

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт машиностроения
(наименование института полностью)

Кафедра Промышленная электроника
(наименование)

13.03.02 Электроэнергетика и электротехника
(код и наименование направления подготовки, специальности)

Интеллектуальные энергетические системы
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему: Микроконтроллерное устройство для регулирования освещённости помещения

Студент

О.А. Полячков

(И.О. Фамилия)

(личная подпись)

Руководитель

А.К.Кудинов

(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

на бакалаврскую работу Полячкова Олега Алексеевича
по теме: «Микроконтроллерное устройство для регулирования
освещенности помещения»

Бакалаврская работа состоит из 52 с., 3 таблиц, 31 рисунка, списка литературы из 28 источников.

Цели работы:

1 Сделать обзор и изучить состояния вопроса по теме микроконтроллерных устройств для регулирования освещенности помещения.

2 Поиск комплектующих соответствующих критериям освещенности.

3 Разработка принципиальной схемы

Степень внедрения — установка по разработанной документации является опытным образцом.

Областью применения данной системы являются офисные пространства и жилые помещения.

В бакалаврской работе (ВКР) проведена разработка и спроектировано на объекте микроконтроллерное устройство для автоматического регулирования освещенности с установкой режима использования помещения и управлением умными стеклами, защищающими от наружного освещения.

Abstract

The title of senior thesis is Microcontroller device for adjusting the room illumination.

The graduation work consists of an introduction, three parts, a conclusion, tables, the list of references, including foreign sources, and a graphic part on 6 A1 sheets.

The key issue of the thesis is the creation of a microcontroller device for automatic regulation of room illumination.

The problem of adjusting lighting in offices and residential premises for a comfortable work as well as maintaining the desired level of room illumination in the range from 0 lux to 500 lux is touched upon in the thesis.

The purpose of the work is to design a microcontroller device, the task of which is to automatically control the level of illumination in the room.

The graduation project can be divided into following connected parts which are analysis of existing solutions; task formulation; searching for components; developing the structure and writing the program.

Finally, we have designed a device for lighting control that meets the original conditions of the problem and calculated the cost of this microcontroller device.

In conclusion, we would like to say that this technology can be used in smart homes and offices.

The work is of interest for wide circle of readers.

Содержание

Введение.....	5
1 Состояние вопроса.....	6
1.1 Анализ исходных данных.....	6
1.2 Обзор известных решений	7
1.3 Формулировка задач работы.....	13
2 Основная часть	15
2.1 Расчет требуемого количества светильников	15
2.2 Выбор комплектующих устройства	19
2.3 Разработка структурной схемы устройства.....	24
2.4 Разработка конструкции устройства	32
2.5 Разработка алгоритма программы устройства.....	37
2.6 Калибровка датчика света устройства	41
2.7 Составление программы устройства.....	42
3 Оценочная часть	48
Заключение	50
Список используемой литературы	51

Введение

В современном мире сохраняется уверенная тенденция, что всё больше компаний считают своим главным ресурсом интеллектуальную силу своих рабочих. Взгляните на современные офисы - они созданы, чтобы человек, работающий в них, чувствовал себя максимально комфортно и не хотел покидать организацию, где сейчас работает. Фитнес залы, игровые, живые сады внутри офиса, бассейны, ланч зоны и многое другое, является неотъемлемой частью какой-нибудь современной it компании.

В связи с этим предлагается сделать устройство для регуляции освещения, с помощью которого не только можно будет сэкономить на электроэнергии, но что более важно, уберечь сотрудников от излишнего или избыточного освещения на рабочем месте. Это позволит повысить психическую работоспособность, снять излишнее напряжение с глаз, тем самым уменьшив шанс затруднений со здоровьем в будущем.

Устройство не обладает громоздкостью, благодаря чему, при его установке не требуются кардинальные перепланировки в дизайне уже существующих офисных помещений. Это достигается путём полной модульности всей конструкции и благодаря этой же особенности можно спокойно устанавливать систему как в маленькие (одиночные) офиса, так и в большие офисные пространства занимающие целые этажи. Настройка системы позволит не закреплять за помещениями статусов офиса или условной столовой, благодаря выбору настроек внутри, можно быстро задать новые функции, что позволит выбирать разные режимы работы для одного и того же помещения. Это позволит на время условной встречи превратить офис в переговорную, с комфортными условиями для всех находящихся внутри людей.

Так же к очевидным достоинствам такой системы дома можно отнести возможность её интеграцию в систему умный дом.

1 Состояние вопроса

1.1 Анализ исходных данных

В соответствии с заданием на выполнение бакалаврской работы разрабатываемое микроконтроллерное устройство для регулирования освещенности помещения питается от однофазной сети с напряжением 220В и частотой переменного тока 50Гц, управляет потолочными светодиодными светильниками и умными стеклами с регулируемым светопропусканием, обеспечивая при этом автоматическое управление освещенностью на рабочем месте в зависимости от времени суток, наружного освещения и назначения помещения, которое задается в ручную.

В задании предполагается использование устройства в помещении с площадью пола 3х6 метров и высотой потолков 3 метра. В помещении есть одно окно с умными стеклами в стеклопакетах.

Нормы освещенности берем из СНиП 23-05-95 в зависимости от назначения помещения. Рассматриваемое в задании на проектирование помещение может быть использовано как:

- кладовая комната с нормируемой освещенностью в 50 лк;
- книгохранилище с нормируемой освещенностью в 75 лк;
- спальная комната с нормируемой освещенностью в 100 лк;
- жилая комната с нормируемой освещенностью в 150 лк;
- учебный кабинет информатики и вычислительной техники с нормируемой освещенностью в 200 лк;
- офисное помещение с нормируемой освещенностью в 300 лк;
- лаборантская комната с нормируемой освещенностью в 400 лк;
- учебный кабинет технического черчения и рисования с нормируемой освещенностью в 500 лк.

Причём такая освещённость должна достигать максимума на рабочей

поверхности (обычно принимается величина — 0,8 метров от пола) [7].

На эффективность распространения лучей ламп может повлиять высота помещения, чем она выше – тем дальше источник света от плоскости стола.

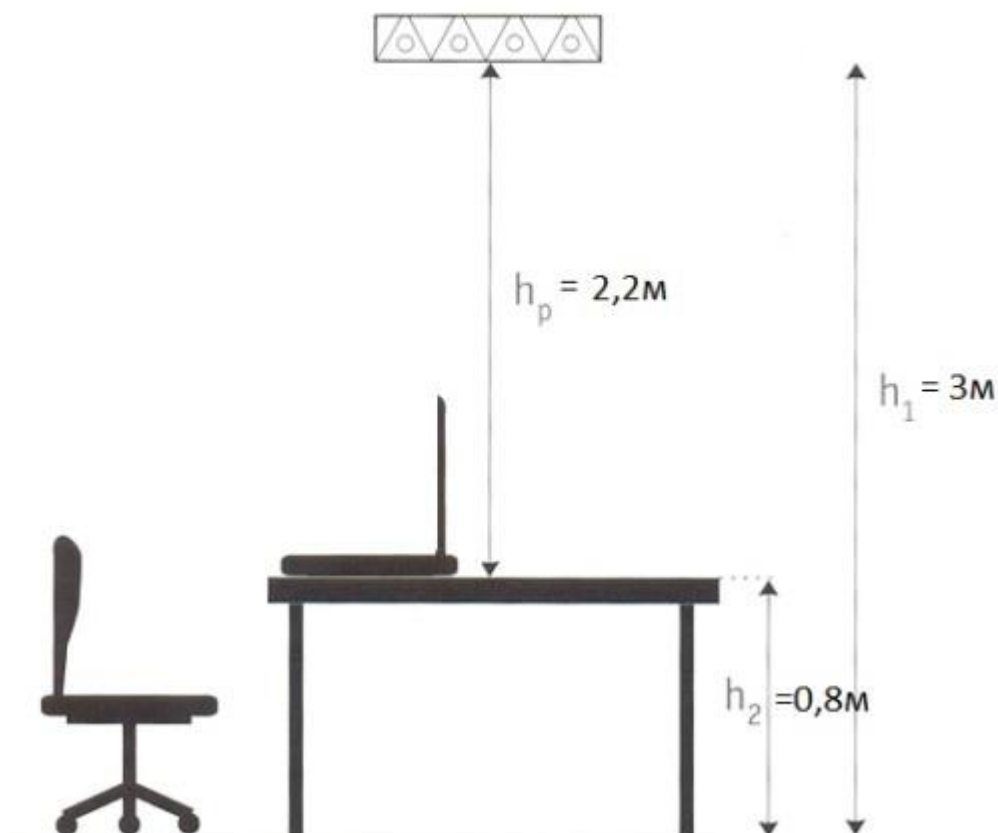


Рисунок 1 – Расположение потолочного светильника над рабочим местом

1.2 Обзор известных решений

Управление световыми приборами с помощью микроконтроллеров, на сегодняшний день обыденная операция при инсталляции систем «Умного дома», обладающих очень широкими возможностями. Разработано и выпускается множество приборов для автоматизации «Умного дома». Например, наиболее популярные на отечественном рынке контроллеры ZONT C2000+ и Wiren Board 4 могут устанавливаться на DIN-рейку [13]. Система «Умного дома» FIBARO Homecenter 2 имеет развитый

беспроводный интерфейс с современным графическим решением терминалов управления, что очень нравится большинству потребителей.

Управление освещением является одной из ключевых функций в комплексе «умный дом». Автоматическое регулирование рассчитано на категорию пользователей, стремящихся жить в комфортных условиях, но при этом старающихся экономить средства. Для обустройства «умного дома» компания Яндекс выдала ряд разработок, вписывающих голосового ассистента в систему управления. Запустить приложение можно со смартфона или планшета. Для управления с мобильного устройства нужно активировать ассистента и после этого диктовать команду. Для активации обращаются к Алисе или Яндексу. Аналогичные решения есть у умных колонок Amazon и Google, сервисов Apple и Microsoft, и многих других.

Рассмотрим возможности некоторых наиболее популярных на российском рынке контроллеров «умного дома».

ZONT C-2000+ универсальный контроллер для системы "умный дом", предназначен для Охраны, Мониторинга и Автоматизации (рис.2) [12]. Обеспечивает контроль и дистанционное управление различными устройствами через веб-сервис и мобильное приложение. Оповещает при срабатывании датчиков и других нештатных ситуациях.



Рисунок 2 – контроллер ZONT C-2000+

Осуществляет управление каскадом котлов, бойлером ГВС, прямыми и смесительными контурами, циркуляционными насосами, а также любыми электрическими приборами и поддерживает заданные температурные диапазоны каждого контура. Обеспечивает дистанционный контроль параметров системы через веб-сервис и мобильное приложение и оповещает при срабатывании аварийных датчиков и других нештатных ситуациях (рис.3) [19].



Рисунок 3 – применение контроллера ZONT C2000+

Особенности контроллера ZONT C-2000+:

- любое количество контролируемых датчиков и зон;
- позонная охрана и управление доступом;
- ситуационное поведение;
- настройка сценариев автоматизации;
- дистанционное управление э/приборами;
- поддержка цифровых и аналоговых датчиков температуры;
- модули расширения входов/выходов/интерфейсов;
- прямое подключение к Ethernet;
- мобильный интернет (GSM/GPRS);
- интерактивная настройка через WEB интерфейс;

- 12 выходов: 6 - релейных, 6 -универсальных;
- K-line вход для подключения адаптеров цифровой шины (OpenTherm, E-BUS, Navien) для систем отопления.

При подключение управляемых элементов (сервоприводов, насосов, котлов в релейном режиме, сервоприводов теплого пола, бойлера ГВС и пр.) к устройствам ZONT H-1000, ZONT H-2000, ZONT H-2000+, C-2000+ и панели расширения ZE-66 к универсальным выходам (6 шт.) обязательно использование промежуточного реле РЭК78/3 [5].

Wiren Board 4 - универсальный контроллер для автоматизации с открытым ПО на базе Linux, предназначен для домашней и промышленной автоматизации и мониторинга: опроса датчиков и счетчиков, использования в качестве УСПД, в системах АСКУЭ, в системах “умного дома” (рис.4).



Рисунок 4 – контроллер Wiren Board 4

Характеристики контроллера Wiren Board 4:

- беспроводная связь: Wi-Fi, GSM/GPRS, 433 МГц;
- порты: Ethernet 10/100, USB 2.0 [9], 2 оптоизолированных RS-485, 1-wire, опции: RS-232, CAN;

- ВХОДЫ/ВЫХОДЫ:

5 входов АЦП до 30 В;

2 входа для резистивных датчиков;

4 входа для датчиков “сухой контакт”;

9 выходов “открытый коллектор”;

2 реле 250В/2А;

- корпус: 103x87x20 мм, на DIN-рейку;

- условия эксплуатации: 0..70С (-40..85С по запросу);

- Watchdog, динамик, часы реального времени RTC, влагозащитное покрытие, разъём для подключения Li-Ion аккумулятора [11].

Категории поддерживаемых устройств:

- водо- и энергосчётчики с импульсными выходами и RS-485;

- устройства управления подачей питания (релейные модули) с интерфейсом RS-485;

- цифровые и аналоговые датчики, в том числе удалённые по RS-485.

Измеряемые параметры: температура, влажность, освещённость, наличие газов, давление;

- устройства управления освещением: релейные модули, диммеры, контроллеры светодиодных лент, работающие по RS-485, радио 433 МГц.



Рисунок 5 – система умного дома FIBARO Homecenter 2

FIBARO Homecenter 2 занимает среднюю нишу на рынке домашней автоматизации - это не дешевая система, но в то же время она не стоит безумных денег.

