

Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

К ЗАЩИТЕ ДОПУСТИТЬ

_____ И.Л.Селезнёв

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту по дисциплине
«Схемотехника»
на тему

УСТРОЙСТВО УПРАВЛЕНИЯ ОБЪЕКТОМ
БГУИР КП 1-40 02 01 06 ПЗ

группа 050541

Студент

А.Э.Лейкина

Руководитель

И.Л.Селезнёв

Минск 2023

Учреждение образования
«Белорусский государственный университет информатики
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ

(подпись)

« _____ » _____ 2023 г.

З А Д А Н И Е
по курсовому проектированию

Студенту Лейкиной Александре Эдуардовне
(фамилия, имя, отчество)

1. Тема проекта Устройство управления объектом

2. Срок сдачи студентом законченного проекта с 14.11.2023 по 20.11.2023

3. Исходные данные к проекту:

1. Микроконтроллер
2. Датчик движения, угол обзора не менее 100 градусов, ИК, тип сигнала цифровой
3. Акустический датчик, чувствительность не менее 20 дБ, диапазон частот 20 кГц
4. Датчик освещенности, диапазон не менее 50000 люкс, тип сигнала цифровой, интерфейс I2C
5. Устройство отображения информации
6. Орган управления

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

Введение

1. Обзор литературы по устройству управления объектом

2. Разработка структуры устройства управления объектом

3. Обоснование выбора элементов функциональной схемы устройства управления объектом

4. Разработка принципиальной электрической схемы устройства управления объектом

5. Разработка программного обеспечения

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)

4. Схема алгоритма программы устройства

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 %;

раздел 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.....	5
1 ОБЗОР ЛИТЕРАТУРЫ	6
1.1 Структура разрабатываемого устройства	6
1.2 Микроконтроллер.....	6
1.3 Датчик движения.....	8
1.4 Акустический датчик.....	10
1.5 Датчик освещенности	11
2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА УПРАВЛЕНИЯ ОБЪЕКТОМ	12
2.1 Проектирование устройства управления объектом.....	12
2.2 Микроконтроллер.....	12
2.3 Датчик движения.....	12
2.4 Акустический датчик.....	12
2.5 Датчик освещенности	13
2.6 Органы управления.....	13
2.7 Устройство индикации	13
2.8 Устройство вывода информации.....	13
2.9 Структурная схема устройства.....	13
3 ОБОСНОВАНИЕ ВЫБОРА ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА УПРАВЛЕНИЯ ОБЪЕКТОМ	14
3.1 Состав устройства.....	14
3.2 Микроконтроллер.....	14
3.3 Датчик движения.....	17
3.4 Акустический датчик.....	18
3.5 Датчик освещенности	18
3.6 Органы управления.....	19
3.7 Устройство индикации	19
3.8 Орган отображения информации.....	19
3.9 Подключение исполнительного устройства.....	20
3.10 Расчет системы питания.....	20
3.11 Функциональная схема устройства.....	21

4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА УПРАВЛЕНИЯ ОБЪЕКТОМ	22
4.1 Состав устройства	22
4.2 Микроконтроллер	22
4.3 Датчик движения	24
4.4 Акустический датчик	26
4.5 Датчик освещенности	27
4.6 Органы управления	28
4.7 Устройство индикации	28
4.8 Устройство отображения информации	29
4.9 Подключение исполнительного устройства	30
4.10 Питание устройства	31
4.11 Принципиальная схема устройства	31
5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	32
5.1 Алгоритм работы устройства	32
5.2 Схема программы	33
5.3 Исходный текст программы	35
ЗАКЛЮЧЕНИЕ	37
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	38
ПРИЛОЖЕНИЕ А	
ПРИЛОЖЕНИЕ Б	
ПРИЛОЖЕНИЕ В	
ПРИЛОЖЕНИЕ Г	
ПРИЛОЖЕНИЕ Д	
ПРИЛОЖЕНИЕ Е	

ВВЕДЕНИЕ

Достоинствами автоматического регулирования освещения является удобство в пользовании и уменьшение электропотребления.

Для проектирования устройства управления освещением необходимо контролировать ряд параметров: уровень освещенности, наличие движения и шума в контролируемом помещении.

Проектируемое устройство должно выполнять следующие функции:

- получать данные с окружающей среды, с помощью сенсоров,
- обрабатывать полученные данные, согласно алгоритму,
- получать данные с органа управления,
- обрабатывать полученные данные, согласно алгоритму,
- выводить информацию на устройство отображения информации,
- управлять работой исполнительного устройства.

В соответствии с требованиями проектируемое устройство должно включать:

- микроконтроллер,
- блок датчиков (датчик движения, акустический датчик, датчик освещения),
- орган управления,
- устройство индикации,
- дисплей для вывода информации,
- устройство управления работой исполнительного устройства.

1 ОБЗОР ЛИТЕРАТУРЫ

1.1 Структура разрабатываемого устройства

Разрабатываемое устройство предназначено для управление освещением в помещении. Планируется, что устройство будет контролировать уровень освещенности и наличие движения в контролируемом помещении, а также наличие звука определенного диапазона. Для ручного управления устройством предполагается наличие органа управления, и устройство отображения информации для удобного использования.

В соответствии с требованиями устройство должно включать:

- микроконтроллер,
- блок датчиков,
- орган управления,
- устройство отображения информации,
- блок исполнительных устройств.

1.2 Микроконтроллер

В соответствии с техническим заданием можно использовать 8-разрядный микроконтроллер.

8-разрядные микроконтроллеры предназначены для реализации малопотребляющих устройств с несложным программным кодом. Оптимизированные по экономической эффективности 8-разрядные микроконтроллеры имеют меньший программный код, малое энергопотребление и обеспечивают хорошую устойчивость к шумам.

Такие микроконтроллеры выпускают ряд производителей: Intel, Motorola, Microchip, Atmel, Scenix Semiconductor, Zilog, Holtek, Analog Devices и др.

Ниже представлен обзор трех наиболее популярных 8-разрядных микроконтроллеров.

1. PIC-микроконтроллеры фирмы Microchip.

Фирма Microchip предоставляет линейку 8-разрядных микроконтроллеров с различными особенностями выводов и корпусов. Выпускаемые этой фирмой МК семейств PIC12 и PIC 16/17 (Peripheral Interface Controller) объединяют все передовые технологии микроконтроллеров: электрически программируемые пользователем ППЗУ, минимальное энергопотребление, высокую производительность, хорошо развитую RISC-архитектуру, функциональную законченность и минимальные размеры. Широкая номенклатура изделий обеспечивает использование микроконтроллеров в устройствах, предназначенных для разнообразных сфер применения.

Например, PIC16F630/629 имеют ядро среднего уровня с 8-ми уровневой аппаратной стеком и 35 инструкциями. Эти микроконтроллеры обеспечивают до 5 MIPS, до 1,75 Кбайт памяти программ, до 64 байт ОЗУ и 128 байт EEPROM данных. На борту имеется генератор, откалиброванный на заводе с точностью $\pm 1\%$.

В основу контроллеров положена 8-разрядная RISC-архитектура с системой простых однословных команд. Все команды (кроме команд перехода) выполняются за один машинный цикл (четыре такта). Отличительной особенностью PIC-микроконтроллеров, реализованных на CMOS-технологии, является низкое энергопотребление: 2мА при питании 5 В на тактовой частоте 4 МГц; 15 мкА при питании 3 В на тактовой частоте 32 кГц и менее 3 мкА в режиме ожидания. В связи с этим такие микроконтроллеры широко используются в контактных приборах с питанием от батареек [1].

2. Микроконтроллеры фирмы Atmel.

Фирма Atmel, один из ведущих лидеров по производству однокристалльных микроконтроллеров, представляет линейку 8-разрядных микроконтроллеров серии ATmega с AVR-ядром, который характеризуется минимальным уровнем потребления энергии, возможность запрограммировать до 130 команд высокопроизводительного типа, показатель производительности доходит до 16 MIPS.

AVR-архитектура объединяет 8-разрядный гарвардский RISC-процессор, 32 регистра общего назначения, каждый из которых может работать как аккумулятор и развитую систему команд фиксированной 16-битной длины. Большинство команд выполняются за один машинный такт с одновременным исполнением текущей и выборкой следующей команды.

Характерной особенностью микроконтроллеров фирмы Atmel является встроенная Flash-память с возможностью внутрисхемного программирования через последовательный 4-проводной интерфейс, что обеспечивает быстроту программирования и уменьшение цикла разработки приложений. Их периферия включает: таймеры-счетчики, широтно-импульсные модуляторы, поддержку внешних прерываний, аналоговые компараторы, 10-разрядный 8-канальный АЦП, параллельные контакты (от 3 до 48 линий ввода и вывода), интерфейсы UART и SPI, сторожевой таймер и устройство сброса по включению питания [1].

3. Микроконтроллеры фирмы Analog Devices.

