиоды, задержка между воспроизведением звука 50 мс, задержка между миганием 100 мс. Описанные строки относятся к 18 блоку схемы данных.

В строках 212 – 222 описана функция `void blink()`, отвечающая за мигание светодиодов. Светодиоды мигают 6 раз с частотой одного мигания каждую 1 секунду. Описанные строки относятся к 3 блоку схемы данных.

В строках 227 – 232 описана функция `void reset()`, отвечающая за сброс информации при смене режима работы устройства. Очищается дисплей, сбрасывается таймер проверки засыпания водителя, а также если активирован режим измерения пульса, то на дисплее выводится сообщение об отсутствии пульса. Описанные строки относятся к 7 и 8 блокам схемы данных.

ЗАКЛЮЧЕНИЕ

В результате работы над курсовым проектом было разработано устройство контроля состояния водителя автопоезда, оснащенное 3 датчиками для мониторинга состояния машиниста автопоезда, пассивным зуммером, матричной клавиатурой и жидкокристаллическим дисплеем.

Разработанное устройство имеет 2 режима работы – режим измерения пульса и режим измерения температуры с контролем состояния сна. Режим работы переключается матричной клавиатурой.

Датчик касания используется для контроля состояния сна водителя. Если водитель долгое время не взаимодействует с датчиком касания, то в таком случае пассивный зуммер начнет воспроизводить специальный звуковой сигнал.

Дополнительно в устройстве установлен датчик температуры, предназначенный для мониторинга температуры в кабине автопоезда. Температура регулярно считывается и выводится на дисплей.

Для мониторинга физиологического состояния водителя используется датчик пульса. Пульс регулярно измеряется и выводится на дисплей в том случае, если активирован режим измерения пульса.

В дальнейшем планируется усовершенствование данного курсового проекта. Одним из таких улучшений является оптимизация алгоритма контроля состояния сна машиниста, добавление датчика насыщения кислородом крови и датчика давления, а также создание более удобного пользовательского интерфейса.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Вычислительные машины, системы и сети: дипломное проектирование (методическое пособие) [Электронный ресурс] : Минск БГУИР 2019. – Электронные данные. – Режим доступа: https://www.bsuir.by/m/12_100229_1_136308.pdf.
- [2] Документация Arduino [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docs.arduino.cc/>.
- [3] Микроконтроллер ATmega328PB [Электронный ресурс] : Datasheet. – Режим доступа: <https://www.promelec.ru/fs/sources/a7/17/d1/79/dc5796a6110ff25cb8344d15.pdf>.
- [4] Микроконтроллер STM8S003F3P6 [Электронный ресурс] : Datasheet. – Режим доступа: <https://www.st.com/resource/en/datasheet/stm8s003f3.pdf>.
- [5] Микроконтроллер ATmega2560 [Электронный ресурс] : Datasheet. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/107092/ATMEL/ATMEGA2560.html>.
- [6] Дисплей LCD1602 [Электронный ресурс] : Datasheet. – Режим доступа: https://supereyes.ru/img/instructions/simvolnyy_displey_lcd1602_datasheet.pdf.
- [7] Дисплей ILI9341 TFT LCD [Электронный ресурс] : Datasheet. – Режим доступа: https://supereyes.ru/img/instructions/ILI9341_ILITEK_datasheet_V1.02.pdf.
- [8] Дисплей LCD2004 [Электронный ресурс] : Datasheet. – Режим доступа: https://handsontec.com/dataspecs/I2C_2004_LCD.pdf.
- [9] Датчик температуры DS18B20 [Электронный ресурс] : Datasheet. – Режим доступа: https://download.mikroe.com/documents/datasheets/DS18B20_datasheet.pdf.
- [10] Датчик температуры GSMN DHT11 [Электронный ресурс] : Datasheet. – Режим доступа: <https://robocraft.ru/files/datasheet/DHT11.pdf>.
- [11] Датчик температуры LM35DZ [Электронный ресурс] : Datasheet. – Режим доступа: <https://make.net.za/wp-content/datasheets/TI%20LM35DZ.pdf>.
- [12] Датчик касания TTP223B [Электронный ресурс]. – Режим доступа: <https://datasheethub.com/ttp223b-capacitive-touch-sensor-module>.
- [13] Датчик касания MPR121 V12 [Электронный ресурс]. – Режим доступа: <https://www.sparkfun.com/datasheets/Components/MPR121.pdf>.
- [14] Датчик касания TTP224 [Электронный ресурс] : Datasheet. – Режим доступа: <https://datasheet-pdf.com/PDF/TTP224-Datasheet-TONTEK-835811>.
- [15] Датчик пульса MAX30102 [Электронный ресурс] : Datasheet. – Режим доступа: https://download.mikroe.com/documents/datasheets/MAX30102_datasheet.pdf.