Возможности этой системы намного больше рассмотренных ранее. Особенностью FIBARO Homecenter 2 является использование надежной беспроводной сети Z-WAVE для связи с датчиками и исполнительными устройствами, а также встроенная защита от взлома извне и от DDoS атак.

Но все же наиболее популярным на рынке является самостоятельное изготовление разного рода автоматики на платформах быстрого прототипирования, таких как Arduino и ей подобных [2].

К устройствам «умного дома» можно также отнести смарт плёнку (SMART FILM) - листовой полимерный материал типа «самоклейка», наносится на стекло, подключается к контроллеру. Напряжение может изменяться в диапазоне от 12В до 36 В, и подается на стеклянную поверхность по скрытой проводке.

Работа электронного стекла основывается на поляризации микрочастиц (PDLC / LCD, SPD, ECD) в электромагнитном поле. Под воздействием электрического тока стеклянная поверхность изменяет светопропускную способность. Под напряжением прозрачная, без напряжения белая матовая.

Окна с регулируемой прозрачностью получили особые свойства за счет полимерной жидкокристаллической пленки, расположенной между листами стекла. Жидкие кристаллы без электричества приходят в твердое состояние, на пленке образуются вкрапления, расположенные в хаотичном порядке, и рассеивающие световые лучи. Такое состояние позволяет получить матовую поверхность.

Изменение переменной прозрачности происходит моментально. При этом не возникает полутона. В окнах с регулированием величины пропускания света степень тусклости меняется медленно. Под действием

разных значений напряжения тока можно легко установить необходимую степень пропускания световых лучей.

На ее основе собирают смарт стеклопакеты. Кроме функции изменяемой прозрачности они обладают повышенной шумоизоляцией, ударопрочностью и блокируют значительную часть УФ и ИК излучения.

В непрозрачном состоянии смарт-пленка и смарт-стекло могут работать как экран обратной проекции. Если направить проектор на одну сторону смарт-пленки, то с другой стороны будет картинка. Эта особенность используется для создания видеовитрин. В дневное время проекцию не видно, на пленку подается напряжение, стекло прозрачно. Вечером и ночью, когда проекция эффективна, пленка переводится в непрозрачное состояние, включается проектор – витрина или окно превращается в большой рекламный дисплей.

1.3 Формулировка задач работы

В соответствии с заданием, необходимо разработать микропроцессорный регулятор света с установкой уровня освещенности исходя из применения помещения. Необходимо рассчитать требуемое количество светильников и выбрать их тип для обеспечения требуемой максимальной освещенности в 500 лк. Другие уровни освещенности обеспечиваются понижением напряжения или тока питания светодиодных линеек светильников.

Необходимо также рассчитать влияние внешнего света на освещенность на рабочем месте и рассчитать уровни регулировки светопропускания умных стекол, встроенных в стеклопакеты окна помещения.

Структура устройства должна обеспечивать автоматическую регулировку питания потолочных светильников и умных стекол исходя из

пользовательского режима работы и замера уровня освещенности на рабочем месте с учетом времени суток (день-ночь, восход и заход солнца).

Таким образом, структура разрабатываемого устройства должна состоять из следующих блоков:

- микроконтроллер (МК);
- блок измерения освещенности на рабочем месте (датчик света);
- панель программирования и индикатор МК;
- блок управления нагрузками;
- блок питания AC 220В, 50Гц / DC 5В.

Для функционирования микроконтроллерное устройство для регулирования освещенности помещения (диммер) необходимо написать программу (sketch), выполняющую следующие функции:

- выбор режима управления;
- сохранение настроек устройства в энергонезависимую память;
- суточный таймер с установкой времени перехода день-ночь и обратно;
- автоматическая регулировка освещенности с управлением двумя нагрузками (светильник и умное стекло) в зависимости от времени суток и уровня освещенности на рабочем месте, измеренного датчиком света;
- индикация состояния устройства на экране.

Основная петля регулировки освещенности обеспечивает изменение питания потолочных светильников, дополнительная петля регулировки обеспечивает снижение уровня освещенности от солнца уменьшением светопропускания умных стекол. С целью безопасности, обеспечивая «защиту от посторонних глаз», светопропускание умных стекол может меняться по желанию пользователя.

Разработанный Sketch записывается в постоянную память МК, через соответствующие программные средства (IDE) с ПК, в соответствии с инструкцией по программированию МК.

2 Основная часть

2.1 Расчет требуемого количества светильников

Проведём расчет освещенности исходя из требуемых максимальных параметров:

- помещение с площадью пола 3х6 метров и высотой потолков 3 метра;
- максимальная освещенность на рабочем месте 500 лк;
- светодиодные потолочные светильники со световым потоком 2800 лм.

Будем учитывать поправки на цвет и степень отражаемости всех поверхностей [14]. Допустим все поверхности покрыты белой побелкой, пол - серый, а значит, индексы отражения составляют: для потолка - 80, для стен - 80, для пола – 30 (табл.1).

Таблица 1 – коэффициенты отражения

Отражающая поверхность	Коэффициент отражения, %
Плоскость с белой поверхностью (побеленный потолок, побеленные стены с окнами, закрытыми белыми шторами)	80
Плоскость со светлой поверхностью (побеленные стены при не завешанных окнах; побеленный потолок в сырых помещениях; чистый бетонный и светлый деревянный потолок)	50
Плоскость с серой поверхностью (бетонный потолок в грязных помещениях; деревянный потолок; бетонные стены с окнами; стены оклеенные светлыми обоями)	30
Плоскость с темной поверхностью (стены и потолки в помещениях с большим количеством темной пыли;	10
сплошное остекление без штор; красный не оштукатуренный кирпич; стены с темными обоями)	10

Определяем индекс помещения I_p по формуле 1 [14]:

$$I_p = S/((h_1-h_2)*(a+b)) = 3*6/((3-0,8)*(3+6)) = 0,91, \quad (1)$$

где $S=a*b$ – площадь пола помещения в кв.м.;

h_1 – высота подвеса светильников в метрах;

h_2 – высота поверхности стола от пола в метрах.

Воспользовавшись таблицей 2, определяем коэффициент использования (U) – в нашем случае он составит 65.

Таблица 2 – использование отражения от поверхностей

		Отражение поверхностей							
Потолок		80	80	80	70	50	50	30	0
Стены		80	50	30	50	50	30	30	0
Пол		30	30	10	20	10	10	10	0
Индекс помещения	0,6	53	38	32	37	35	31	31	27
	0,8	60	45	38	44	41	38	37	34
	1	65	51	43	49	46	43	42	38
	1,25	70	57	49	54	51	48	47	44
	2	76	66	56	61	57	55	54	51
	2,5	78	70	59	64	60	58	57	54
	3	80	73	62	67	62	60	59	57
	4	81	76	64	69	63	62	61	58
	5	82	78	65	70	65	64	62	60

Расчет количества светильников $K_{св}$ производим по формуле 2:

$$K_{св} = (E*S*100*K_3)/(U*СП_{л}) = (500*3*6*100*1,2)/(65*2800) \approx 6 \text{ шт.}, \quad (2)$$

где E – освещенность на рабочем месте в лк;

S – площадь пола в кв.м.;

$K_3=1,2$ – коэффициент запаса на старение ламп и запыленность помещения;

U – коэффициент использования;

$СП_{л}$ – световой поток одного светильника.

Таким образом, для рассматриваемого помещения рекомендуется не менее 6 потолочных светодиодных светильников, таких как, например, светильник для потолка типа Амстронг размерами 595*595мм и потребляемой мощностью 36 Вт, который создает световой поток в 2800 лм. Предлагаемое расположение светильников на потолке показано на рис.6. Суммарная мощность потребления 216 Вт [7].

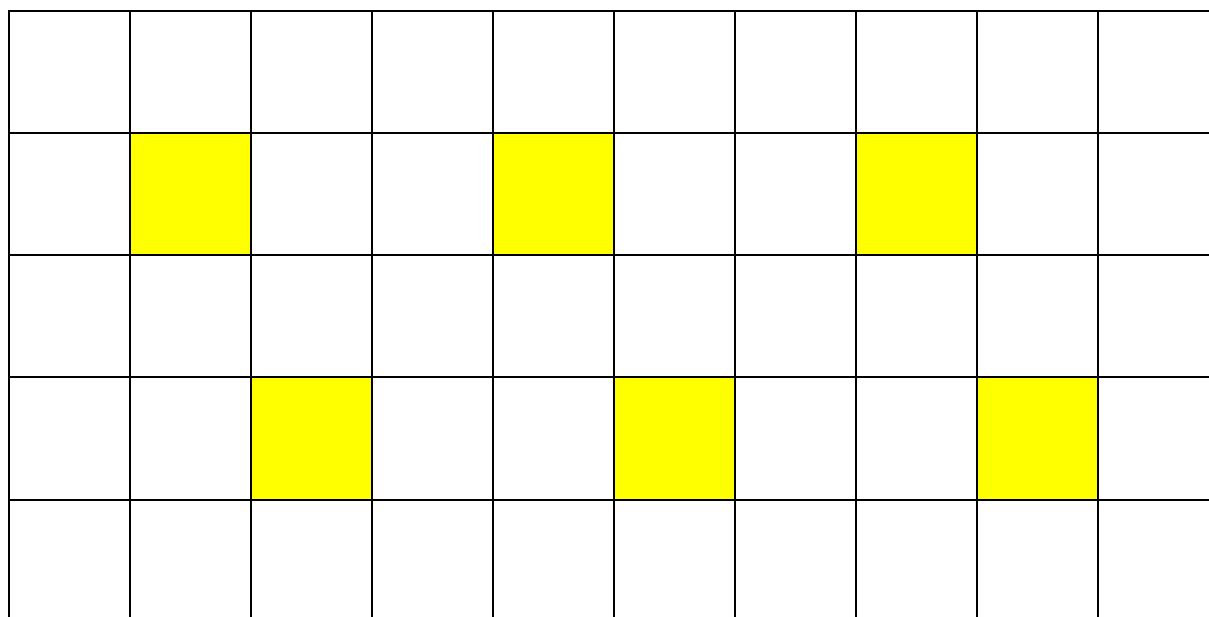


Рисунок 6 – примерное расположение светильников на потолке

Можно рассмотреть применение точечных светильников. Требуемую суммарную интенсивность освещения $I_{осв}$, выражаемую в лк/м² или лм, определим для площади пола в 18 кв.м., исходя из требуемой максимальной освещенности в 500 лк [14] по формуле 3:

$$I_{осв} = S * E * k = 18 * 500 * 2 = 18000 \text{ лм.} \quad (3)$$

где S – площадь пола в кв.м.;

E – освещенность на рабочем месте;

k – поправочный коэффициент на высоту потолка.

Светодиодные лампы обладают большой эффективностью использования питания (КПД) по сравнению с лампами накаливания, и существует определенная зависимость для ориентировочного расчета необходимого количества точечных светильников, отображенная на рис. 7.



Рисунок 7 – сравнение характеристик ламп накаливания и светодиодов

На рис. 8 приведено предлагаемое расположение 46 точечных светильников.

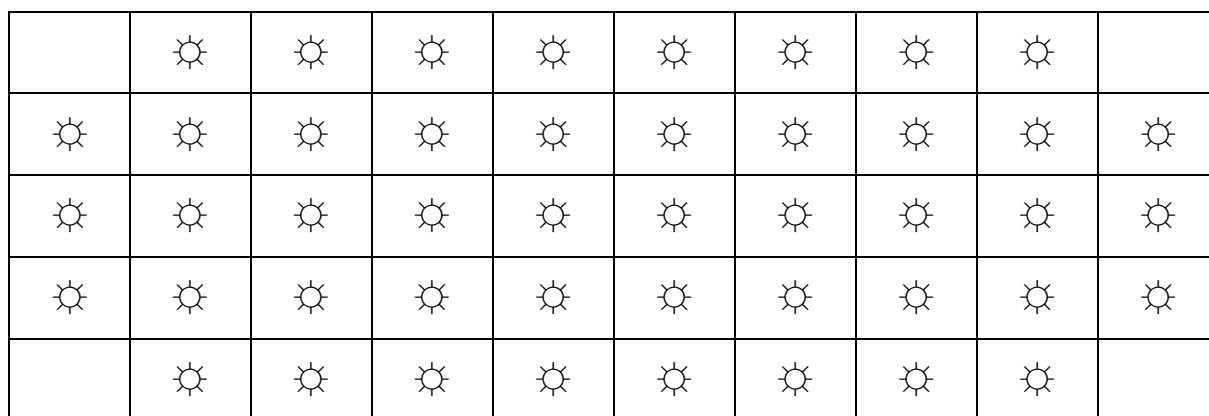


Рисунок 8 – примерное расположение светильников на потолке

Если в рассматриваемом помещении применяется потолок типа Армстронг с плитками 600*600мм, то, учитывая площадь плитки в 0,36 м², плиток будет 50 шт. А это значит, что практически в каждой плитке должен

стоять точечный светильник со световым потоком в 400лм [1]. Исходя из выше сказанного можно рекомендовать реализацию светового решения на точечных светодиодных светильниках мощностью 6,3Вт питанием 12В, применение которых допускается с диммером, в количестве $18000/400 = 45$ штук, что составляет $18/45 = 0,4$ шт/м² или 5 штук на 3 м². Суммарная мощность светильников составит $45*6,3 = 283,5$ Вт. Можно использовать понижающий трансформатор 230В/12В [19], например, трансформатор Aquant 300Вт-12В. изображённый на рис.9.



Рисунок 9 – Внешний вид понижающего трансформатора Aquant 300Вт-12В

2.2 Выбор комплектующих устройства

В качестве МК выбран Arduino MEGA2560 [8], имеющий в своем составе 6 программируемых таймеров: 2 восьми битных (макс. отсчитываемое значение 256) и 4 шестнадцати битных (макс. отсчитываемое значение 65 536).