Фирма Analog Device представляет серию 8-разрядных микроконтроллеров AduC со встроенным прецизионным АЦП и ЦАП семейства MicroConverter, применяемые в устройствах обработки данных аналоговых сигналов.

В микроконтроллерах используется ядро 8051/8052, встроенная Flash-память и блок ввода-вывода аналоговой информации, включающий многоканальные ЦАП и АЦП.

Процессоры оснащены либо 12-разрядным АЦП последовательного приближения с многоходовым мультиплексором, либо 16- и 24-разрядными сигма-дельта АЦП. Быстродействие представителей семейства доходит до 16–20 MIPS.

Сравнительная характеристика микроконтроллеров PIC16F716 [1], ADUC812BSZ [2], ATMEGA328P [3] представлена в таблице 1.1.

Таблица 1.1 – Сравнительная характеристика микроконтроллеров

Фирма	Microchip	Analog Devices	Atmel
Наименование	PIC16F716	ADUC812BSZ	ATMEGA328P
Ядро	PIC	8052	AVR
Тактовая частота, МГц	20	16	20
Количество входов/выходов	13	34	27
Объем памяти программ, КБ	3,5	8	32
Интерфейс	spi	i2c, spi, uart	i2c, spi, uart
Встроенная периферия	brown-outdetect/reset, por, pwm, wdt	psm, temp sensor, wdt	brown-outdetect/reset, por, pwm, wdt
Тип памяти	flash	flash	flash
Напряжение, В	2-5,5	2,7-5,5	1,8-5,5

1.3 Датчик движения

Датчик движения – это прибор, который позволяет отслеживать перемещения различных объектов на контролируемой территории.

Основные типы датчиков движения: инфракрасные, ультразвуковые, микроволновые, комбинированные. [5]

Краткое описание основных трех видов датчиков:

1. ИК детекторы движения – устройства, реагирующие на тепловые волны, испускаемые теплыми или холодными движущимися объектами.

Когда объект движется, его ИК излучение поочередно фокусируется различными линзами системы на сенсоре (количество линз обычно варьируется от двадцати до шестидесяти штук), это и является сигналом к выполнению заложенной в датчике функции. Чем больше линз в системе датчика движения – тем выше его чувствительность. Так же, чем больше площадь поверхности системы линз – тем шире зона охвата у датчика движения.

К достоинствам ИК датчиков можно отнести простоту и удобство в эксплуатации, безопасность для человека и животных, точная регулировка параметров обнаружения движущегося объекта, адекватное реагирование на движение.

Недостатками ИК датчиков является возможность ложных срабатываний, чувствительность к условиям окружающей среды, нераспознавание объектов, защищенных материалами, не пропускающими ИК-лучи.

2. СВЧ детекторы (микроволновые датчики) – активные устройства, реагирующие на СВЧ электромагнитные волны, отраженные от объектов.

Используются, когда необходимо контролировать большие площади и работать в широком температурном диапазоне в сильно зашумленных условиях: при ветре, акустических помехах, в тумане, пыли, влажности и т.д. Принцип действия СВЧ детекторов основан на излучении электромагнитных радиочастотных волн в сторону контролируемой зоны.

Работа ультразвукового датчика движения основана на взаимодействии микроволновых волн с материалом и использовании эффекта Доплера – изменение частоты волны, отраженной от движущихся объектов. Само название "микроволновый" говорит о том, что он работает в диапазоне сверхвысоких частот, его длина волны в приблизительном диапазоне от одного миллиметра до одного метра.

Когда в зоне обнаружения микроволнового датчика движения появляется перемещающийся токопроводящий объект, это регистрируется им и сразу поступает сигнал на выполнение встроенной в него функции.

К достоинствам можно причислить способность распознавать движение за несложными препятствиями, чувствительность даже к незначительным колебаниям, независимость от условий среды.

Недостатки: ложное срабатывание, небезопасность для здоровья.

3. Ультразвуковые датчики – устройства, аналогичные СВЧ датчикам, только в них вместо электромагнитных волн используются ультразвуковые колебания. Принцип работы: детекторы посылают акустические волны на объект, принимают и анализируют отраженные от него волны.

Внутри ультразвукового датчика движения расположен генератор звуковых волн (в зависимости от производителя и модели обычно генерируется частота звуковой волны 20–60 кГц), которые излучаются в зоне действия датчика и отражаясь от окружающих объектов поступают обратно в приемник.

Когда в зоне обнаружения ультразвукового датчика движения появляется движущийся объект, частота отраженной от объекта волны изменяется (эффект Доплера), что регистрируется приемником датчика и от него поступает сигнал на выполнение заложенной в ультразвуковой датчик движения функции.

К достоинствам ультразвуковых датчиков относятся нечувствительность к природным явлениям, воздействие на любые объекты независимо от материала.

Недостатки – невосприятие медленных и плавных движений, негативное влияние на животных, способных улавливать ультразвук.

Сравнительные характеристики датчиков PIR-SB312 [6], Troyka-Motion Sensor (на базе RD-624) [7], HC-SR04 [8], HC-SR501 [9] приведена в таблице 1.2.

Таблица 1.2 – сравнительная характеристика датчиков движения

Наименование	Тип	Угол обзора, расстояние, м	Напряжение, В	Рабочий диапазон температур	Выход сигнала
PIR-SB312	ИК	100°, 3-5	2,7-12	-20°-+60°	Цифровой
Troyka-Motion Sensor (на базе RD-624)	ИК	110°, 7	3,3-5	-20°-+60°	Цифровой
HC-SR04	УК	15°, 0,6	5	-20°-+80°	Цифровой
HC-SR501	ИК	120°, 3-7	4,5-20	-20°-+80°	Цифровой

1.4 Акустический датчик

Датчики, работающие в слышимом диапазоне – микрофон. Микрофон характеризуется чувствительностью, направленностью, полосой частот, динамическим диапазоном. [5]

Распознавание звука происходит на основе сравнения с эталоном, который уже записан в память устройства.

Принцип действия: волна проникает внутрь прибора и создает в нем отклонение от стандартного параметра тишины. В качестве контрольных точек выступают скорость звуковой волны и ее амплитуда. Скорость регистрируется благодаря определению частоты и фазности.

Сравнительные характеристики различных акустических датчиков КУ-038 [10], Trema-модуль v2.0 на основе усилителя ОРА134 [11], модуль датчика звука с усилителем МАХ9814[12] приведены в таблице 1.3.

Таблица 1.3 – Сравнительная характеристика акустических датчиков

Наименование	КУ-038	Trema-модуль v2.0 на основе ОРА134	Модуль датчика звука с усилителем МАХ9814
Диапазон частот, кГц	0,1-10	0,035-10	0,02-20
Чувствительность, дБ	20	-40 – +2	40,50,60
Тип сигнала	Аналоговый, цифровой	Аналоговый	Аналоговый
Диапазон раб. t, °С	0 – +70	0 – +70	-40 – +85
Напряжение, В	3,3-5,5	5	2,6-5,5

1.5 Датчик освещенности

Фотоприемник – это электронный прибор, который реагирует на изменение светового потока, падающего на его чувствительную часть.

Существуют несколько разновидностей датчиков освещенности [13]:

1. Фоторезисторы – фотоприбор, изменяющий сопротивление в зависимости от количества света, падающего на его поверхность. Чем интенсивнее освещенность чувствительной области, тем меньше сопротивление.

Фоторезистор не предназначен для точного измерения освещенности, по его показаниям можно определить светлее или темнее стала окружающая среда. Недостаток: граничная частота, описывающая максимальную частоту синусоидального сигнала, которым модулируется световой поток. Фоторезистор непригоден для использования в устройствах, когда нужно высокое быстродействие.

2. Фотодиод – элемент, который преобразует свет, попадающий на чувствительную зону, в электрический заряд.

У фотодиодов также имеется граничная частота, но на порядок больше, чем у фоторезистора, что обеспечивает большее быстродействие.

3. Фототранзистор – биполярный транзистор, у которого в месте вывода базы есть в корпусе окошко для попадания света. Фототранзисторы управляются количеством света, протекающего через базу. Задав необходимый ток базы, можно выставить чувствительность фототранзистора, подбирая его базовый резистор, в связи с чем он может улавливать даже самый тусклый свет.

Сравнительная характеристика датчиков освещенности TSL25911FN [14], GY-302 на основе BH1750 [15], TЕМТ6000 [16] представлена в таблице 1.4.

Таблица 1.4 – Сравнительная характеристика датчиков освещенности

Наименование	TSL25911FN	GY-302	ТЕМТ6000
Тип элемента	Фотодиод	Фотодиод	Фототранзистор
Интерфейс связи	I2C	I2C	–
Эффективный диапазон, Lux	0-88000	0-65535	10-1000
Тип сигнала	Цифровой	Цифровой	Аналоговый
Напряжение, В	3,3-5	3-5	3,3-5

2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА УПРАВЛЕНИЯ ОБЪЕКТОМ

2.1 Проектирование устройства управления объектом

Разрабатываемое устройство предназначено для контроля и управления освещенностью в помещении. Для выполнения своих функций устройство должно контролировать ряд параметров: освещенность, наличие движения, звук в контролируемом помещении. Также должна быть предусмотрена возможность управления устройствам с помощью органа управления.