[16] Датчик пульса Pulse Sensor Kit [Электронный ресурс] : Datasheet. – Режим доступа: <https://pulsesensor.com/pages/two-or-more-pulse-sensors>.

[17] Активный зуммер ТМВ-12А05 [Электронный ресурс]. – Режим доступа: <https://www.estudioelectronica.com/wp-content/uploads/2018/09/SHT-TMB12A05.pdf>.

[18] Пассивный зуммер КУ-006 [Электронный ресурс]. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/1284499/JOY-IT/KY-006.html>.

ПРИЛОЖЕНИЕ А
(обязательное)
Схема структурная

ПРИЛОЖЕНИЕ Б
(обязательное)
Схема функциональная

ПРИЛОЖЕНИЕ В
(обязательное)
Схема принципиальная

ПРИЛОЖЕНИЕ Г
(обязательное)
Схема программы

ПРИЛОЖЕНИЕ Д
(обязательное)
Исходный текст программы

```
001 /*
002   Гр. 050541, Тимурзиев Идрис Юнусович
003   Проект: Микроконтроллерное устройство контроля состояния водителя автопоезда
004 */
005 #define USE_ARDUINO_INTERRUPTS true
006 #include <Wire.h>
007 #include <Keypad.h>
008 #include <OneWire.h>
009 #include <DallasTemperature.h>
010 #include <LiquidCrystal_I2C.h>
011 #include <PulseSensorPlayground.h>
012
013 #define ONE_WIRE_BUS 2
014 #define SENSOR_PIN 3
015 #define ZOOMER_PIN 5
016 #define LED_PIN_MIN 9
017 #define LED_PIN_MAX 12
018 #define PULSE_PIN A0
019 #define PULSE_THRESHOLD 550
020 #define KEYBOARD_ROWS 1
021 #define KEYBOARD_COLS 2
022
023 // Установка OneWire для связи
024 OneWire oneWire(ONE_WIRE_BUS);
025 // Передача ссылки OneWire на Dallas Temperature.
026 DallasTemperature sensors(&oneWire);
027
028 // Определение выводов строк и столбцов клавиатуры
029 const byte keyboardRowPins[KEYBOARD_ROWS] = { 6 };
030 const byte keyboardColPins[KEYBOARD_COLS] = { 7, 8 };
031 // Определение символов клавиш клавиатуры
032 const char keyboardKeys[KEYBOARD_ROWS][KEYBOARD_COLS] = {
033   { '0', '1' }
034 };
035
036 // Создание объекта клавиатуры
037 Keypad keypad = Keypad(
038   makeKeymap(keyboardKeys),
039   keyboardRowPins,
040   keyboardColPins,
041   KEYBOARD_ROWS,
042   KEYBOARD_COLS);
043
044 // Создание объекта LCD-дисплея
045 // Адрес I2C: 0x27, кол-во столбцов: 16, кол-во строк: 2
046 LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```

047 // Используется для отсчета времени до звукового сигнала
048 unsigned long previousSleepMillis = 0;
049 const long sleepInterval = 15000;
050
051 // Используется для асинхронного обновления оставшегося времени
052 unsigned long lastTimeRequest = 0;
053 const long timeInterval = 100;
054
055 // Используется для асинхронного обновления температуры
056 unsigned long lastTemperatureRequest = 0;
057 const long temperatureInterval = 1000;
058
059 // Режим измерения пульса
060 bool pulseMode = false;
061
062 // Создание объекта для работы с пульсовым датчиком
063 PulseSensorPlayground pulseSensor;
064
065 // Функция инициализации настроек
066 void setup() {
067     // Инициализация серийной связи между платой и другими устройствами через USB
068     Serial.begin(9600);
069     // Настройка выводов
070     for (int ledPin = LED_PIN_MIN; ledPin <= LED_PIN_MAX; ledPin++) {
071         pinMode(ledPin, OUTPUT);
072     }
073
074     pinMode(ZOOMER_PIN, OUTPUT);
075     pinMode(SENSOR_PIN, INPUT);
076     // Настройка пульсового датчика
077     pulseSensor.analogInput(PULSE_PIN);
078     pulseSensor.setThreshold(PULSE_THRESHOLD); // Установка порога
079     if (!pulseSensor.begin()) {
080         Serial.println("PulseSensor not begined");
081     }
082
083     // Настройка датчика температуры
084     sensors.begin();
085     // 9 бит: точность до 0.5°C, время измерения около 93.75 мс.
086     sensors.setResolution(9); // Установка разрешения 9 бит для DS18B20
087     sensors.requestTemperatures(); // Запрос на обновление температуры
088     blink();
089     // Настройка дисплея
090     lcd.init();
091     lcd.backlight();
092     // Сброс таймера
093     previousSleepMillis = millis();
094 }
095
096 // Основной цикл программы
097 void loop() {