Другие платы Arduino на основе 8-ми битного процессора, и подобные им, содержат в своем составе 8-ми битный таймер, за исключением платы Duo, имеющей 32-битный процессор и более высокую стоимость [12].

Другой возможный вариант решения задачи – использование внешних часов реального времени (RTC) с кварцевой стабилизацией и батарейным питанием, например, DS3231.

МК Arduino MEGA2560 построен на основе БИС ATmega2560 с тактовой частотой 16МГц [4]. МК имеет 54 порта ввода/вывода; 15 выходных портов могут работать, как выход с ШИМ сигналом (аналоговый выход) [21]; 16 аналоговых входов с АЦП могут обрабатывать сигналы с аналоговых датчиков. Ток выдаваемый каждым портом в нагрузку – не более 40 мА [6].

Внешнее питание 7÷20В с понижением через встроенный в плату стабилизатор 5В к выходу которого возможно подключение внешних плат и датчиков с суммарным током не более 800мА [19].

Для хранения программы предусмотрена Flash-память размером 256 КБ (8 КБ из них используются загрузчиком); оперативная память ОЗУ SRAM размером 8 КБ; энергонезависимая память EEPROM размером 4 КБ [10].

Для связи с разными устройствами предусмотрено 4 UART интерфейса, в их роли выступают выводы 0, 1, 14-19. Один из портов направлен на USB через микроконтроллер ATmega16U2 [20].

На плате предусмотрены разъемы для установки плат расширения Shield. Размеры платы: 53x110x12,5мм. Разъем внешнего питания под штекер диаметром 2,1мм с плюсом по середине. Внешний вид платы с двух сторон приведен на рисунке 10.

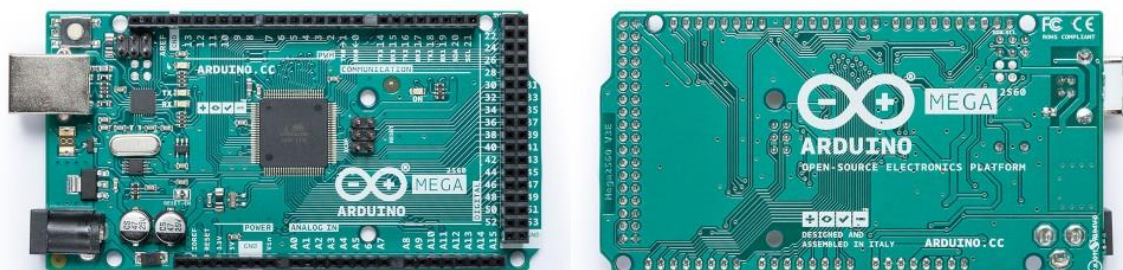


Рисунок 10 – Внешний вид платы Arduino MEGA2560

Панель управления устройством и индикатор МК, выбраны с учетом совместимости с МК Arduino MEGA2560 и стоимости реализации, в виде готовой платы расширения Arduino LCD Keypad Shield [16]. Внешний вид платы с двух сторон приведен на рисунке 11.



Рисунок 11 – Внешний вид платы расширения LCD Keypad МК

Плата расширения LCD Keypad Shield содержит: индикатор LCD1602, позволяющий отображать два ряда по 16 символов, подключенный к четырем цифровым выходам Arduino D4-D7; шесть кнопок (RIGHT, UP, DOWN, LEFT, SELECT), подключенных к аналоговому входу Arduino A0; кнопка RESET, подключенная к входу RST; резистор настройки контрастности дисплея LCD. Размеры платы 60x83x20мм.

Датчик освещенности на основе фоторезистора применяется в виде готовой платы Z-light sensor [11] (рис. 12).

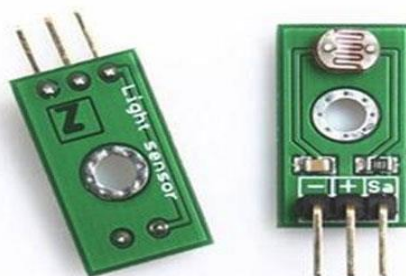


Рисунок 12 – Внешний вид платы датчика освещенности

В качестве блока управления нагрузками выбрана готовая плата RobotDyn AC Light Dimmer 4 channel, имеющая четыре независимых канала регулировки нагрузок переменного напряжения 220/380В (до 600В в импульсе) с частотой 50/60Гц и током до 16А в импульсе. Внешний вид платы с двух сторон приведен на рисунке 13.



Рисунок 13 – Внешний вид платы управления нагрузками

На плате размещены 4 симмистора ВТА16-600V с оптопарами МОС3021S [17]. Есть детектор перехода напряжения питания 220В через ноль. Плата предназначена для плавной регулировки света. Выбор готовой платы сделан исходя из соображений невысокой стоимости и удобства применения данной платы. Размеры платы 66x113x30мм.

Импульсный блок питания обеспечивает низковольтную часть электронного таймера напряжением 9В. Питание поступает на плату МК, где расположен стабилизатор на микросхеме МС33269D-5.0 (аналог LM117-5), формирующий 5В с током нагрузки до 800мА. Все платы устройства питаются от 5В, при этом токи потребления [4]:

МК Arduino MEGA2560 (CPU+I/O) $I_{\text{потр}} \approx 50\text{мА}$;

LCD Keypad Shield (LCD) $I_{\text{потр}} \approx 1\text{мА} + 130\text{мА}$ (Back Light LED);

RobotDyn AC Light Dimmer 4 channel $I_{\text{потр}} \approx 20\text{мА} * 4 = 80\text{мА}$.

Итого суммарно: $I_{\text{потр}} \approx 260\text{мА}$.

Выберем блок питания импульсного типа со стабилизацией Coreset SM PLG06A-09 с параметрами AC 220В, 50Гц / DC 9В, 600мА. Внешний вид

платы с двух сторон приведен на рисунке 14.

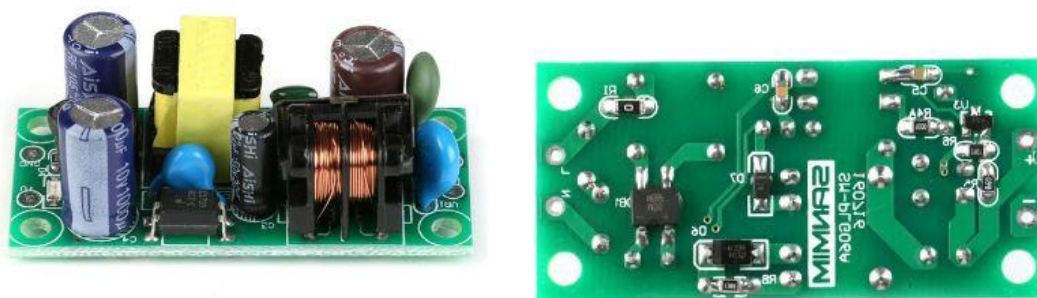


Рисунок 14 – Внешний вид платы блока питания МК

Блок питания импульсный стабилизированный с фильтром ЭМП на входе. При его применении на стабилизаторе 5В МК будет выделяться мощность порядка 1Вт. Можно уменьшить мощность рассеяния до 0,5Вт [8], перенеся питание Back Light LED на шину VIN=9В. Размеры платы: 51x24x19мм.

Металлический корпус с розетками для нагрузок разрабатывается для настольного применения. На передней панели располагаются: расположенные в ряд розетки для включения нагрузок, общий выключатель питания, ниже - экран LCD и кнопки управления электронным таймером, сбоку матовый светофильтр светового датчика [24]. Регулировка контрастности LCD и кнопка RESET для сброса МК доступны через отверстия в панели возле индикатора.

Загрузка Sketch относится к сервисным функциям устройства, поэтому производится при снятой крышке корпуса, путем передачи данных с ПК через кабель USB [16]. Передняя панель съемная для сервисного обслуживания и доступа к USB МК [4].

2.3 Разработка структурной схемы устройства

Принципиальная схема основной процессорной части МК Arduino MEGA2560 [9] изображена на рисунке 15.

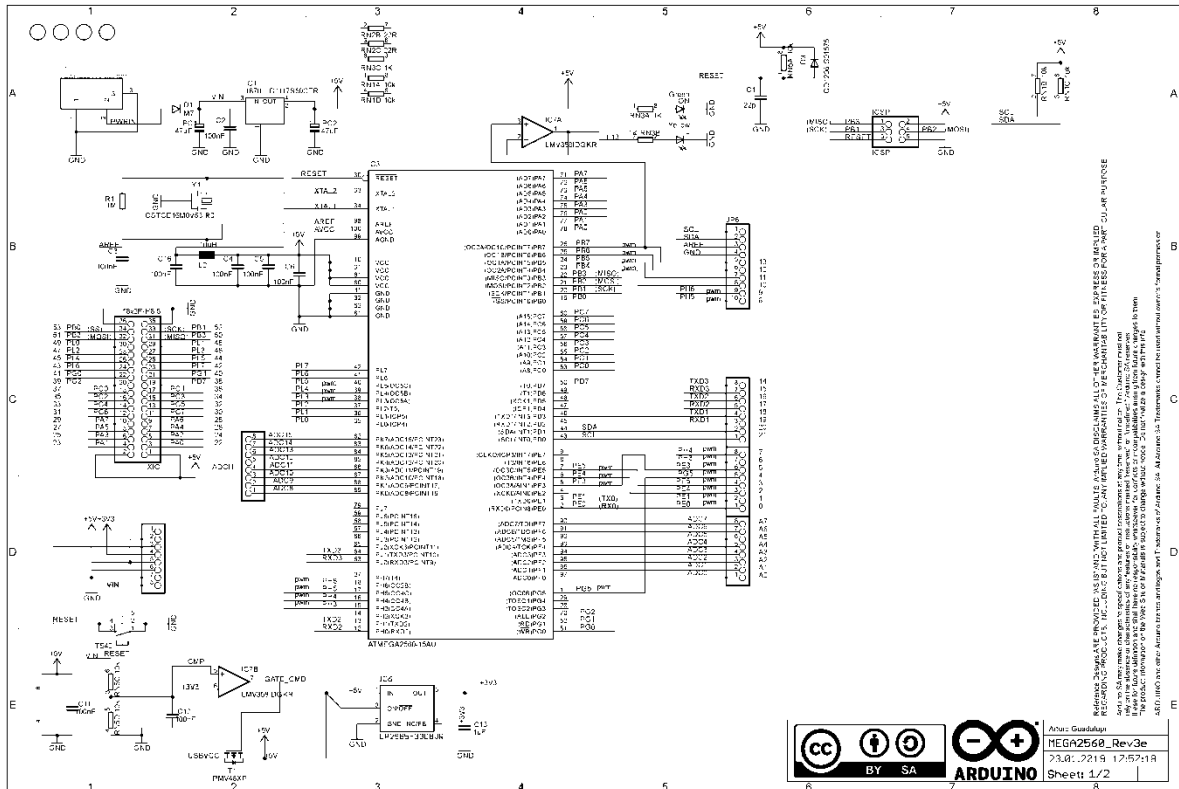


Рисунок 15 – Принципиальная схема основной процессорной части МК

На этой схеме изображены основные элементы: центральный микроконтроллер ATmega2560-15AU и кварцевый резонатор на 16МГц, задающий основную тактовую частоту, интегральный стабилизатор на 5В типа LD1117S50 или аналог, интегральный стабилизатор на 3,3В типа LP2985-33 или аналог, светодиоды, показывающие режимы работы микроконтроллера и несколько элементов обвязки микроконтроллера обеспечивающих нормальный режим работы микропроцессора [21].

Принципиальная схема дополнительного микроконтроллера загрузки с USB, расположенного на плате Arduino MEGA2560 [22], изображена на рис. 16.

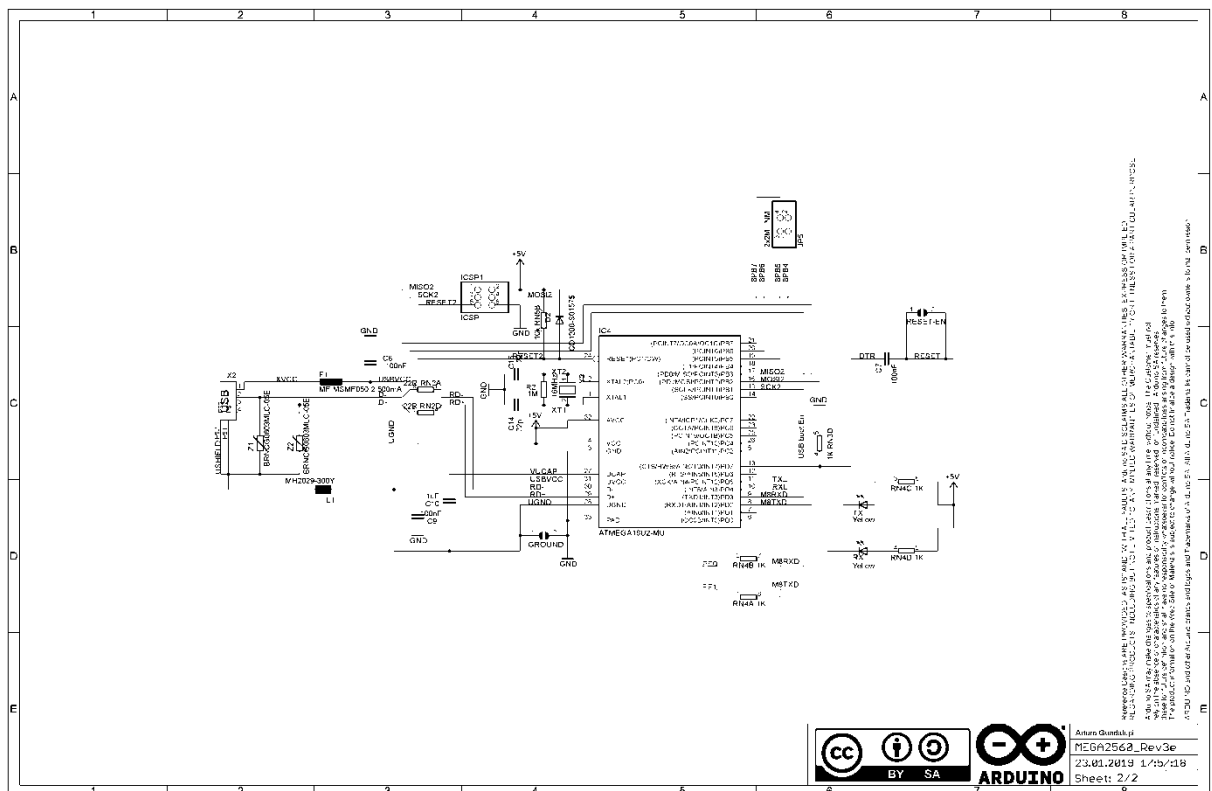


Рисунок 16 – Принципиальная схема контролера USB МК

На схеме изображены основные элементы: микроконтроллер ATmega16U2-MU и кварцевый резонатор на 16МГц, задающий основную тактовую частоту этого контроллера, USB-порт, светодиоды, показывающие режимы работы микроконтроллера и несколько элементов обвязки микроконтроллера обеспечивающих нормальный режим работы микропроцессора [26].