Устройство должно иметь следующие блоки:

- микроконтроллер,
- блок датчиков (датчик движения, акустический датчик, датчик освещения),
- орган управления,
- устройство индикации,
- устройство вывода информации.

2.2 Микроконтроллер

Микроконтроллер в разрабатываемом устройстве является управляющим элементом, организующим весь процесс функционирования устройства, при этом он должен выполнять ряд задач:

- получать сигналы от блока датчиков и органа управления,
- обрабатывать полученные сигналы,
- посылать информацию на устройство вывода информации,
- посылать сигнал исполнительным устройствам согласно заложенной в нем программе.

2.3 Датчик движения

Датчик движения должен улавливать наличие движения в контролируемом помещении. При обнаружении движения датчик посылает сигнал микроконтроллеру для его дальнейшей обработки.

2.4 Акустический датчик

Акустический датчик должен измерять уровень шума в контролируемом помещении. При обнаружении шума он должен посылать сигнал микроконтроллеру для дальнейшей обработки.

2.5 Датчик освещенности

Датчик освещенности контролирует параметр освещенности в помещении. В зависимости от уровня освещенности он должен посылать сигнал микроконтроллеру для дальнейшей обработки.

2.6 Органы управления

Для управления, изменения программы микроконтроллера в разрабатываемом устройстве должен быть орган управления. Ими могут являться клавиатура или пульт.

2.7 Устройство индикации

Устройство индикации предназначено для проверки работоспособности устройства. Устройство индикации должно определенным способом показывать отсутствие неисправностей или их наличие.

2.8 Устройство вывода информации

Разрабатываемое устройство должно иметь устройство вывода информации для вывода необходимой информации о состоянии системы и для удобства пользования разрабатываемым устройством.

2.9 Структурная схема устройства

В соответствии с вышеприведенным требованиям построена структурная схема, изображенная в Приложении А, включающая в себя следующие блоки:

- микроконтроллер,
- блок датчиков,
- орган управления,
- устройство индикации,
- устройство отображения информации,
- исполнительное устройство.

3 ОБОСНОВАНИЕ ВЫБОРА ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА УПРАВЛЕНИЯ ОБЪЕКТОМ

3.1 Состав устройства

Разрабатываемое устройство, предназначенное для контроля освещенности в помещении, должно контролировать: освещенность, наличие движения, звук в контролируемом помещении; предоставлять возможность ручного управления и отображении информации.

Устройство включает следующие блоки:

- микроконтроллер,
- блок датчиков (датчик движения, акустический датчик, датчик освещения),
- орган управления,
- устройство индикации,
- устройство вывода информации.

3.2 Микроконтроллер

Для работы устройства требуется 8-разрядный микроконтроллер с низким уровнем потребления энергии и высокопроизводительностью. Проанализировав сравнительные характеристики микроконтроллеров компаний Microchip, Analog Devices, Atmel, предоставленных в таблице 1.1, можно сделать вывод, что оптимальным микроконтроллером является ATMEGA328P: тактовая частота 20 МГц, количество входов/выходов – 27, объем памяти – 32 КБ, наличие интерфейсов I2C, SPI, UART, напряжение – 1,8-5,5 В (характеристики PIC16F716 компании Microchip – 20 МГц, 13 вх/вых, 3,5 КБ, SPI, 2-5,5В соответственно и ADUC812BSZ компании Analog Devices – 16 МГц, 34 вх/вых, 8 КБ, I2C, SPI, UART, 2,7-5,5В).

Платы на основе ATMEGA328P выпускает Arduino. Преимущества Arduino заключаются в бесплатной среде разработки, универсальности и модульности, невысокой цене, большое количество информации и инструкций, большое количество периферии, обеспечивающие взаимодействие микроконтроллера с окружающим миром.

Платы Arduino на основе микроконтроллера ATMEGA328: Uno, Nano, Mini, LilyPad. Их основные различия заключаются в габаритах и USB-разъеме.

Краткий обзор плат Arduino на основе микроконтроллера ATMEGA328:

1. Arduino Uno

Arduino Uno является стандартной платой Arduino и возможно наиболее распространенной. Имеет 32 КБ флэш-памяти, 2 Кб SRAM и 1 Кбайт EEPROM памяти. На периферии имеет 14 дискретных (цифровых) каналов ввода /

вывода и 6 аналоговых каналов ввода / вывода, позволяющие перекрывать большинство задач в области микроконтроллерной техники.

2. Arduino Nano

Arduino Nano — это функциональный аналог Arduino Uno, но размещенный на миниатюрной плате. Отличие заключается в отсутствии собственного гнезда для внешнего питания, использованием чипа FTDI FT232RL для USB-Serial преобразования и применением mini-USB кабеля для взаимодействия вместо стандартного. Остальные характеристики совпадают с базовой моделью. Платформа имеет штырьковые контакты, что позволяет легко устанавливать её на макетную плату.

3. Arduino Mini

Та же Arduino Uno, но в другом форм-факторе. Компактная: всего 30×18 мм. Из-за форм-фактора нельзя без ухищрений устанавливать платы расширения Arduino. Предполагается соединение с дополнительными модулями проводами и/или через макетную плату. На плате нет USB-контакта, поэтому прошивать нужно через отдельный USB-Serial адаптер.

4. Arduino LilyPad

Arduino LilyPad имеет не стандартную прямоугольную, а круглую форму. Не поддерживает механические соединения с шилдами. Оно предназначено для небольших автономных устройств. Круглая форма продиктовала то, что разъемы равномерно распределены по окружности, и его небольшой размер (2 дюйма в диаметре) делает его идеальным для переносных устройств. У LilyPad нет регулятора напряжения на борту, так что ему для питания будет необходимо обеспечить по крайней мере 2.7 вольт, и не более 5.5 вольт. Сравнительная характеристика плат представлены в таблице 3.1.

Таблица 3.1 — Сравнительная характеристика плат на основе микроконтроллера ATMEGA328P

Плата	Uno	Nano	Mini	LilyPad
Рабочее напряжение, В	5	5	5	2,7-5,5
Цифровые входы/выходы	14	14	14	20
Выходы с ШИМ	6	6	6	6
Аналоговые входы/выходы	6	8	6	6
Макс. ток с контакта ввода/вывода, мА	40	40	40	40
Flash-память, КБ	32	32	32	32
ОЗУ, КБ	2	2	2	2
EEPROM-память, КБ	1	1	1	1
Тактовая частота, МГц	16	16	16	8
Габариты, мм	69x53	18x45	30x18	Ø50
USB-разъем	USB A-B	Mini-USB	USB-serial	USB-serial

Плата Uno является одной из самых дешевых и наиболее часто используемых, Nano используется там, где важна компактность, а возможностей Arduino Mini недостаточно. В разрабатываемом проекте нет необходимости в компактности устройства, однако наличие разъема USB A-B, возможность напрямую подключать питание упрощает разработку и программирование устройства.

Проанализировав сравнительную таблицу 3.1 и обзор плат для разработки устройства выбрана плата Arduino Uno.

Характеристики Arduino Uno:

- микроконтроллер ATMEGA328,
- рабочее напряжение: 5 В,
- диапазон допустимого напряжения: 6-20 В,
- рекомендуемое входное напряжение питания: 7-12 В,
- количество цифровых вводов/выводов: 14 (6 могут быть использованы как выводы ШИМ),
- количество аналоговых вводов/выводов: 6,
- постоянный ток на выводе питания от 3,3 В: 50 мА,
- постоянный ток на один выход не более: 40 мА,
- Flash-память: 32 КБ, 0,5 КБ используются для загрузчика,
- ОЗУ: 2 КБ,
- энергонезависимая память EEPROM: 1 КБ,
- тактовая частота: 16 МГц.

На рисунке 3.1 изображен внешний вид платы Arduino Uno.

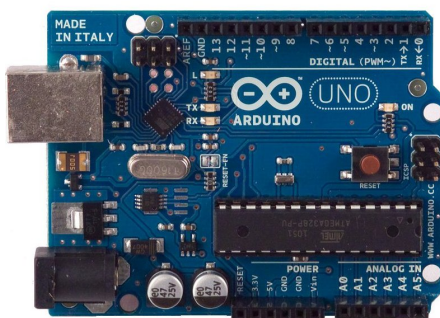


Рисунок 3.1 – Изображение внешнего вида платы Arduino Uno

Рабочее напряжение — 5 В при подключении через USB с любых устройств (компьютер, ноутбук, зарядка от смартфона и т.д.). При одновременном подключении внешнего адаптера (аккумулятора, кроны, блока питания), питание автоматически переключается, но плату можно по-прежнему программировать через компьютер.