```

```

098 readKeyboard();
099 // Если измеряю пульс
100 if (pulseMode) {
101     checkPulseState();
102 } else {
103     checkSleepState();
104     checkTemperature();
105 }
106 }
107
108 // Функция обработки ввода с клавиатуры
109 void readKeyboard() {
110     // Возвращает символ нажатой клавиши
111     char key = keypad.getKey();
112     if (key) {
113         if (key == '0') {
114             if (pulseMode) {
115                 pulseMode = false;
116                 reset();
117             }
118             } else if (!pulseMode) {
119                 pulseMode = true;
120                 reset();
121             }
122         }
123     }
124
125 // Функция регулярного измерения и отображения пульса
126 void checkPulseState() {
127     // Проверяем начало удара сердца
128     if (pulseSensor.sawStartOfBeat()) {
129         // Получаем удары в минуту
130         int pulse = pulseSensor.getBeatsPerMinute();
131         // Выводим показания на дисплей
132         lcd.clear();
133         lcd.setCursor(0, 0);
134         lcd.print("Pulse: ");
135         lcd.print(pulse);
136         lcd.print(" BPM");
137     }
138
139     delay(20);
140 }
141
142 // Функция контроля состояния сна
143 void checkSleepState() {
144     // Читаем состояние датчика касания
145     int currentSensorState = digitalRead(SENSOR_PIN);
146     // Если датчик нажат
147     if (currentSensorState == HIGH) {
148         // Сброс таймера

```

```

149     previousSleepMillis = millis();
150 }
151
152 unsigned long currentMillis = millis();
153 // Проверяем, прошло ли 15 секунд с последнего нажатия кнопки
154 if (currentMillis - previousSleepMillis >= sleepInterval) {
155     // Если прошло 15 секунд, издаем звуковой сигнал
156     warning();
157 } else if (currentMillis - lastTimeRequest >= timeInterval) {
158     lastTimeRequest = currentMillis;
159     // Рассчитываем оставшееся время
160     long remainingTime = round(
161         (float)(sleepInterval - (currentMillis - previousSleepMillis)) / 1000);
162     // Выводим оставшееся время на дисплей
163     lcd.setCursor(0, 0);
164     lcd.print("Time left: ");
165     lcd.print(remainingTime);
166     lcd.print(" s ");
167 }
168 }
169
170 // Функция регулярного измерения и отображения температуры
171 void checkTemperature() {
172     unsigned long currentMillis = millis();
173     if (currentMillis - lastTemperatureRequest >= temperatureInterval) {
174         // Сохранение текущего времени
175         lastTemperatureRequest = currentMillis;
176
177         // Считывание и вывод температуры
178         float temperatureC = sensors.getTempCByIndex(0);
179         lcd.setCursor(0, 1);
180         lcd.print("TEMP: ");
181         lcd.print(temperatureC);
182         lcd.print(" C");
183
184         // Запрос на обновление температуры
185         sensors.requestTemperatures();
186     }
187 }
188
189 // Функция звукового и светового оповещения
190 void warning() {
191     for (int ledPin = LED_PIN_MIN; ledPin <= LED_PIN_MAX; ledPin++) {
192         digitalWrite(ledPin, HIGH);
193     }
194
195     // Воспроизведение звука частотой 300, 600, 900 Гц
196     tone(ZOOMER_PIN, 300);
197     delay(50);
198     tone(ZOOMER_PIN, 600);
199     delay(50);

```



```
200   for (int ledPin = LED_PIN_MIN; ledPin <= LED_PIN_MAX; ledPin++) {
201       digitalWrite(ledPin, LOW);
202   }
203
204   tone(ZOOMER_PIN, 900);
205   delay(50);
206   // Остановка воспроизведения звука
207   noTone(ZOOMER_PIN);
208 }
209
210 // Функция мигания светодиода
211 void blink() {
212     for (int i = 0; i < 12; i++) {
213         for (int ledPin = LED_PIN_MIN; ledPin <= LED_PIN_MAX; ledPin++) {
214             digitalWrite(ledPin, !digitalRead(ledPin));
215         }
216
217         delay(500);
218     }
219
220     for (int ledPin = LED_PIN_MIN; ledPin <= LED_PIN_MAX; ledPin++) {
221         digitalWrite(ledPin, LOW);
222     }
223 }
224
225 // Функция сброса информации при смене режима работы устройства
226 void reset() {
227     lcd.clear();
228     previousSleepMillis = millis();
229     if (pulseMode) {
230         lcd.setCursor(0, 0);
231         lcd.print("No pulse...");
232     }
233 }
```

ПРИЛОЖЕНИЕ Е
(обязательное)
Перечень элементов

ПРИЛОЖЕНИЕ Ж
(обязательное)
Ведомость документов