Принципиальная схема платы датчика освещенности приведена на рисунке 17.

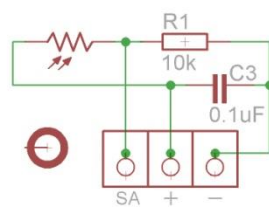


Рисунок 17 – Принципиальная схема датчика освещенности

Здесь применен фоторезистор с цепью смещения на резисторе R1 [9]. Конденсатор C3 служит для уменьшения импульсных помех аналоговому входу Arduino, к которому подключается плата датчика освещенности. Фоторезистор промышленностью выпускается с большим разбросом параметров и применение разрабатываемого устройства возможно в разных помещениях с разными световыми приборами, поэтому в нашем устройстве применяется процедура калибровки показаний освещенности на рабочем месте по внешнему образцовому люксметру. Калибровка проводится единовременно при инсталляции устройства в конкретном помещении.

На рисунке 18 приведена принципиальная схема платы управления и индикации Keypad LCD Shield [3].

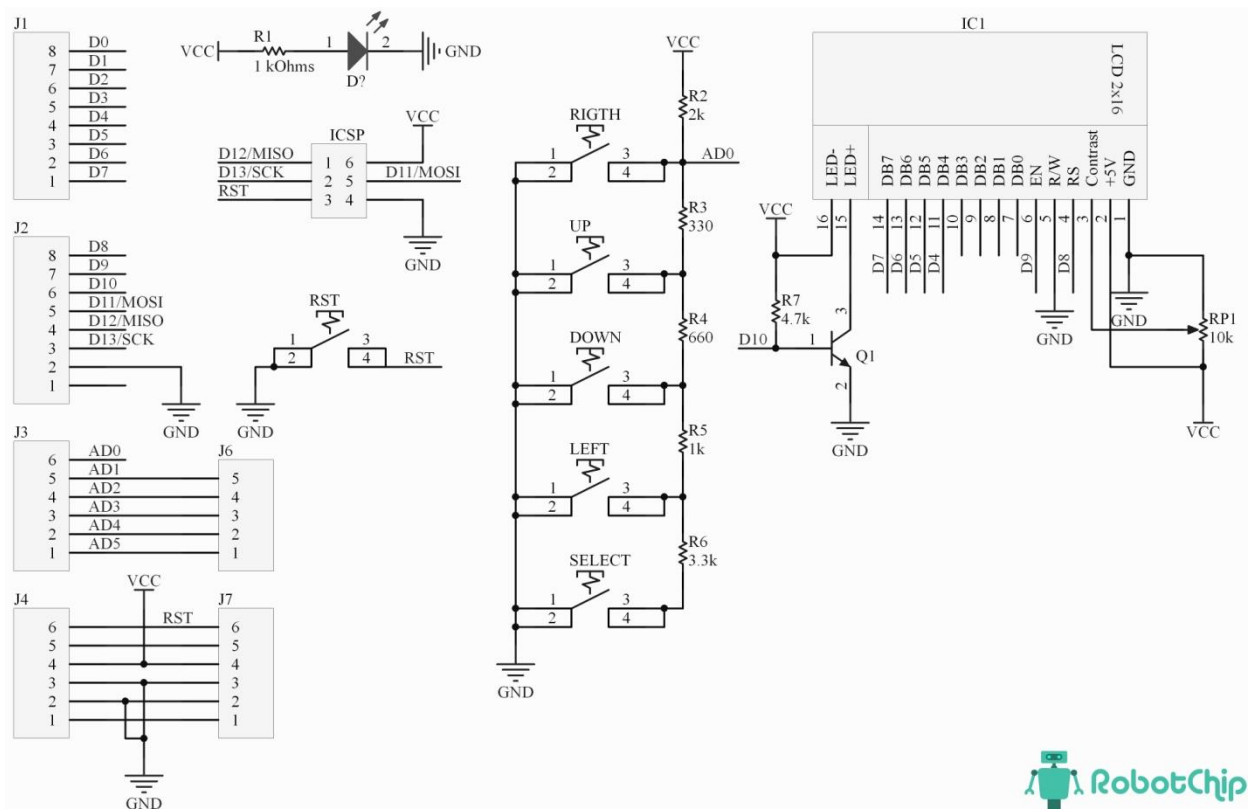


Рисунок 18 – Принципиальная схема платы индикации и управления

На схеме изображены:

- разъемы платы расширения J1-J7 и ICSP;
- светодиод индикации питания VCC;

- кнопка RST, подключённая к выводу RST платы Arduino, необходимая для сброса микроконтроллера, т.к. плата расширения закрывает кнопку RESET на плате Arduino;

- кнопки RIGHT, UP, DOWN, LEFT и SELECT с резисторной матрицей подключаются к аналоговому входу AD0 платы Arduino;

- элементы индикатора LCD1602;

- резистор регулировки контрастности изображения RP1, транзистор коммутации подсветки Q1 по выводу D10 платы Arduino.

На рисунке 19 изображена часть схемы AC Light Dimmer Module.

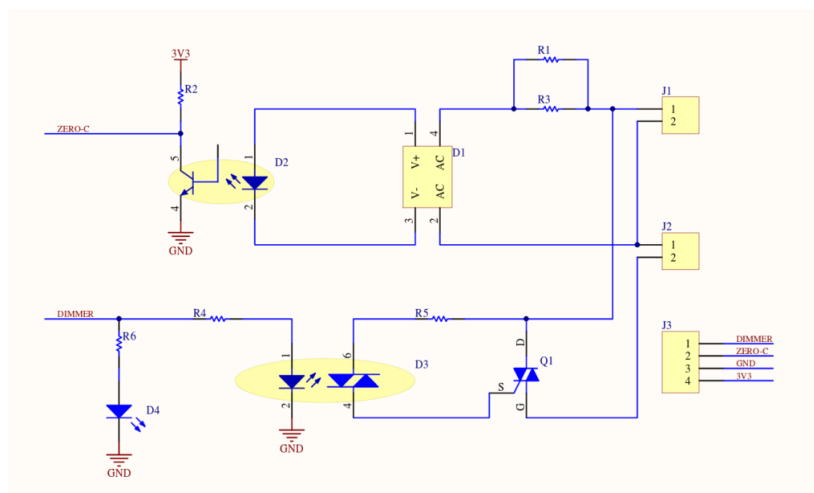


Рисунок 19 – Часть принципиальной схемы платы управления нагрузками

Схема выполнена на 4 симмисторах Q1 типа BTA16-600V с оптической развязкой D3 на динисторных оптопарах MOC3021S и индикацией состояния входа на светодиоде D4 [2]. Резисторы R4-R6 служат для ограничения тока через полупроводниковые элементы и задания рабочих режимов. На оптопаре D2, диодном мосте D1 и ограничительных резисторах R1-R3 построена схема детектора нуля, необходимая для правильного переключения симмистора во время перехода сетевого напряжения питания нагрузки через нуль.

На рисунке 20 приведена принципиальная схема импульсного блока питания Coreset SM-PLG06A-09, собранного по трансформаторной схеме

ИППН типа FlyBack [1], на специализированной интегральной микросхеме с входным фильтром ЭМИ, регулировкой выходного напряжения через оптопару типа PC817 и интегральный стабилизатор типа TL431.

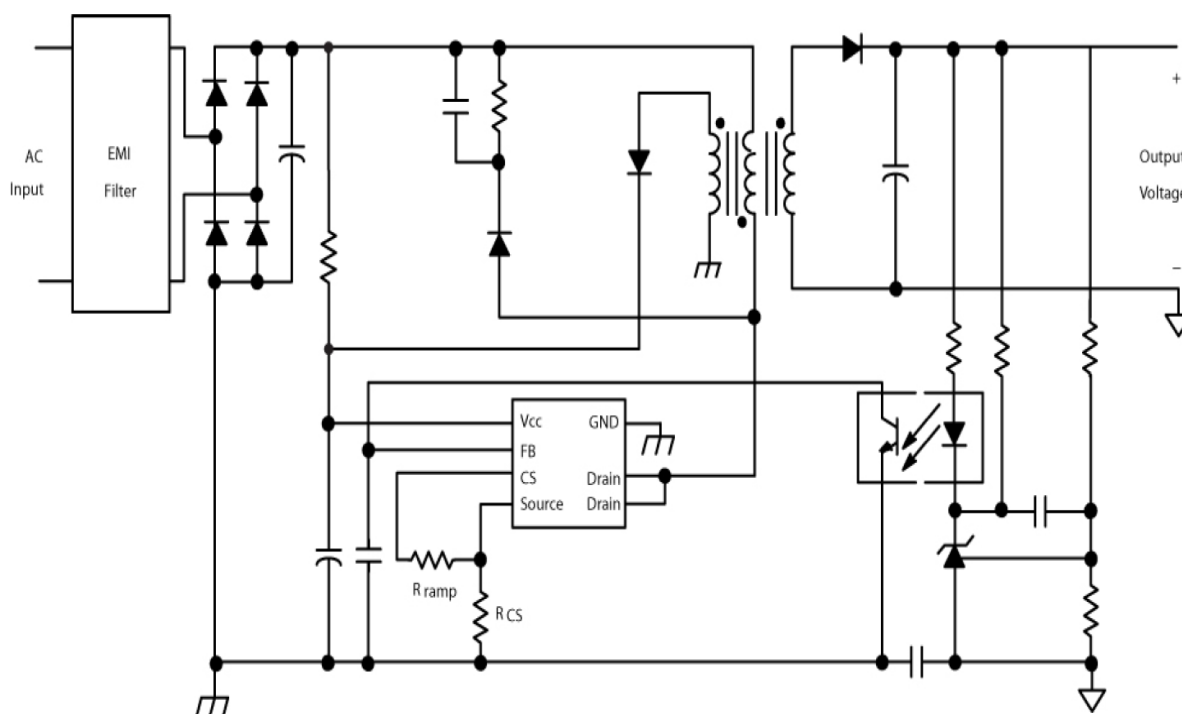


Рисунок 20 – Принципиальная схема импульсного блока питания

После схемы подавления импульсных помех (EMI Filter) переменное напряжение поступает на диодный мост. На выходе диодного моста пульсации выпрямленного напряжения сглаживаются электролитическим конденсатором. Выводы микросхемы Drain защищены от импульсных перенапряжений цепочкой из диода, резистора и конденсатора [8]. Защита по току организована через внешний резистор R_{CS} , подключенный к выводу Source микросхемы. Питание микросхемы осуществляется с вторичной обмотки импульсного трансформатора, выпрямляется диодом и сглаживается электролитическим конденсатором. Для облегчения первичного запуска, подается дополнительное питание через высокоомный резистор с диодного сетевого моста. Со вторичной обмотки трансформатора импульсное напряжение выпрямляется диодом и сглаживается электролитическим конденсатором. Автоматическая регулировка выходного напряжения

осуществляется через интегральный стабилизатор TL431 [23], на вход которого выходное напряжение попадает через резисторный делитель.