3.3 Датчик движения

Датчик движения улавливает наличие движения в контролируемом помещении. Проанализировав обзор датчиков движения и сравнительную таблицу 1.2 оптимальным датчиком движения для платы Arduino Uno является HC-SR501 (рисунок 3.2), угол обзора которого 120 градусов, дальность срабатывания от 3 до 7 метров.



Рисунок 3.2 – Внешний вид датчика HC-SR501

Модуль датчика состоит из PIR-датчика 500BP с дополнительной электрической развязкой на микросхеме BISS0001 и линзы Френеля, которая используется для увеличения радиуса обзора и усиления инфракрасного сигнала.

Управление работой датчика осуществляет микросхема BISS0001. На плате имеются два потенциометра, с помощью которых настраивается дистанция обнаружения объектов и задержка после первого срабатывания датчика (5–300 сек). Модуль имеет два режима – L и H. Режим работы устанавливается с помощью переключки. Режим L – режим единичного срабатывания, при обнаружении движущегося объекта на выходе OUT устанавливается высокий уровень сигнала на время задержки и датчик не реагирует. В режиме H датчик срабатывает каждый раз при обнаружении движения.

Основные характеристики датчика HC-SR501:

- напряжение питания: 4,5–20 В,
- потребляемый ток: 20 мА,
- напряжение на выходе: TTL уровни 3,3 В,
- расстояние обнаружения: 3–7 м,
- длительность задержки после срабатывания: 5–300 сек,
- угол наблюдения: 120°,
- время блокировки до следующего замера: 2,5 сек,
- рабочая температура: -20°+80°,
- тип сигнала: цифровой.

3.4 Акустический датчик

Основной функцией акустического датчика является срабатывание на хлопок в ладоши. По сравнительной таблице 1.3 подходящим для платы Arduino Uno является модуль датчика звука KY-038 (рисунок 3.3).

Этот модуль находят свое применение в различных системах управления. Например, такие модули часто устанавливают в подъездах для включения освещения при звуке шагов, что позволяет значительно сэкономить электроэнергию. Также с помощью этих модулей можно управлять роботом посредством хлопков или других звуков.

Модуль состоит из микрофона и потенциометра, позволяющего регулировать чувствительность датчика. Также в модуле имеется компаратор – устройство, посылающее цифровой сигнал при достижении аналоговым сигналом на входе определенных значений. Данный модуль является недорогим, легкодоступным и несложным в программировании.



Рисунок 3.3 – Внешний вид модуля датчика звука KY-038

Основные характеристики:

- напряжение питания: 3,7-5,5 В,
- ток потребления: 10 мА.

3.5 Датчик освещенности

В качестве датчика освещенности для Arduino Uno используется датчик GY-302 на основе чипа BH1750 (рисунок 3.4), так как он легкодоступен и прост в использовании.

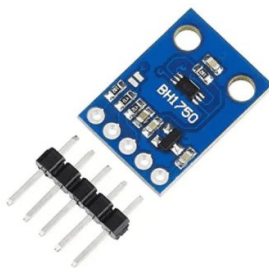


Рисунок 3.4 – Внешний вид датчика GY-302

Основные характеристики:

- диапазон измерения освещенности: 1-65535 люкс,
- подавление светового шума 50-60 Гц,
- интерфейс I2C,
- напряжение питания: 3-5 В,
- ток потребления: 120 мкА.

3.6 Органы управления

В разрабатываемом проекте в качестве органа управления выбрана клавиатура. При управлении освещением нет необходимости в большом количестве цифр (для ввода параметров, паролей и так далее), в связи с этим выбрана матричная клавиатура 1x3 (рисунок 3.5). Преимущества матричной клавиатуры – подходит для Arduino Uno, низкая стоимость, легкодоступность.



Рисунок 3.5 – Внешний вид матричной клавиатуры 1x3

Основные характеристики:

- максимальное напряжение: 12 В,
- максимальный ток: 100 мА.

3.7 Устройство индикации

В качестве устройства индикации будет использован светодиод.

Основные характеристики:

- напряжение питания: 5 В,
- номинальный ток потребления: 20 мА,

3.8 Орган отображения информации

В качестве органа управления для Arduino Uno выбран ЖК дисплей LCD 1602[18]. Внешний вид дисплея изображен на рисунке 3.6.

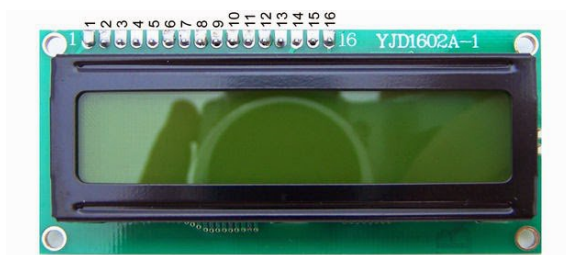


Рисунок 3.6 – Внешний вид дисплея LCD 1602

Основные характеристики:

- рабочее напряжение: 5 В,
- максимальный потребляемый ток: 150 мА,
- цвет подсветки: синий,
- цвет символов: белый,
- контроллер: HD44780,
- количество строк: 2,
- количество символов в строке: 16.

3.9 Подключение исполнительного устройства

К плате Arduino Uno нельзя напрямую подключать устройства, работающие в режимах с относительно большими токами и напряжениями. Для подключения таких устройств используется реле. Наиболее распространенное реле для платы Arduino выполняется в виде модуля SONGLE SRD-05VDC. Устройство управляется напряжением 5 В, может коммутировать до 10 А 30 В DC и 10 А 250 В AC.

3.10 Расчет системы питания

Разрабатываемое устройство в своем составе имеет ряд модулей. Для расчета суммарной мощности устройства необходимо суммировать мощность, потребляемый ток каждого модуля и рассчитать напряжение питания.

По характеристикам модулей посчитаем мощность каждого модуля по формуле:

$$P = U \cdot I, \quad (3.1)$$

где U – напряжение питания, I – потребляемый ток.

Напряжение питания, максимальный потребляемый ток и вычисленная потребляемая мощность каждого модуля представлены в таблице 3.2

Таблица 3.2 – Характеристика модулей устройства

Наименование модуля	Напряжение питания, В	Максимальный потребляемый ток, А	Потребляемая мощность, Вт
Плата Arduino Uno	5	0,02	0,1
Датчик HC-SR501	5	0,065	0,1
Датчик KY-038	5	0,01	0,05
Датчик GY-302	5	0,00012	0,1
Матричная клавиатура	5	0,1	0,5
Дисплей LCD 1602	5	0,04	0,2
Светодиод	2,2	0,02	0,1
Модуль SONGLE SRD-05VDC	5	0,08	0,4

Суммарная мощность устройства составляет 1,6 Вт.

С целью обеспечения надежности работы устройства добавим 20 процентов к полученной мощности, в результате мощность устройства составляет 1,9 Вт. Минимальный выходной ток источника питания составляет 0,35 А.

Вычислим напряжение питания устройства по формуле:

$$U=P/I, \quad (3.2)$$

Напряжение устройства: $U = 1,9/0,35 = 5,4$ В.

Таким образом разрабатываемое устройство имеет характеристики:

- потребляемый ток – 0,35А,
- напряжение – 5,4 В.

3.11 Функциональная схема устройства

В соответствии с вышеприведенным требованиям построена функциональная схема, изображенная в Приложении Б, включающая в себя следующие блоки:

- плата Arduino Uno на основе микроконтроллера ATMEGA328P,
- датчик движения HC-SR501,
- датчик звука KY-038,
- датчик освещенности GY-302,
- матричная клавиатура 1х3,
- дисплей LCD 1602,
- блок питания,
- модуль реле,
- исполнительное устройство.

С датчиков сигнал поступает на микроконтроллер. От микроконтроллера сигналы идут на модуль реле и дисплей. Датчики GY-302 и HC-SR501 имеют цифровой сигнал выхода, модуль датчика звука KY-038 имеет аналоговый сигнал выхода.

Описание контактов платы Arduino Uno, изображенных на рисунке 4.1 представлены в таблице 4.1.

Таблица 4.1 – Обозначение контактов платы Arduino Uno

Контакт	Назначение
Цифровой 0	Порт UART (RX)
Цифровой 1	Порт UART (TX)
Цифровой 2	Вход для прерываний
Цифровой 3	Вход для прерываний, ШИМ
Цифровой 4	
Цифровой 5	ШИМ
Цифровой 6	ШИМ
Цифровой 7	
Цифровой 8	
Цифровой 9	ШИМ
Цифровой 10	SPI (SS), ШИМ
Цифровой 11	SPI (MOSI), ШИМ
Цифровой 12	SPI (MISO)
Цифровой 13	SPI (SCK). К выходу подсоединен встроенный светодиод
Аналоговый A0	
Аналоговый A1	
Аналоговый A2	
Аналоговый A3	
Аналоговый A4	I2C (SDA)
Аналоговый A5	I2C (SCL)
VIN	Питание от внешнего источника питания на 7-12 В
USB	Подключение USB-кабеля
5V	Источник напряжения для питания внешних устройств
3.3V	Источник напряжение 3,3 В для питания внешних устройств, формируемое от внутреннего стабилизатора платы (используется для подключения внешних источников, требующих именно 3,3 В)
GND	Земля
AREF	Опорное напряжение для аналоговых входов
IOREF	Рабочее напряжение
Reset	Сброс микроконтроллера

Подача внешнего напряжения 5 В на контакт 5V недопустимо, так как питание Arduino Uno обходит стороной стабилизатор, что может привести к поломке. Все цифровые контакты на плате выдают стабилизированное напряжение в 5 В. Аналоговые контакты считывают сигналы с аналоговых датчиков и преобразуют его в цифровой сигнал. Цифровые контакты используются для цифрового ввода и вывода.

Использование контактов платы Arduino Uno и их наименование на принципиальной схеме представлены в таблице 4.2.

Таблица 4.2 – Обозначение используемых контактов платы Arduino Uno на принципиальной схеме

Контакт	Модуль, использующий контакт	Номер выхода контакта	Контакт соединения
Цифровой 2	LCD 1602	19	12
Цифровой 3	LCD 1602	18	11
Цифровой 4	LCD 1602	17	10
Цифровой 5	LCD 1602	16	9
Цифровой 6	Матричная клавиатура	15	8
Цифровой 7	Матричная клавиатура	14	7
Цифровой 8	Матричная клавиатура	13	6
Цифровой 9	Матричная клавиатура	12	5
Цифровой 10	Реле SRD-05VDC	11	4
Цифровой 11	LCD 1602	10	3
Цифровой 12	LCD 1602	9	2
Цифровой 13	Светодиод	8	1
Аналоговый А0	Модуля датчика звука с усилителем MAX9814	1	1
Аналоговый А1	Датчик движения HC-SR501	2	1
Аналоговый А4	Модуль GY302	3	1
Аналоговый А5	Модуль GY302	4	2
5V		6	
GND		7	
VIN		5	

4.3 Датчик движения

Датчик движения HC-SR501 имеет регулировку чувствительности, которая позволяет определять движение от 3 до 7 метров, а его выход можно настроить так, чтобы он оставался высоким в течение времени от 3 секунд до 5 минут. Так же, датчик имеет встроенный стабилизатор напряжения, поэтому он может питаться от постоянного напряжения от 4,5 до 20 В и потребляет небольшое количество тока. HC-SR501 имеет 3-контактный контакт. Изображение контактов представлено на рисунке 4.2 [18].



Рисунок 4.2 – Изображение и назначение контактов датчика HC-SR501

Назначение выводов:

- VCC – положительное напряжение постоянного тока,
- OUT – логический выход на 3,3 вольта, высокий уровень сигнала указывает на обнаружение движения,
- GND – заземление.

На плате установлены два потенциометра для настройки параметров:

- чувствительность (дистанция) – устанавливает максимальное и минимальное расстояние (от 3 метров до 7 метров),
- время (задержка) – время, в течение которого выход будет оставаться HIGH после обнаружения. Как минимум, 3 секунды, максимум 300 секунд или 5 минут.

Назначение переключателя режима срабатывания:

- H – это настройка Hold или Repeat. В этом положении HC-SR501 будет продолжать выдавать высокий уровень сигнала, пока он продолжает обнаруживать движение.
- L – это параметр прерывания или без повтора. В этом положении выход будет оставаться в высоком уровне сигнала в течение периода, установленного настройкой потенциометра задержки

Назначение дополнительных отверстий [20]:

- RT — предназначено для термистора или чувствительного к температуре резистора. Добавление этого позволяет использовать HC-SR501 в экстремальных температурах, а также в некоторой степени повышает точность работы детектора.
- RL — это соединение для светозависимого резистора или фоторезистора. Добавляя компонент, HC-SR501 будет работать только в темноте, что является общим приложением для чувствительных к движению систем освещения.

Использование контактов датчика движения HC-SR501 и их наименование на принципиальной схеме представлены в таблице 4.3.

Таблица 4.3 – Обозначение используемых контактов датчика HC-SR501 на принципиальной схеме

Контакт	Контакт соединения	Номер выхода контакта
VCC	6	1
OUT	2	2
GND	7	3

4.4 Акустический датчик

Преимущество модуля датчика звука KY-038 (рисунок 4.3) заключается в возможности программирования его как на аналоговый, так и на цифровой контакты устройства.



Рисунок 4.3 – Изображение модуля датчика звука KY-038 с контактами

Назначение контактов усилителя:

- G – заземление,
- “+” – напряжение питания,
- A0 – аналоговый выход, передающий выходное напряжения на микрофоне,
- D0 – цифровой выход, посылающий логическую единицу при достижении порогового уровня громкости.

Стандартное подключение модуля к контроллеру выполняется по трем контактам: Vdd (“+”), GND (“-”), A0. При необходимости получения цифрового сигнала D0 подключается к цифровому контакту.

Использование контактов и их наименование на принципиальной схеме представлены в таблице 4.4.

Таблица 4.4 – Обозначение используемых контактов модуля датчика звука KY- 38 на принципиальной схеме

Контакт	Контакт соединения	Номер выхода контакта
A0	1	1
VCC	6	2
G	7	3

4.5 Датчик освещенности

Модуль GY-302 на базе чипа BH1750 (рисунок 4.4), представляет собой высокоточный цифровой датчик интенсивности света, выдающий значение в люксах.



Рисунок 4.4 – Выводы модуля датчика GY-302

Назначение контактов:

- VCC – питание 5 В,
- GND – заземление,
- ADDR – выбор адреса модуля на шине I2C, выход притянут к массе и адрес модуля 0x23, если подать логическую единицу, адрес будет 0X5C,
- SCL – синхронизация I2C
- SDA – данные I2C.

ADDR можно не подключать, при подключении, в зависимости от уровня сигнала модель может иметь два адреса, что позволяет подсоединить одновременно два датчика BH1750.

Использование контактов и их наименование на принципиальной схеме представлены в таблице 4.5.

Таблица 4.5 – Обозначение используемых контактов датчика GY-302 на принципиальной схеме

Контакт	Контакт соединения	Номер выхода контакта
SDA	3	1
SCL	4	2
VCC	6	3
GND	7	4

4.6 Органы управления

В качестве органа управления выбрана клавиатура 1х3. Каждая кнопка, при нажатии на нее, замыкает контакты между столбцом и конкретным рядом, создавая цепь, которую можно программно обнаружить (рисунок 4.5).

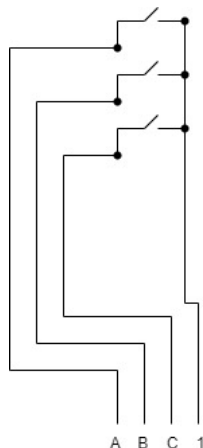


Рисунок 4.5 – Схема матричной клавиатуры

Контакты А-С отвечают за ряды клавиатуры, контакт 1 – столбец.

Использование контактов и их наименование на принципиальной схеме представлены в таблице 4.6.

Таблица 4.6 – Обозначение используемых контактов клавиатуры на принципиальной схеме

Контакт	Контакт соединения	Номер выхода контакта
D1	5	1
D2	6	2
D3	7	3
D4	8	4

4.7 Устройство индикации

Цифровые контакты Arduino способны выдавать ток до 40 мА, светодиод имеет значение предельно допустимого тока в 20 мА. Следовательно, подключив светодиод напрямую к цифровому контакту платы, он быстро выйдет из строя. Светодиод характеризуется падением напряжения при прохождении тока в прямом направлении. Падение напряжения существенно выше чем у обычных диодов и составляет 2-3 вольта. Это означает, что светодиод не будет светиться если поданное на него напряжение меньше этого значения. Но если на светодиод подать напряжение больше этого порога, то через него потечет максимально возможный ток, и

светодиод выйдет из строя. Что бы этого не произошло светодиод необходимо подключать через токоограничивающий резистор.

Расчет сопротивления токоограничивающего резистора производится по следующей формуле:

$$R_{CB} = (V_{УПР} - V_{CB}) / I_{CB}, \quad (4.1)$$

где $V_{УПР}$ – напряжение управления (напряжение вывода микроконтроллера – 5В), V_{CB} – падение напряжения на светодиоде, I_{CB} – сила тока светодиода (10-20 мА).

Для обычных светодиодов падение напряжения составляет 2 В и рекомендуемый ток свечения 15 мА. При подключении к выводу, запитанному от 5 В:

$$R_{CB} = (5-1)/0,015 = 200 \text{ Ом.}$$

С учетом допуска можно использовать резистор номиналом 220 Ом.

Использование контактов и их наименование на принципиальной схеме представлены в таблице 4.7.

Таблица 4.7 – Обозначение используемых контактов светодиода на принципиальной схеме

Контакт	Контакт соединения	Номер выхода контакта
Анод	1	1
Катод	7	2

Резистор 220 Ом подключается к катоду и GND.

4.8 Устройство отображения информации

ЖК дисплей LCD 1602 имеет 16 контактов подключения (рисунок 4.6).