MEGA PINOUT

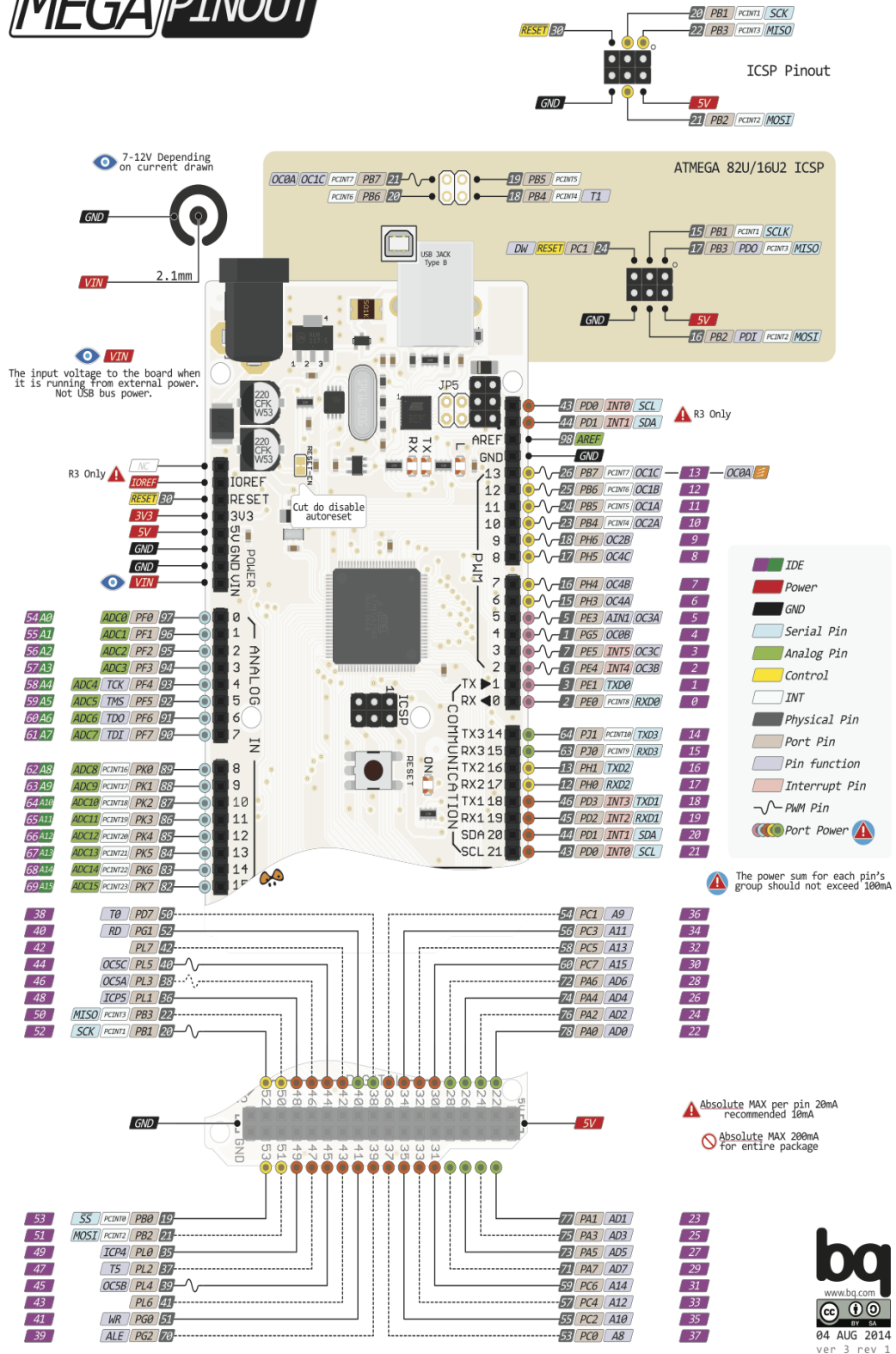


Рисунок 21 – Назначение контактов на плате МК

**AC Light Dimmer Module,
4 Channel, 3.3V/5V logic,
AC 50/60hz, 220V/110V**

- Power
- Control
- GND
- PWM

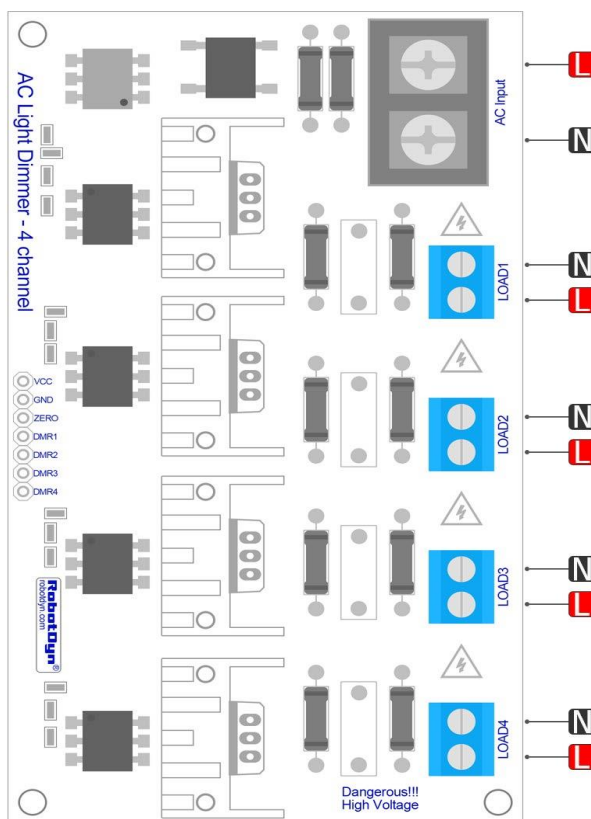
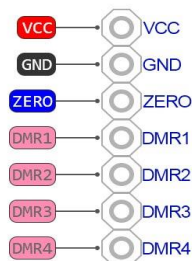


Рисунок 22 – Назначение контактов платы управления нагрузками

Для проверки правильности сборки устройства была произведена сборка в симуляторе FRITZING. На рисунке 23 приведена макетная схема устройства, на которой платы расширения LCD Keypad Shield и RobotDyn AC Light Dimmer 4 channel собраны на макетных платах Breadbord, т.к. готовых модулей в этом симуляторе для выбранных плат расширения платформы

Arduino не оказалось.

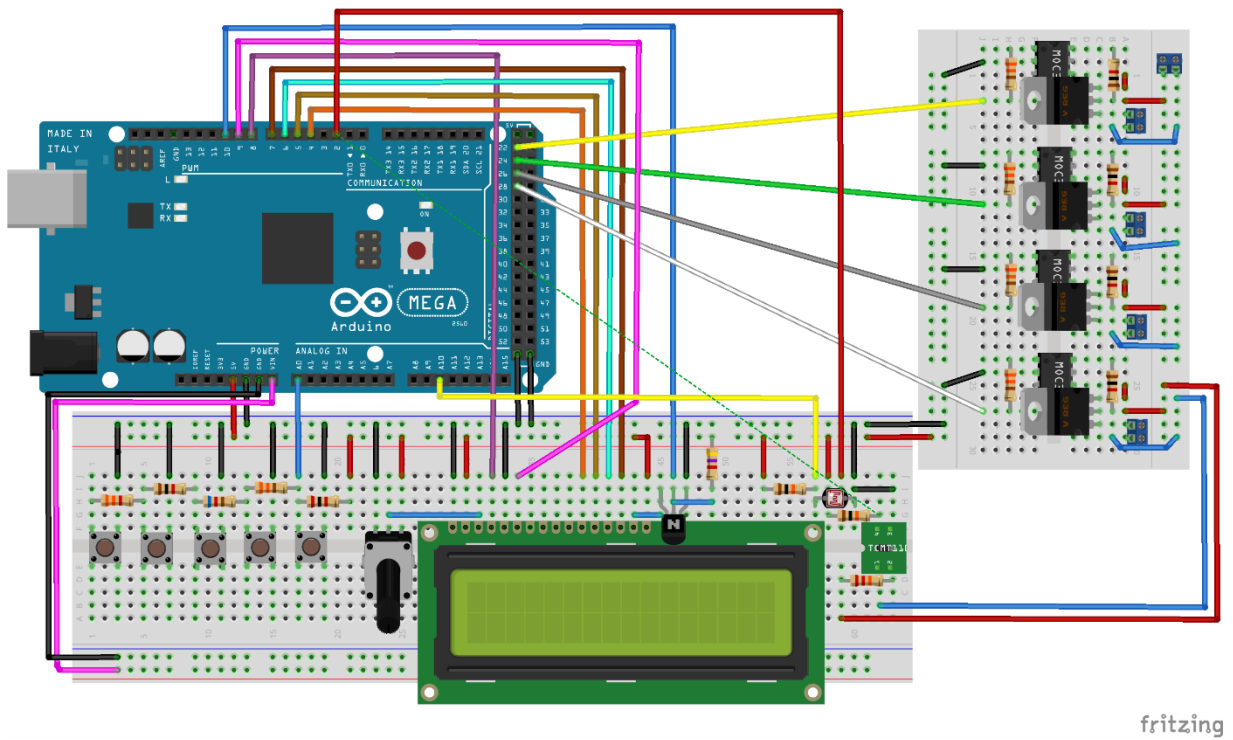


Рисунок 23 – Макетная схема устройства в FRITZING

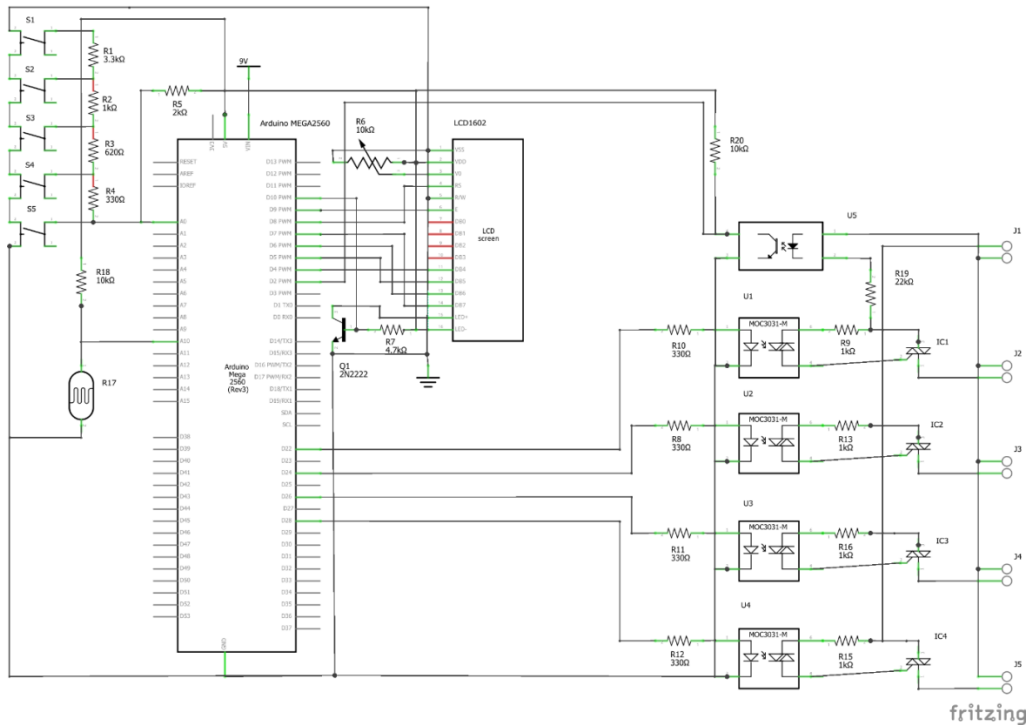


Рисунок 24 – Принципиальная схема устройства в FRITZING

Кроме того, в наборе элементов этого симулятора, также не оказалось симмистора ВТА16-600 и он был создан как новый пользовательский элемент. На основе собранной макетной схемы симулятор создал принципиальную схему, которая приведена на рисунке 24. При составлении макетной и принципиальной схем использовалась справочная информация по назначению выводов плат Arduino. На рисунке 21 приведены данные по назначению контактов на плате МК Arduino MEGA2560. На рисунке 22 приведено назначение контактов платы коммутации нагрузок RobotDyn AC Light Dimmer 4 channel. Назначение контактов LCD Keypad Shield приведено ранее на рисунке 18.

2.4 Разработка конструкции устройства

Для устройства выбран корпус размерами 150x200x50мм. Передняя панель выполнена с вырезами под розетки, общий выключатель питания, индикатор и подпружиненные толкатели кнопок. Применение металлической передней панели с металлическими толкателями кнопок управления уменьшает возможность статического пробоя МК при программировании таймеров. Часть передней панели открывается дверкой для предоставления доступа к USB порту МК для программирования с внешнего персонального компьютера [20].

Для подключения нагрузок применим панельные розетки промышленного типа с крышками (рис.25). Размер розеток 50x50мм, с учетом шарнира крышки 65мм, глубина установки 36мм, высота крышки 15мм. Диаметр отверстия в панели под установку розетки 43мм. Розетки крепятся к панели 4-мя болтами М4 или М5 [18], через 4 отверстия диаметром 5мм, шарнирами крышек к индикатору МК.

Панельная розетка РП10-3

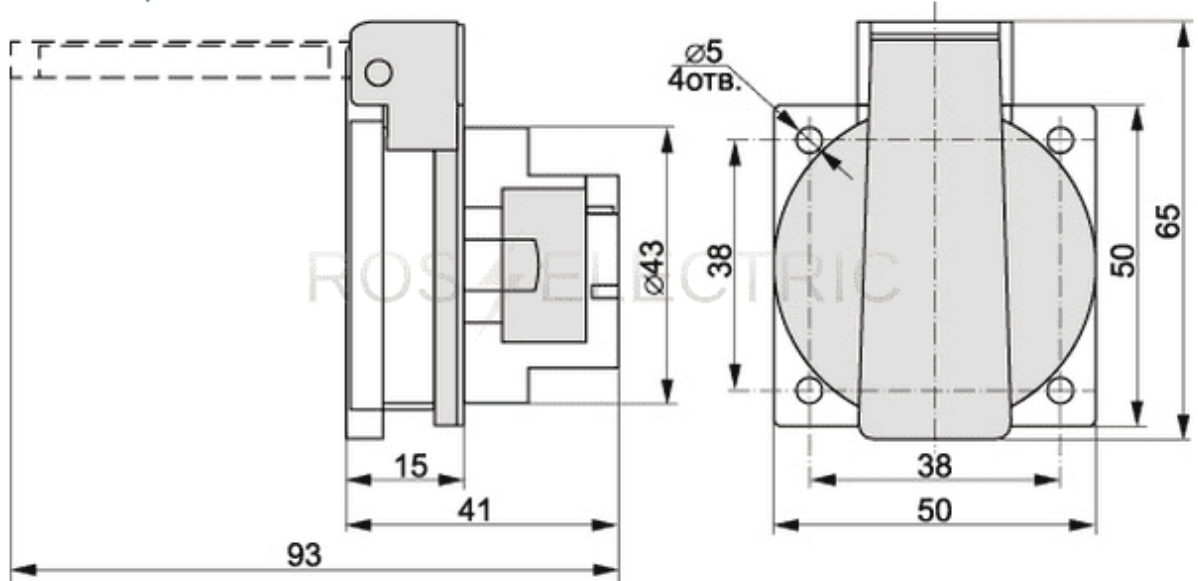


Рисунок 25 – Панельная промышленная розетка 220В/16А

Все платы внутри корпуса располагаются параллельно лицевой поверхности корпуса, причем, платы индикатора и МК крепятся внутри корпуса к передней панели прибора. Плата индикации устанавливается в разъемы платы МК (рис.26).

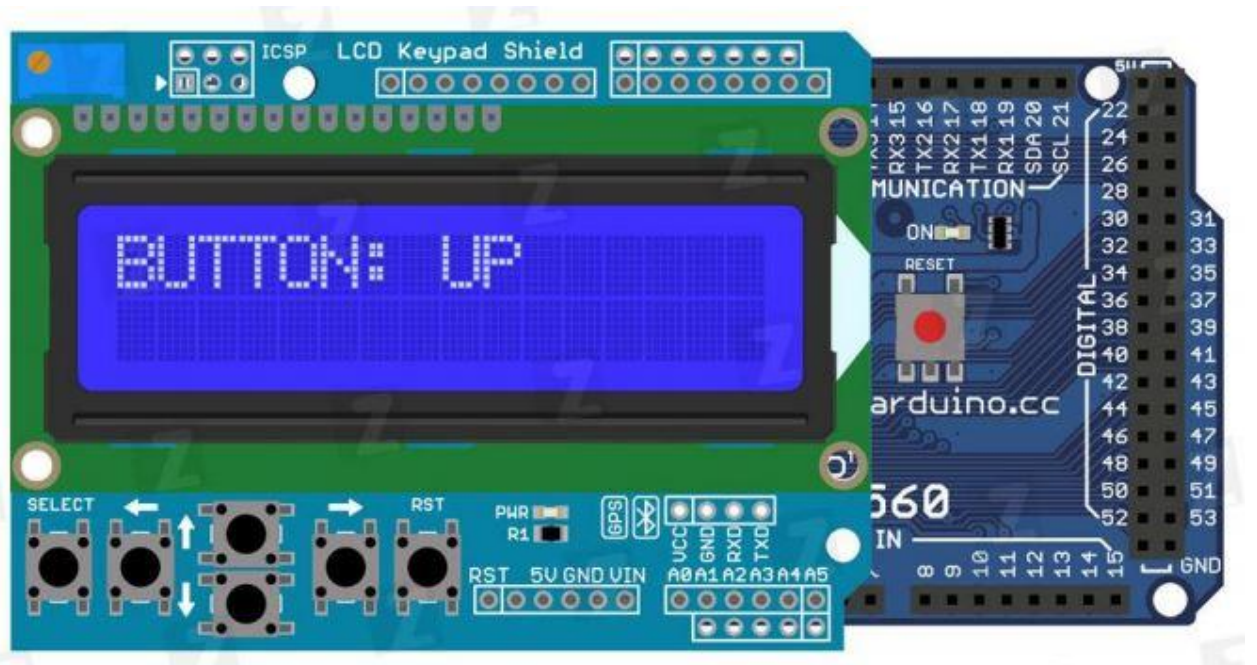


Рисунок 26 – Вид платы индикации подключенной к плате МК

Плата управления нагрузками и блок питания крепятся на задней крышке корпуса устройства. Соединение МК с платой управления нагрузками осуществляется плоским 7-ми проводным кабелем [7].

Питание 9В от блока питания подключается к МК штекером с внешним диаметром контактной части 2,1мм.

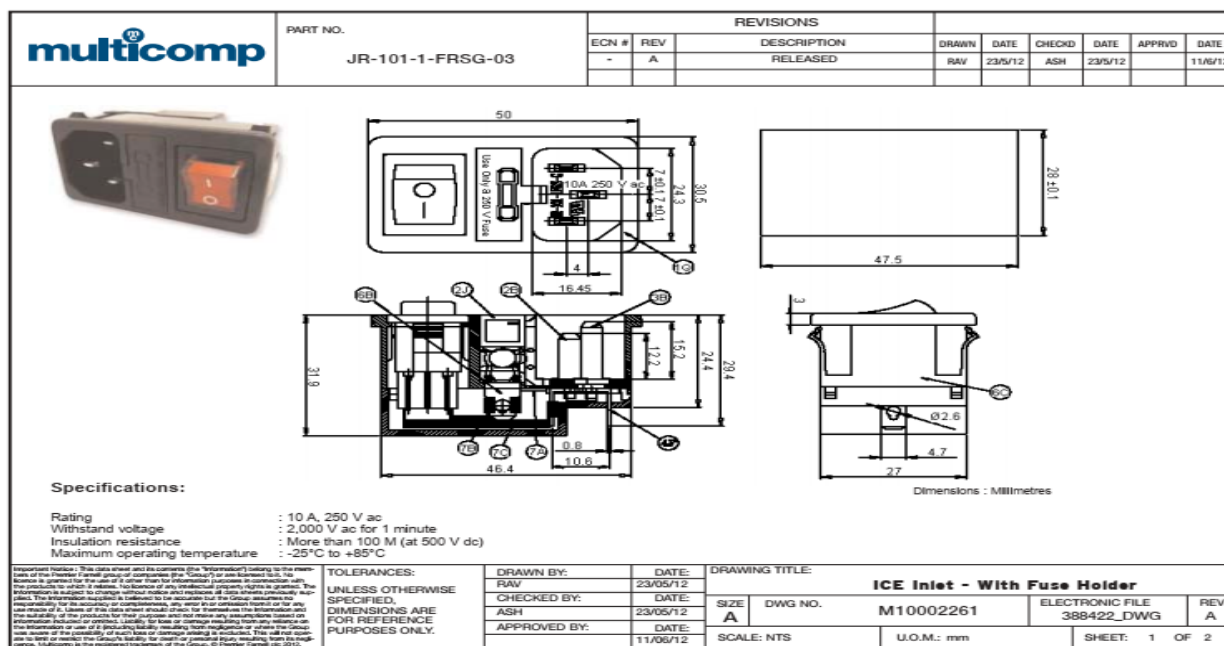


Рисунок 27 – Модуль подключения кабеля питания

Кабель питания электронного таймера подключается через 3-х контактный разъем типа IEC320-C14 с выключателем и колодкой предохранителя модели Multicomp JR-101-1-FRSG-03 (рис. 27). На боковой панели под него делается прямоугольное отверстие размерами 28мм на 48мм.

Датчик освещенности крепится на передней панели под матовым светофильтром винтом М3 и соединяется с платой МК тремя проводами: к контакту аналогового входа МК, к выходам +5В и GND [2].

Размеры основных плат, с отметками крепежных отверстий диаметром 3,2мм под винт М3, приведены на рисунках 28-31. На рисунке 32 изображено взаимное расположение плат в собранном устройстве.