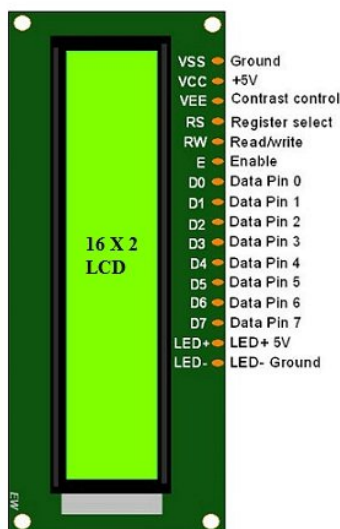


Рисунок 4.6 – Внешний вид дисплея LCD 1602

В таблице 4.8 приведено назначение контактов дисплея LCD 1602.

Таблица 4.8 – Назначение контактов дисплея LCD 1602

Контакт	Назначение
VSS	GND (земля)
VCC	питание 5 В
VEE	установка контрастности дисплея
RS	Командный режим и режим данных
RW	Записывание и чтение данных
E	Контакт включения/управления
D0-D7	Линии данных
LED+	Контакт питания подсветки дисплея
LED-	Вывод катода светодиода подсветки дисплея

К плате Arduino дисплей подключается через переменный резистор, который регулирует контрастность дисплея.

Использование контактов и их наименование на принципиальной схеме представлены в таблице 4.9.

Таблица 4.9 – Обозначение используемых контактов LCD 1602 на принципиальной схеме

Контакт	Контакт соединения	Номер выхода контакта
D4	9	1
D5	10	2
D6	11	3
D7	12	4
RS	2	5
E	3	6
VEE	13	7
VCC	6	8
LED+	6	9
VSS	7	10
RW	7	11
LED-	7	12

4.9 Подключение исполнительного устройства

Исполнительное устройство подключается с помощью реле SRD-05VDC. Контакты реле рассчитаны на силу тока до 10 А при напряжении до 250 В переменного напряжения и до 30 В постоянного напряжения. Для быстрого подключения модуля реле, предусмотрены две группы контактов, первая группа «слаботочная» шаг 2,54 мм, вторая группа силовая.

Назначение контактов видно из рисунка 4.7 [21].



Рисунок 4.7 – Выводы реле SRD-05VDC

Использование контактов и их наименование на принципиальной схеме представлены в таблице 4.10.

Таблица 4.10 – Обозначение используемых контактов реле SRD-05VDC на принципиальной схеме

Контакт	Контакт соединения	Номер выхода контакта
5V	6	1
S	4	2
GND	7	3

4.10 Питание устройства

Согласно таблице 3.2 для разрабатываемого устройства необходим блок питания не менее 5,4 В и 0,35 А. Так как плата Arduino имеет рекомендуемое входное напряжение 7-12 В, то для питания устройства от сети 220В необходим источник питания от сети переменного тока, преобразующий напряжение в 9В. Таким устройством является источник питания от сети переменного тока 220В модели MB101, которое имеет входное напряжение 110-220В, выходное напряжение 9В, выходной ток - 1 А.

Подключается к VIN платы Arduino Uno, номер линии на принципиальной схеме 50.

4.11 Принципиальная схема устройства

В соответствии с вышеприведенным требованиям построена принципиальная схема, изображенная в Приложении В, включающая в себя следующие блоки:

- плата Arduino Uno,
- датчик движения HC-SR501,
- модуль датчика звука KY-038,
- датчик освещенности KY-016,
- клавиатура,
- дисплей LCD 1602,
- блок питания,
- реле.

5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1 Алгоритм работы устройства

Разрабатываемое устройство управляет автоматическим освещением с помощью датчиков движения, освещенности и акустического датчика. В программе присутствует функция ручного ввода времени, которое запускает таймер обратного отсчета, в течение которого будет работать освещение. Дисплей отображает время таймера обратного отсчета.

Программа управляющая устройством выполняет следующие действия:

1. Инициализация

- настройка аналоговых и цифровых контактов для датчиков, дисплея и клавиатуры согласно принципиальной схеме устройства,
- объявление и инициализация переменных для работы устройства с дисплеем.
- объявление и инициализация переменных для работы с клавиатурой.

2. Объявление отдельной функции установки таймера включения света.

По умолчанию переменные секунды и минуты установлены в ноль. Считываются данные о нажатии кнопки на клавиатуре:

- первая кнопка, увеличивает отсчет таймера на одну минуту
- вторая кнопка уменьшает отсчет таймера на минуту.
- третья кнопка запускает обратный отсчет таймера, в течение которого на контакт реле подается высокий сигнал.

Если третья кнопка нажата без установки таймера, то таймер начнет отсчет с одной минуты.

3. Чтение данных из цифровых контактов датчика движения и освещенности. Если освещенность меньше допустимого значения (недостаточно освещенности) и уровень сигнала датчика движения высокий (обнаружено движение), то засекается время, в течение которого подается высокий уровень сигнала на контакт, к которому подключено реле, управляющее исполнительным устройством.

4. Вызывается функция установки таймера. Если нажата кнопка, то выполняется установленный алгоритм функции.

5. Чтение данных из аналогового контакта акустического датчика. При обнаружении сигнала, выше установленного (обнаружен громкий звук), программа инвертирует состояние логической переменной освещения и засекается время, в течение которого подается сигнал на контакт реле согласно состоянию переменной.

6. Вызывается функция установки таймера. Если нажата кнопка, то выполняется установленный алгоритм функции.

5.2 Схема программы

В Приложении Г приведен алгоритм основной программы устройства в виде блок-схемы.

Назначение блоков алгоритма основной программы:

01 – начало программы;

02 – объявление и инициализация переменных: объявление объекта для управления дисплеем, объявление и инициализация массива для работы с дисплеем, объявление и инициализация массивов для работы клавиатуры, объявление и инициализация переменных секунды (SEC), минуты (MIN), времени (timer), контроля освещенности (LIGHT), состояния освещенности (ledState);

03 – вход в функцию loop (), которая бесконечно повторяет заложенный в ней код программы;

04 – чтение уровня сигналов с контакта датчика движения HC-SR501 и датчика освещенности GY302;

05 – проверка условия на уровень освещенности и наличия движения, если уровень освещения меньше установленного и присутствует движение, то переход к блоку 06, иначе переход к блоку 13;

06 – установление логической переменной состояния освещенности в 1;

07 – установление текущего времени программы в переменную времени;

08 – вход в цикл while, по условию которого в течение 10 секунд выполняются условия блока 09;

09 – подача высокого уровня сигнала на контакт реле, вызов функции установки таймера;

10 – выход из цикла while() по окончании установленного времени;

11 – инвертирование переменной состояния освещенности;

12 – подача низкого уровня сигнала на контакт реле;

13 – чтение уровня сигнала с контакта датчика звука KY-038;

14 – проверка условия на наличие звука выше установлено: если уровень звука превышает допустимый, переход к блоку 15, иначе переход к блоку 22;

15 – инвертирование переменной состояния освещенности;

16 – установление текущего времени программы в переменную времени;

17 – вход в цикл while, по условию которого в течение 10 секунд выполняются условия блока 18;

18 – подача высокого уровня сигнала на контакт реле, вызов функции установки таймера;

19 – выход из цикла while() по окончании установленного времени;

20 – инвертирование переменной состояния освещенности;

21 – подача низкого уровня сигнала на контакт реле;

22 – вызов функции установки таймер;

23 – конец функции loop();

24 – выход из программы.

В Приложении Д приведен алгоритм подпрограммы установки таймера в виде блок-схемы.

Назначение блоков алгоритма подпрограммы:

25 – начало подпрограммы;

26 – вызов функции чтения нажатой кнопки на клавиатуре;

27 – проверка условия: если нажата кнопка «3», то переход к блоку 28, иначе переход к блоку 30;

28 – увеличение переменной минуты на 1, установление переменной секунды в 0;

29 – вызов функций настройки отображения информации на дисплее;

30 – проверка условия: если нажата кнопка «2», то переход к блоку 31, иначе переход к блоку 33;

31 – уменьшение переменной минуты на 1, установление переменной секунды в 0;

32 – вызов функций настройки отображения информации на дисплее;

33 – проверка условия: если нажата кнопка «1», то переход к блоку 34, иначе переход к блоку 26;

34 – установление курсора на дисплее для отображения информации;

35 – проверка условия: если секунды равны нулю, то переход к блоку 36, иначе переход к блоку 37;

36 – установление переменной секунды в значение 60, уменьшение переменной минуты на 1, установление логической переменной состояния освещенности в 1;

37 – проверка условия: если минуты меньше 0, то переход к блоку 38, иначе переход к блоку 39;

38 – установление переменной минуты в значение нуль;

39 – установление логической переменной состояния освещенности в 1;

40 – подача высокого уровня сигнала на контакт реле, установление логической переменной контроля освещенности в 1;

41 – вход в цикл while, по условию которого, пока переменная контроля освещенности находится в состоянии логической 1 выполняются условия блоков 42-53;

42 – проверка условия: если прошло 995 миллисекунд с момента присвоения переменному времени значения времени работы программы, то переход к блоку 43, иначе переход к блоку 54;

43 – установление текущего времени программы в переменную времени, уменьшение переменной секунды на 1;

44 – проверка условия: если секунды равны нулю и минуты равны нулю, то переход к блоку 45, иначе переход к блоку 48;

45 – вызов функций настройки отображения информации на дисплее;

46 – установление логических переменных состояния и контроля освещенности в 0;

47 – подача низкого уровня сигнала на контакт реле, переход к блоку 54;

48 – проверка условия: если секунды равны нулю, то переход к блоку 49, иначе переход к блоку 52;
49 – установление переменной секунды в значение 59, уменьшение переменной минуты на 1;
50 – проверка условия: если минуты меньше нуля, то переход к блоку 51, иначе переход к блоку 52;
51 – установление переменной минуты в значение нуль;
52 – проверка условия: если переменная контроля освещенности равна единице, то переход к блоку 53, иначе переход к блоку 54;
53 – вызов функций настройки отображения информации на дисплее;
54 – выход из цикла while() если переменная контроля освещенности в 1;
55 – выход из подпрограммы.

5.3 Исходный текст программы

Программа управляющая устройством написана на языке программирования C++ с помощью компилятора Arduino IDE. Исходный текст программы с комментариями приведен в Приложении Е.

Описание исходного текста программы:

Строки 001-007 – Подключение внешних библиотек для работы с готовыми функциями.

Строки 010-013 – Определение директив для использования переменных в программе.

Строки 015-038 – Инициализация глобальных переменных, представлены в блок-схеме 02.

Строка 040 – Объявление прототипа функции установки таймера обратного отсчета.

Строка 042 – Вход в функцию setup().

Строки 042-059- Функции setup(), представляющей собой объявление и инициализацию первоначальных значений переменных, представлены в блок-схеме 02.

Строки 061-066 – Цикл for() для индикации начала устройства.

Строка 070 – Вход в функцию loop (), представлен на блок-схеме 03..

Строки 071-073 – Чтение данных с контактов датчиков движения и освещенности.

Строки 075-088 – Проверка условия. При условии, что в контролируемом помещении недостаточно освещенности и при этом обнаружено движение, то переменная состояния освещенности устанавливается в 1 и вызывается функция для подачи высокого уровня сигнала на контакт реле с помощью цикла while() в течение 10 сек. В течение данного времени вызывается функция установки таймера обратного отсчета

Данный цикл `for()`, процессы и вложенный в него цикл `while()` представлены на блок-схемах 05-12.

Строки 092-101 – Чтение данных с контакта датчика звука. Проверка условия. Если обнаружен звук выше установленного, то происходит инвертирование переменной состояния освещенности и подача сигнала на контакт реле с помощью цикла `while()` в течение 10 сек, согласно значению переменной состояния освещенности. Данный цикл представлен на блок-схемах 13-21.

Строка 102 – Вызов функции для настройки таймера обратного отсчета. Представлена на блок-схеме 22.

Строка 103 – Конец функции `loop()`/

Строка 109 – Вход в функцию `setTimer()`. Представлена на блок-схеме 25.

Строка 111 – Чтение данных о нажатой кнопке (блок-схема 26).

Строки 113-122 – Цикл условия. Если нажата кнопка «3», то к минутам прибавляется единица, секунды устанавливаются в значение нуль, вывод информации на дисплей. Данный цикл представлен на блок-схемах 27-29.

Строки 124-136 – Цикл условия. Если нажата кнопка «2», то у минут отнимается единица, секунды устанавливаются в значение нуль, вывод информации на дисплей. Данный цикл представлен на блок-схемах 30-32.

Строки 139-153 – Цикл условия. Если нажата кнопка «3», то начинается обратный отсчет. В данный цикл включены циклы проверки минут и секунд, если секунды равны нулю, то значению переменной секунд присваивается 60, у минут отнимается 1. Если минуты меньше нуля, то присваивается значение нуль. Переменная состояния освещения и контроля освещенности устанавливаются в логическую единицу. На контакт реле подается сигнал высокого уровня. Данный цикл представлен на блок-схемах 33-40.

Строка 155 – Вход в цикл `while()`. В течение времени, пока переменная контроля освещенности имеет значение 1 выполняется цикл. Представлено на блок-схеме 41.

Строки 156-158 – Вход в цикл условия. Если прошло времени больше, чем 995 миллисекунд, то от секунды отнимается 1. Представлено на блок-схеме 42-43.

Строки 162-173 – Цикл условия. Если секунды и минуты равны нулю, информация выводится на дисплей, переменным состоянию освещенности и контроля освещенности присваиваются нулю. На контакт реле подается сигнал низкого уровня. Данный цикл представлен на блок-схемах 44-47.

Строки 176-179 – Цикл условия. Если секунды равны нулю, то секундам присваивается значение 59, от минут отнимается 1, если минуты меньше нуля, то минутам присваивается значение 0. Представлено на блок-схеме 48-51.

Строки 183-191 – Цикл условия. Если переменная контроля освещенности находится в состоянии 1, то информацию отображается на дисплей.

Строка 194 – Конец функции `setTimer()`.

ЗАКЛЮЧЕНИЕ

В данном курсовом проекте было спроектировано и разработано устройство управления освещением.

Устройство выполняет следующие функции:

- контроль за уровнем освещения в контролируемом помещении,
- контроль за наличием движения в контролируемом помещении,
- контроль за наличием звука в контролируемом помещении,
- установление таймера обратного отсчета для установки времени освещения.

При работе устройство считывает данные об уровне освещенности и наличия движения, если оба условия соблюдены передается команда на включение освещения. При достаточном освещении по желанию пользователя можно включить освещение на короткое время с помощью датчика звука, а также можно установить обратного отсчета на любое удобное время кратное минуте.

Устройство работает от сети 220 В, имеет в своем составе блок реле для подключения аппаратуры, требующей большую мощность, чем 5В.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Пашкевич А.П. Микропроцессорные системы управления: конспект лекций / А. П. Пашкевич, О. А. Чумаков. – Мн.: БГУИР, 2006. – 72 с.
- [2] PIC16F716 Datasheet [электронный ресурс] / Alldatasheet. – Режим доступа: <https://html.alldatasheet.com/html-pdf/93054/MICROCHIP/PIC16F716/408/1/PIC16F716.html>.
- [3] ADUC812BSZ Datasheet [Электронный ресурс] / Alldatasheet. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/539298/AD/ADUC812BSZ.html>.
- [4] ATMEGA328 Datasheet [Электронный ресурс] / Alldatasheet. – Режим доступа: <https://html.alldatasheet.com/html-pdf/392243/ATMEL/ATMEGA328/152/1/ATMEGA328.html>.
- [5] Фрайден ДЖ. Мир электроники. Современные датчики: справочник / Дж. Фрайден. – М.: Техносфера, 2005. – 592 с.
- [6] PIR-SB312 [Электронный ресурс] : Datasheet / Chipdip. – Режим доступа: <https://static.chipdip.ru/lib/859/DOC003859656.pdf>
- [7] Troyka-Motion Sensor (на базе RD-624) [Электронный ресурс]: Datasheet. – Режим доступа: <https://datasheetspdf.com/pdf-file/1415595/Winsen/RD-624/1>.
- [8] HC-SR04 Datasheet [Электронный ресурс] / Alldatasheet. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/1132203/ETC2/HC-SR04.html>.
- [9] HC-SR501 Datasheet [Электронный ресурс] / Alldatasheet. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/1131987/ETC2/HC-SR501.html>.
- [10] KY-038 Datasheet [Электронный ресурс] / Alldatasheet. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/1138845/ETC2/KY-038.html>.
- [11] OPA134 [Электронный ресурс]: Datasheet / Chipdip. – Режим доступа: <https://static.chipdip.ru/lib/223/DOC000223960.pdf>.
- [12] MAX9814: Datasheet [Электронный ресурс] / Alldatasheet. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/217128/MAXIM/MAX9814.html>.
- [13] Виглеб Г. Датчики. Устройство и применение / Г. Виглеб. – М.: Мир, 1989. – 196 с.
- [14] TSL25911: Datasheet [Электронный ресурс] / Alldatasheet. – Режим доступа: <https://html.alldatasheet.com/html-pdf/621672/AMSCO/TSL25911/151/1/TSL25911.html>.
- [15] BH1750FVI: Datasheet [Электронный ресурс] / Rohm semiconductor. – Режим доступа: <https://www.elechouse.com/elechouse/images/product/Digital%20light%20Sensor/bh1750fvi-e.pdf>.

[16] ТЕМТ6000: Datasheet [Электронный ресурс] / Alldatasheet. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/117488/VISHAY/ТЕМТ6000.html>.

[17] Изображение платы Arduino Uno [Электронный ресурс] / 100mom. – Режим доступа: <https://100mom.ru/arduino/uroki/arduino-uno-raspinovka>.