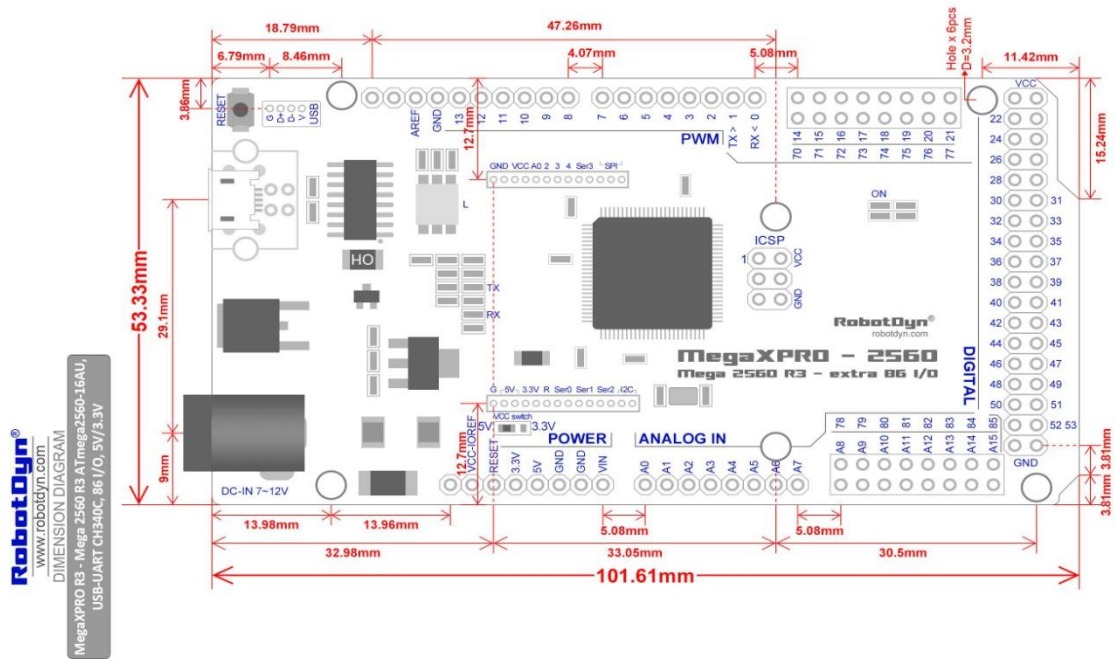


Рисунок 28 – Модуль микроконтроллера

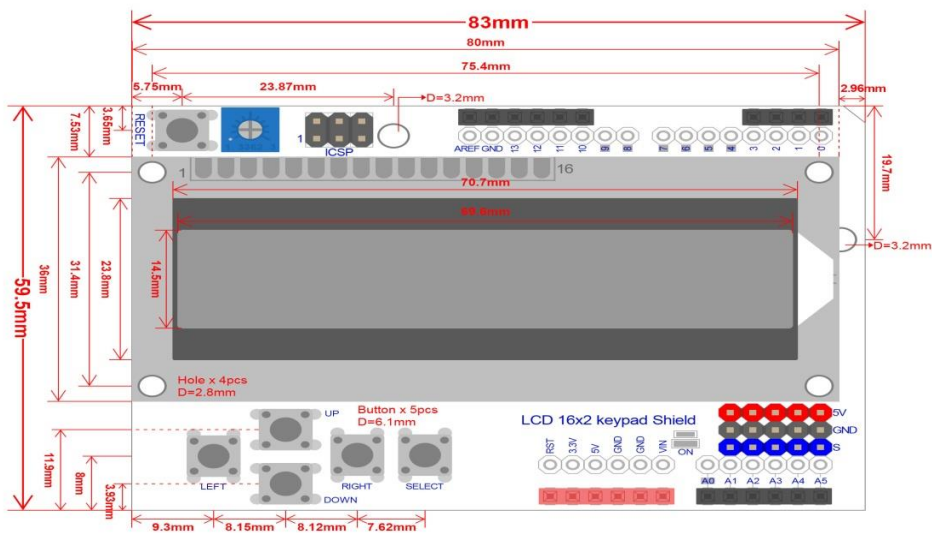


Рисунок 29 – Модуль индикации и управления

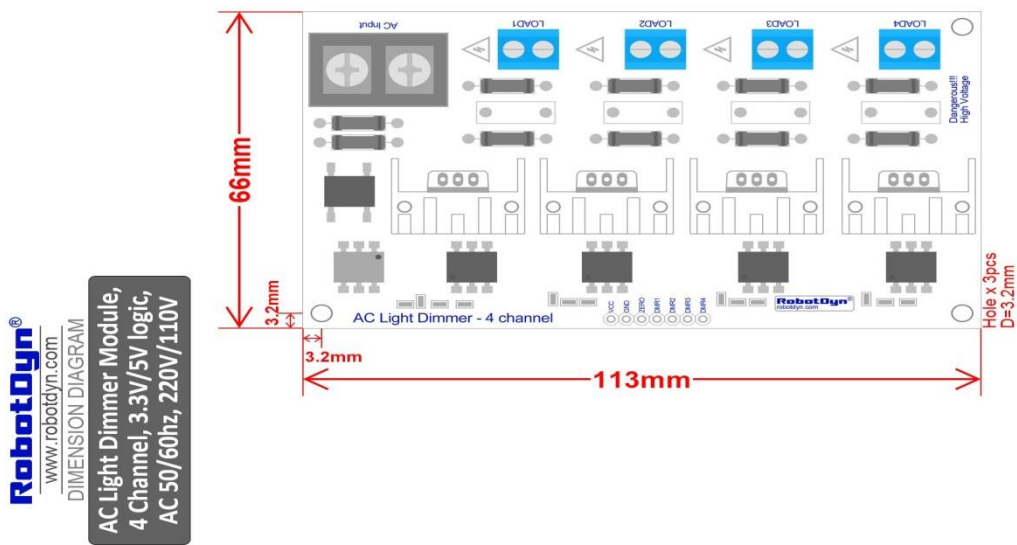


Рисунок 30 – Модуль управления нагрузками

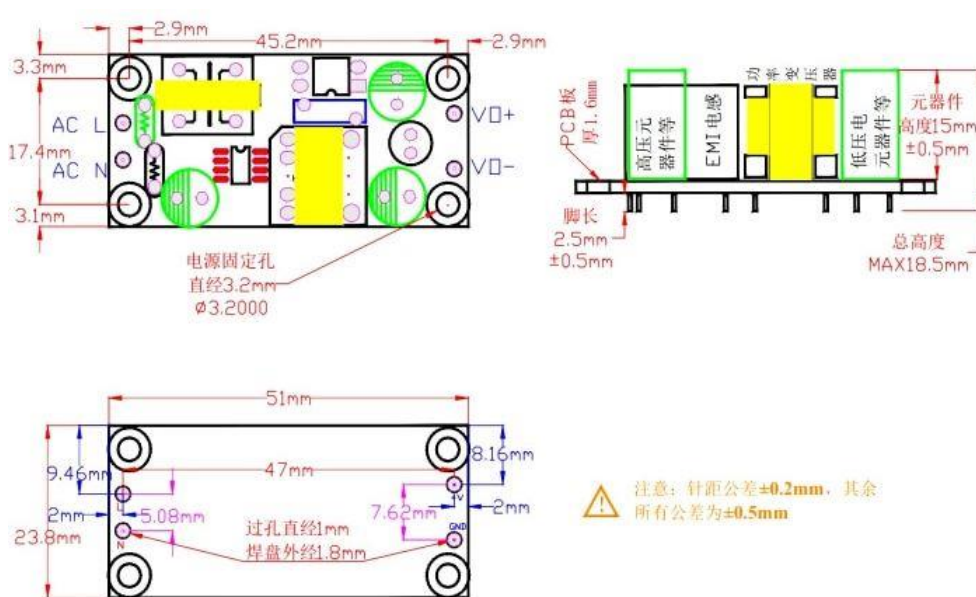


Рисунок 31 – Модуль блока питания

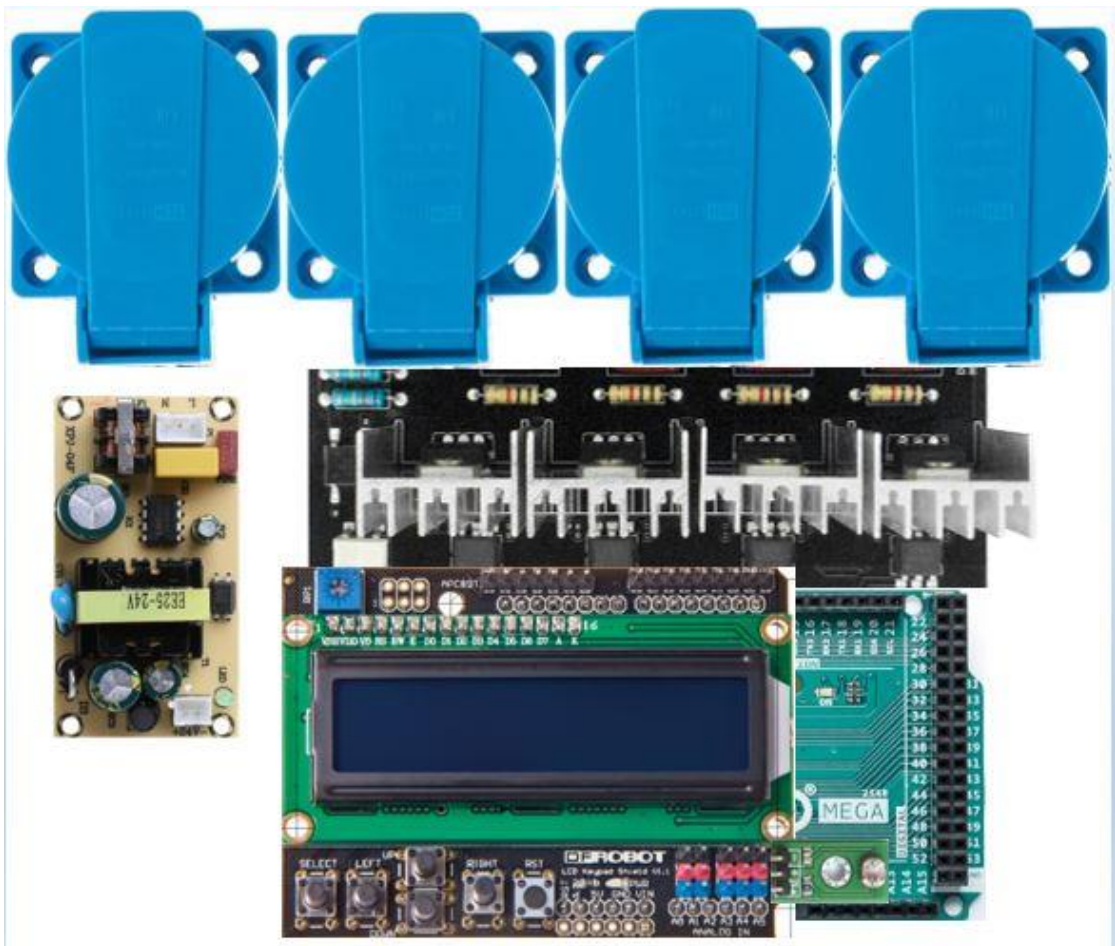


Рисунок 32 – Расположение плат в собранном устройстве

2.5 Разработка алгоритма программы устройства

Функционирование микроконтроллерного устройства для регулирования освещенности помещения обеспечивается программой (прошивкой), написанной на языке программирования C/C++ в среде программирования Arduino IDE и выполняющей следующие функции [3]:

- выбор режима управления светом;
- установка дневного времени включения таймера в пределах суток;
- установка ночного времени выключения таймера в пределах суток;
- установка уровня освещенности для каждого режима (калибровка);
- сохранение настроек устройства в энергонезависимую память;
- автоматическая регулировка освещенности с управлением двумя

типами нагрузок (светильник и умное стекло) в зависимости от времени суток и уровня освещенности на рабочем месте, измеренного датчиком света;

- индикация состояния всех таймеров на одном экране.

Определяем режимы работы освещения:

- кладовая комната с нормируемой освещенностью в 50 лк – «Skład»;
- книгохранилище с нормируемой освещенностью в 75 лк – «Books»;
- спальная комната с нормируемой освещенностью в 100 лк – «Bedroom»;
- жилая комната с нормируемой освещенностью в 150 лк – «Living»;
- учебный кабинет информатики и вычислительной техники с нормируемой освещенностью в 200 лк – «Digital»;
- офисное помещение с нормируемой освещенностью в 300 лк – «Office»;
- лаборантская комната с нормируемой освещенностью в 400 лк – «Techno»;
- учебный кабинет технического черчения и рисования с нормируемой освещенностью в 500 лк – «Drawing» [14].

Для определения времени суток используем таймер микроконтроллера через функцию `millis()`, отсчитывающую миллисекундные интервалы. В сутках 24 часа по 60 минут и 60 секунд в каждом [25]:

$$K_{\text{имп}} = 24 * 60 * 60 * 1000 = 86\,400\,000,$$

что значительно больше 65 565, максимального значения для 16-ти битного таймера. Поэтому, определение времени суток разбиваем на два таймера: минутный

$$K_{\text{имп1}} = 60 * 1000 = 60\,000,$$

и суточный

$$K_{\text{имп2}} = 24 * 60 = 1\,440.$$

Для МК пишется программа (sketch, прошивка), состоящая из:

- программный модуль (функция) формирования минутных импульсов,

в котором задействуется аппаратный 16 битный таймер Atmega 2560, который отсчитывает миллисекунды и, по достижении 60 секунд, записывает соответствующее значение в переменную минут, перезапускает малый цикл программы каждую минуту;

- программный модуль (функция) формирования часовых и суточных импульсов, в котором задействуется аппаратный 16 битный таймер Atmega 2560, который отсчитывает минуты из данных первого таймера и по достижении 60 минут записывает соответствующее значение в переменную часов, по достижении значения 1440 регистрируется суточный интервал и перезапускается суточный цикл программы [18];

- программный модуль (функция) автоматического управления светильником в зависимости от уровня освещенности на рабочем месте, измеренного датчиком света и поданного на МК в аналоговом виде, дает команду управления питанием потолочного светильника в аналоговом виде (ШИМ);

- программный модуль (функция) автоматического управления умным стеклом в зависимости от времени суток - отслеживает переменную «день-ночь» и уровень освещенности на рабочем месте, измеренный датчиком света, а также состояние переменной «защита от посторонних глаз», дает команду управления прозрачностью умного стекла (переключение кнопками UP/DOWN);

- программный модуль установки часов, который записывает текущее время, установленное пользователем, в соответствующие переменные минут и часов, сохраняя их в энергонезависимую память [15];

- программный модуль калибровки датчика света, который приводит показание встроенного датчика освещенности в каждом из режимов работы устройства (переключение кнопками LEFT/RIGHT), к показанию внешнего люксметра с сохранением установленного значения в энергонезависимую память;

- программный модуль (функция) формирования информации на индикаторе МК [13], обрабатывает формирование отображения символов в навигации по меню настройки устройства;

- программный модуль (функция) кнопок панели программирования МК, обрабатывает аналоговый сигнал с панели кнопок и обеспечивает управление навигацией по меню настройки устройства.



Рисунок 33 – основной экран устройства

На основном экране (рис.33) отображаются текущие значения переменных: слева вверху - время, справа вверху – освещенность, слева внизу – режим работы светильников, справа внизу – режим работы умного стекла [15].

Переключение режима работы светильников осуществляется кнопками LEFT/RIGHT. Переключение режима работы умного стекла («затемнение окна») осуществляется кнопками UP/DOWN в основном окне навигации. Выбор пользовательского режима сохраняется в энергонезависимую память.

Для задания новых значений переменных, нужно нажатием на кнопку SELECT выбрать соответствующую переменную для изменения (она будет мигать) и выбрать кнопками LEFT/RIGHT соответствующий знак для изменения и кнопками UP/DOWN изменяем их [2]. Нажатием на кнопку SELECT сохраняем выбранное значение в энергонезависимую память и переключаемся к следующей переменной и по кругу в основной экран (нажать на SELECT несколько раз пока не перестанут мигать переменные).

2.6 Калибровка датчика света устройства

Для правильного функционирования автоматического регулирования света необходимо произвести калибровку датчика освещенности. Для этого располагаем рядом с устройством образцовый люксметр (рис.34), проводим измерение освещенности в соответствии с инструкцией люксметра и через меню настроек устройства заходим в режим калибровки - установки переменной освещенности. Далее кнопками UP/DOWN выставляем показание датчика освещенности такое же как на образцовом люксметре.



Рисунок 34 – Переносной люксметр RGK LM20

Кнопками LEFT/RIGHT переключаем пользовательский режим освещенности, при этом будет переключаться режим работы устройства меняя яркость свечения потолочных светильников, и в каждом из режимов проводим калибровку. Нажатием на кнопку SELECT сохраняем выбранное значение, и переключаемся по кругу в основной экран (нажать на SELECT несколько раз пока не перестанут мигать переменные).

2.7 Составление программы устройства

В среде разработки и отладки программ Arduino IDE создана программа, отвечающая выше приведенным требованиям [22].

```
#include <LiquidCrystal.h>    //подключаем библиотеку дисплея
#include <EEPROM.h> //подключаем библиотеку энергонезависимой памяти
#include <MATH.h> //подключаем библиотеку математических формул

LiquidCrystal lcd(12, 11, 9, 8, 7, 6, 5, 4, 3, 2);
int mins=0, hours=0, days=0, minsL=0, daysL=0, rtcM=0, rtcH=0;
int M = 5, Mx = M, Lu = 300, SG = 100, Sel = 0;
bool b;

String modeL[]={ "", "Sklad ", "Books ", "Bedroom", "Living ", "Digital", "Office
", "Techno ", "Drawing" }; //типы моделей света
int modeLi[]={ 0,50,75,100,150,200,300,400,500 }; // начальные значения
освещенности по типам моделей света
int modeLic[]={ 0,50,75,100,150,200,300,400,500 }; // скорректированные
значения освещенности по типам моделей света

unsigned long startTime; // время

void setup() {
  lcd.begin(16, 2); // инициализируем дисплей: 2 строки по 16 символов
  startTime = millis(); // считываем время, прошедшее с момента запуска
  программы
  readSet(); // считываем ранее сохраненные настройки
  pinMode(44, OUTPUT); // назначение порта 44 на выход PWM управление
  светом
  pinMode(46, OUTPUT); // назначение порта 46 на выход PWM управление
  умным стеклом
}

void loop() {
  rtcGo(); // запуск обработки часов
  Lu=analogRead(59); // чтение текущего значения освещенности с
  фоторезистора, подключенного к выводу A5 (вывод 59 платы Arduino
  MEGA2560)
  xKey(); // запуск обработки нажатий клавиш
```

```

}

void setPlay(){ //выводит на экран мигающие значения переменных
  lcd.setCursor(0, 0); //установить курсор на первую строку и
  первый символ
  int LuY = modeLi[M] - modeLic[M] + Lu; //вычисление отображаемого
  значения освещенности с учетом коррекции
  String tr = ":"; //определить отображение разделителя
  часов и минут
  String rtcHs = String(rtcH); //определить отображение
  переменной часов
  String rtcMs = String(rtcM); //определить отображение
  переменной минут
  String Ls = String(LuY); //определить отображение
  переменной освещенности
  String modeLs = modeL[M]; //определить отображение
  переменной модели света
  String SmartGs = String(SG); //определить отображение
  переменной состояния умного стекла
  if(b){ //определить мигание переменной - стирание
  отображения на текущий проход
    if(M==0) tr = " ";
    if(M==1) rtcHs = " ";
    if(M==2) rtcMs = " ";
    if(M==3) Ls = " ";
    if(M==4) modeLs = " ";
    if(M==5) SmartGs = " ";
  }
  String Firststr = String(" " + rtcHs + ":" + rtcMs + " " + Ls + " lk ");
  //формирование первой строки
  lcd.println(Firststr); //вывод на экран первой
  строки
  String Secondstr = String(" " + modeLs + " SG " + SmartGs + " %");
  //формирование второй строки
  lcd.println(Secondstr); //вывод на экран второй
  строки
  b=!b; //инверсия отображения
  значения переменной отображения
  lcd.clear(); //очистка буфера экрана
}

void rtcGo(){ //функция хода часов рассчитывает текущее время и
  записывает количество часов и минут в переменные часов и минут