[18] LCD Module 1602A-1 [Электронный ресурс]: specification / Chipdip. – Режим доступа: <https://static.chipdip.ru/lib/991/DOC020991196.pdf>

[19] Изображение и назначение разъемов датчика HC-SR501 [электронный ресурс] / Arduino-kit – Режим доступа: <https://arduino-kit.ru/blogs/blog/48-podklyuchenie-datchika-prisutstviya-hc-sr501>.

[20] Описание назначения разъемов датчика HC-SR501 [Электронный ресурс] / Robotchip. – Режим доступа: <https://robotchip.ru/obzor-infrakrasnogo-datchika-dvizheniya-hc-sr501/>.

[21] Описание назначения разъемов реле SONGLE SRD-05VDC [Электронный ресурс] / arduinomaster. – Режим доступа: <https://arduinomaster.ru/datchiki-arduino/podklyuchenie-rele-k-arduino/>.

ПРИЛОЖЕНИЕ А
(обязательное)
Структурная схема устройства управления объектом

ПРИЛОЖЕНИЕ Б
(обязательное)
Функциональная схема устройства управления объектом

ПРИЛОЖЕНИЕ В
(обязательное)
Принципиальная схема устройства управления объектом

ПРИЛОЖЕНИЕ Г
(обязательное)
Схема алгоритма основной программы

ПРИЛОЖЕНИЕ Д
(обязательное)
Схема алгоритма подпрограммы

ПРИЛОЖЕНИЕ Е

(обязательное)

Исходный текст программы

```
000 //Гр. 050541 Лейкина Александра Эдуардовна, тема «Устройство управления объектом»
001 #include <Keypad.h>
002 // Подключение библиотеки для датчика освещенности
003 #include <BH1750.h>
004 // Подключение библиотеки для связи МК с модулями через I2C
005 #include <Wire.h>
006 // Подключение библиотеку для работы с экраном
007 #include <LiquidCrystal.h>
008 // Инициализация переменных для контактов светодиода, акустического датчика
009 // датчика движения, реле
010 #define LED 13
011 #define SOUND A0
012 #define PIR A1
013 #define REL 10
014 // Объявление объекта lightMeter
015 BH1750 lightMeter;
016 // Объявление переменной для контроля состояние освещенности
017 boolean ledState;
018 // Объявление объекта для управления дисплеем,указание контактов
019 LiquidCrystal LCD(12, 11, 5, 4, 3, 2);
020 // Объявление и инициализация переменных и массива для настройки клавиатуры
021 const byte ROWS = 3;
022 const byte COLS = 1;
023 char keys[ROWS][COLS]={
024 {'1'},
025 {'2'},
026 {'3'}}
027 };
028 // Указание контактов для клавиатуры
029 byte rowPins [ROWS] = {9,8,7};
030 byte colPins [COLS] = {6};
031 // Инициализация внутренней раскладки клавиатуры
032 Keypad keypad = Keypad(makeKeymap(keys),rowPins,colPins,ROWS,COLS);
033 // Объявление и инициализация переменных для функции setTimer()
034 int SEC = 0;
035 int MIN = 0;
036 byte LIGHT = 0;
037 // Объявление переменной для работы с функцией millis()
038 unsigned long timer;
039 // Объявление функции установки таймера
040 void setTimer();
041//Функция инициализации переменных, режимов работы выводов при загрузке устройства
042 void setup() {
043   Wire.begin();//инициализация библиотеки Wire
044   pinMode(LED, OUTPUT);//режим работы контакта светодиода
045   pinMode(REL, OUTPUT);//режим работы контакта реле
046   pinMode(SOUND,INPUT);//режим работы контакта датчика звука
047   pinMode(PIR,INPUT);//режим работы контакта датчика движения
048   // Установка состояния освещенности
049   ledState=false;
050   // Настройка отображения дисплея по умолчанию
051   LCD.begin(16, 2);
052   LCD.setCursor(2,0);
053   LCD.print("TIMER SETUP");
054   LCD.setCursor(5,1);
055   LCD.print(MIN);
```

```

056 LCD.print(" : ");
057 LCD.print(SEC);
058 // Инициализация и запуск датчика освещенности (контакты SCL, SDA)
059 lightMeter.begin();
060 // При загрузке устройство заморгает светодиодом
061 for(int i = 3; i >0; i--) {
062     digitalWrite(LED,HIGH);
063     delay (1000);
064     digitalWrite(LED,LOW);
065     delay(500);
066 }
067 }
068 //Конец функции setup()
069// Функция бесконечного цикла устройства
070 void loop() {
071     int pir = digitalRead(PIR);
072     // Чтение показаний с датчика освещенности
073     float lux = lightMeter.readLightLevel();
074     // Проверка: если освещение тусклое и высокий сигнал от датчика движения
075     if (lux < 100 && pir ==HIGH){
076         // Установление состояния освещения в 1
077         ledState = true;
078         // Засекание времени в течение которого будет гореть свет
079         timer = millis();
080         while(millis()-timer<10000){
081             digitalWrite(REL,ledState);
082             // Доступ к возможности установления таймера
083             setTimer();
084         }
085         // Инвертирование состояния освещенности, чтобы его выключить
086         ledState = !ledState;
087         digitalWrite(REL,ledState);
088     }
089     // Проверка если звук громче установленного,
090     // то инвертируется состояние освещенности.
091     // Засекание времени аналогично с вышеописанному условию
092     if (analogRead(SOUND)>45){
093         ledState = !ledState;
094         timer = millis();
095         while(millis()-timer <5000){
096             digitalWrite(REL,ledState);
097             setTimer();
098         }
099         ledState = !ledState;
100         digitalWrite(REL,ledState);
101     }
102     setTimer();
103 }
104 // Конец функции loop()
105
106 //Подпрограмма установки таймера обратного отсчета
107 // Подпрограмма не принимает никаких параметров
108 // В подпрограмме используются глобальные переменные SEC, MIN, timer
109 void setTimer(){
110     // Чтение нажатой кнопки на клавиатуре
111     char key = keypad.getKey();
112     // Если нажата третья кнопка - добавляется минута и отображается на дисплее
113     if (key == '3'){
114         MIN = MIN+1; SEC=0;
115         LCD.clear();
116         LCD.setCursor(2,0);
117         LCD.print("TIMER SETUP");

```

```

118         LCD.setCursor(5,1);
119         LCD.print(MIN);
120         LCD.print(" : ");
121         LCD.print(SEC);
122     }
123     //Если нажата вторая кнопка - отнимается минута и отображается на дисплее
124     if (key == '2'){
125         MIN = MIN-1; SEC=0;
126         if(MIN<0){
127             MIN=0;
128         }
129         LCD.clear();
130         LCD.setCursor(2,0);
131         LCD.print("TIMER SETUP");
132         LCD.setCursor(5,1);
133         LCD.print(MIN);
134         LCD.print(" : ");
135         LCD.print(SEC);
136     }
137     // Если нажата первая кнопка - запускается таймер обратного отсчета
138     // Таймер отображается на дисплее
139     if(key == '1'){
140         LCD.setCursor(3,1);
141         // Начинается обратный отсчет с 60 секунд
142         if(SEC ==0) {
143             SEC=60; MIN=MIN-1;
144         }
145         if (MIN<0){
146             MIN=0;
147         }
148         // Установка состояние освещенности в true
149         ledState = true;
150         digitalWrite(REL, ledState);
151         // Инициализация переменной для контроля отключения освещения
152         LIGHT=1;
153     }
154     // Пока не закончится время на таймере выполняется уменьшение счетчика
155     while (LIGHT == 1 ) {
156         if (millis() - timer > 995) {
157             timer = millis();
158             SEC = SEC - 1;
159             // Проверка: если время на таймере истекло -
160             // переключение состояния освещенности в false
161             // и вывод информации на дисплей
162             if (SEC == 0 && MIN == 0) {
163                 LCD.clear();
164                 LCD.setCursor(2, 0);
165                 LCD.print("TIMER SETUP");
166                 LCD.setCursor(5, 1);
167                 LCD.print(MIN);
168                 LCD.print(" : ");
169                 LCD.print(SEC);
170                 ledState = false;
171                 digitalWrite(REL,ledState);
172                 LIGHT = 0;
173             }
174             //Если секунды равны 0, то присваивается значение 59
175             // Отнимается 1 минута
176             if (SEC == 0) {
177                 SEC = 59; MIN = MIN - 1;
178                 if (MIN < 0 ) { MIN = 0;
179

```



```
180     }
181     // Проверка переменной контроля отключения:
182     // если 1, то на дисплее отображается время таймера
183     if (LIGHT == 1) {
184         LCD.clear();
185         LCD.setCursor(2, 0);
186         LCD.print("TIMER START");
187         LCD.setCursor(5, 1);
188         LCD.print(MIN);
189         LCD.print(" : ");
190         LCD.print(SEC);
191     }
192 }
193 }
194 }
195 //Конец функции setTimer()
```