```

```

unsigned long tX = 0; //переменная текущего времени
if(millis()-tX>=60000){ //если количество миллисекунд станет больше 60сек,
то
  if(rtcM<60){ //если количество минут еще не 60
    rtcM++; //увеличиваем минуты на единицу
  } else { //а иначе
    rtcM=0; //обнуляем количество минут
    if(rtcH<24){ //если количество часов еще не 24
      rtcH++; //увеличиваем часы на единицу
    } else rtcH=0; //а иначе обнуляем количество часов
    }
  do{ //в цикле защиты от переполнения таймеров
микроконтроллера
    tX+=60000; //увеличиваем переменную времени на 60 секунд
    if(tX<60000) break; //если переменная станет меньше 60 секунд, т.е.
счетчик переполнится, выходим из цикла
  }while(tX<millis()-60000); //до тех пор, пока переменная времени меньше
чем разница между текущим значением функции времени и 60 секундами
  }
}

void xKey(){ // Функция обработки нажатий кнопок
  int uKey; // Создаем переменную uKey
  Mx = M; // Запоминаем текущее значение модели света
  uKey = analogRead(0); // Задаем номер порта с которого
производим считывание
  if (uKey > 10){ // отсекаем шум, если нажата кнопка
    if (uKey < 100) { // Если uKey меньше 100 нажата кнопка
      "Вправо" Right
      if(M<7) M++; else M=0; // Переход к следующему режиму
освещения
    }
    else if (uKey < 200) { // Если uKey меньше 200 нажата кнопка
      "Вверх" Up, то в зависимости от вида представления
      if(Sel==0){if(SG<100)SG+=99;else SG=1;} // прозрачность умного окна 1
= матовое, 99 = прозрачное
      if(Sel==1){if(rtcH<24)rtcH++;else rtcH=0;} // увеличение числа часов 00...23
      if(Sel==2){if(rtcM<24)rtcM++;else rtcM=0;} // увеличение числа минут
00...59
      if(Sel==3){if(Lu<501)Lu++;else Lu=0;} // увеличение освещенности
0...500
    }
  }

```

```

else if (uKey < 400){ // Если uKey меньше 400 нажата кнопка
"Вниз" Down, то в зависимости от вида представления
if(Sel==0){if(SG>1)SG-=99;else SG=1;} // прозрачность умного окна 1 =
матовое, 99 = прозрачное
if(Sel==1){if(rtcH>0)rtcH--;else rtcH=23;} // уменьшение числа часов 00...23
if(Sel==2){if(rtcM>0)rtcM--;else rtcM=59;} // уменьшение числа минут
00...59
if(Sel==3){if(Lu>0)Lu--;else Lu=500;} // уменьшение освещенности
0...500
}
else if (uKey < 600){ // Если uKey меньше 600 нажата кнопка
"Влево" Left
if(M>0) M--; else M=7; // Переход к предыдущему режиму
освещения
}
else if (uKey < 800){ // Если uKey меньше 800 нажата кнопка
"Выбор" Select
if(Sel<3) Sel++; else Sel=0; // Переключение режима отображения
информации при каждом нажатии на кнопку по кругу
saveSet(); // Сохранение введенных данных по нажатию
кнопки SELECT
}
}
setPlay(); // Вывод на экран
setOut(); // Установка уровней выходов
}

void setOut(){ // плавное изменение яркости света
int LuOut = map(modeLi[M],0,500,0,255); // Определение уровня PWM на
выходе уровня света
int LuOutX = map(modeLi[Mx],0,500,0,255);
if(M>Mx){ // плавное нарастание яркости света
for(int i=LuOut;i<=LuOutX;i++) {
analogWrite(44,i);
delay(5); // ставим задержку для эффекта
}
}
else if(Mx>M){ // плавное затухание яркости света
for(int i=LuOutX;i>=LuOut;i--) {
analogWrite(44,i);
delay(5); // ставим задержку для эффекта
}
}
}

```

```

else analogWrite(44,LuOut);      // Посылаем на выход 44 сигнал уровня
света

int SGOut = map(SG,0,100,0,255); // Определение уровня PWM на выводе
управления умным стеклом
analogWrite(46,SGOut);          // Посылаем на выход 44 сигнал умного
стекла
}

void saveSet(){
EEPROM.put(0, Lu);              // Обновление текущего значения яркости
по адресам 0,1 (int занимает 2 байта)
EEPROM.put(3, SG);              // Обновление состояния умного стекла по
адресам 3,4
EEPROM.put(5, rtcH);            // Обновление настройки часов по адресам
5,6
EEPROM.put(7, rtcM);            // Обновление настройки минут по адресам
7,8
EEPROM.put(9, M);               // Обновление текущего режима света по
адресам 10,11
for(int e=12;e<30;e+=2){        // Обновление скорректированных значений
яркости по адресам с 12 по 28
int LuX = modeLic[e];
EEPROM.put(e, LuX);
}
}

void readSet(){
EEPROM.get(0, Lu);              // Чтение текущего значения яркости по
адресам 0,1
EEPROM.get(3, SG);              // Чтение состояния умного стекла по
адресам 3,4
EEPROM.get(5, rtcH);            // Чтение настройки часов по адресам 5,6
EEPROM.get(7, rtcM);            // Чтение настройки минут по адресам 7,8
EEPROM.get(9, M);               // Чтение текущего режима света по
адресам 10,11
for(int e=5;e<30;e+=2){        // Чтение скорректированных значений
яркости по адресам с 12 по 28
int LuX;
EEPROM.get(e, LuX);
modeLic[e] = LuX;
}
}

```

Программа откомпилирована в IDE Arduino ver.2.0, т.е. не содержит синтаксических ошибок (рис.35).

```
17 void setup() {
18   lcd.begin(16, 2); // инициализируем дисплей: 2 строки по 16 символов
19   startTime = millis(); // считываем время, прошедшее с момента запуска программы
20   readSet(); // считываем ранее сохраненные настройки

```

Output

```
Sketch uses 7198 bytes (2%) of program storage space. Maximum is 253952 bytes.
Global variables use 248 bytes (3%) of dynamic memory, leaving 7944 bytes for local variables. Maximum is 8192 bytes.
-----
Compilation complete.
```

Рисунок 35 – Отчет о результате компиляции

Далее составленную программу закидываем в МК Arduino в симуляторе PROTEUS и проверяем ее работу на виртуальной модели разработанного устройства.

3 Оценочная часть

Проанализировав всю основную часть можно составить расчет всего необходимого оборудования, которое приведено в таблице 3 ниже.

Таблица 3 – расчёт себестоимости устройства

№ п/п	Наименование	Кол-во, шт.	Стоимость, руб.
1	RobotDyn Arduino MEGA2560	1	1360
2	RobotDyn Keypad LCD Shield	1	333
3	RobotDyn AC Light Dimmer 4 channel	1	422
4	Coreset SM-PLG06A-09	1	210
5	Multicomp JR-101-1-FRSG-03	1	152
6	Z-light sensor	1	48
7	Розетка TDM РП10-3 панельная скрытая с крышкой IP44 16А 2Р+РЕ 220В	4	59
8	Шлейф проводов 28AWG для Arduino с разъемами DuPont «Папа-мама»	1	160
9	Кабель питания с штекером DC5.5*2,1 для Arduino	1	14
10	Стойка крепления для платы М3х10мм	11	6
11	Стойка крепления для платы М3х20мм	4	7
12	Винт М3*6	30	3
13	Винт М4*20	18	5
14	Шайба d3	18	3
15	Гайка М3	18	4
16	Светофильтр фотодатчика	1	52
17	Корпус прибора металлический GAINTA G2119 (200x150x55)	1	1040
18	Кабель питания прибора	1	149
19	Инструкция по эксплуатации	1	56
20	Упаковка из картона (210x160x100)	1	99
21	Smart glass (2000x2000) с трансформаторным блоком питания	1	50000
22	Нормо-час монтажника	12	200
Сумма			57121

Хотелось бы отметить, что основные затраты идут на умные стекла, рациональней было бы использовать автоматические жалюзи с несколькими слоями пропускания света, но умные стекла менее громоздки и не требовательны к дальнейшему уходу за ними. В будущем, когда smart glass активно войдет в обиход и технологии позволят эффективней производить данный продукт, стоимость на устройство контроля освещенности значительно понизится.

Заключение

В данной выпускной квалификационной работе было разработано микроконтроллерное устройство для автоматического контроля уровня освещенности для жилых и офисных помещений.

Главная задача устройства поддерживать уровень освещённости в помещении в диапазоне от 50 лк до 500 лк.

Устройство представляет собой сборку из:

1 умного стекла встроенного в стеклопакет окна и контролируемого микроконтроллером;

2 настраиваемых ламп освещения для изменения уровня освещенности помещения;

3 контроллера регулирования процесса освещенности и способа выбора необходимого уровня.

Вся установка рассчитана на комнату 3х6х3м, где умнее стекло занимает 70% поверхности от стены 3х3м.

В ходе ВКР были произведены:

-Изучение аналогичных устройств на рынке и области их применения.
Рассмотрение актуальности данной тематики.

-Подбор комплектующих. Расчёт количества светильников и их расположение в комнате.

-Разработка электрической схемы. Сборка .

-Составление алгоритма, написание программы.

-Расчёт себестоимости устройства.

В заключении хочу отметить перспективность данной тематики. Уже сейчас идёт бурное развитие системы умного дома, так что в ближайшем будущем контроль уровня освещённости будет интегрирована в систему умного дома.

Список используемой литературы

1. Аливерти, П. Электроника для начинающих / П. Аливерти [пер. с ит. И.В.Потрясиловой] – М.: Эксмо, 2018. – 368с.
2. Бейктал, Дж. Конструируем роботов на Arduino. Первые шаги [Электронный ресурс] / Дж. Бейктал [пер. с англ. О. А. Трефиловой]. — Эл. изд. — Электрон. текстовые дан. (1 файл pdf : 323 с.). — М.: Лаборатория знаний, 2016.
3. Белов, А.В. Arduino: от азов программирования до создания практических устройств / А.В. Белов, - СПб.: Наука и Техника, 2018. – 480с.
4. Бокселл, Дж. Изучаем Arduino. 65 проектов своими руками. / Дж. Бокселл, [пер. с англ.]. — СПб.: Питер, 2017. — 400 с.
5. Быстрый старт. Первые шаги по освоению Arduino. Набор конструктор начинающего изобретателя MaxKit.Ru / Creative commons, San Francisco, USA, 2020 – 80с.
6. Геддес, М. 25 крутых проектов с Arduino / Марк Геддес [пер. с англ. М.А. Райтмана]. — Москва: Эксмо, 2019. — 272 с.
7. ГОСТ 24940-2016- Методы измерения освещенности
8. Жимарши, Ф. Сборка и программирование мобильных роботов в домашних условиях / Ф. Жимарши [перев. с фр. М.А.Комаров] – М.: ИТ Пресс, 2007.- 288с.
9. Колмаков, С. Дело в программировании. Пособие / С. Колмаков, - Курган, 2017. – 101с.
10. Макаров, С.Л. Arduino Uno и Raspberry Pi 3: от схемотехники к интернету вещей. - М.: ДМКПресс, 2018. - 204 с.
11. Мамичев, Д. Программирование на Ардуино. От простого к сложному / Д.Мамичев, — М.: СОЛОН-Пресс, - 2018. - 244 с.
12. Момот, М.В. Мобильные роботы на базе Arduino. / М.В. Момот, — СПб.: БХВ-Петербург, 2017. — 288 с.
13. Монк, С. Электроника. Сборник рецептов: готовые решения на базе Arduino и Raspberry Pi / С. Монк [Пер.с англ.] - СПб.: ООО "Диалектика": 2019. - 480с.
14. Петин, В.А. Arduino и Raspberry Pi в проектах Internet of Things / В.А.Петин

- СПб. БХВ-Петербург, 2019. – 432с.
15. Петин, В.А. Практическая энциклопедия Arduino / В.А. Петин, А.А. Биняклевский, - М.: ДМК Пресс, 2017 – 152с.
 16. Петин, В.А. Проекты с использованием контроллера Arduino / В.А. Петин, - СПб.: БХВ-Петербург, 2019. -496с.
 17. Платт, Ч. Электроника для начинающих / Ч. Платт, [Перев. с англ. М.Райтмана] – СПб.: БХВ-Петербург, 2017. – 416с.
 18. Редкар М. Создание роботов вместе с Arduino / М. Редкар – 2020. 24с.
 19. Хуанг, Б. Arduino для изобретателей на 10 занимательных проектах / Б.Хуанг, Д.Ранберг [пер. с англ.]. — СПб.: БХВ-Петербург, 2019. — 288 с.
 20. Юревич, Е.И. Основы робототехники: учебное пособие / Е.И. Юревич, - СПб.: БХВ-Петербург, 2010. – 368с.
 21. Яценков, В.С. От Arduino до Omega: платформы для мейкеров шаг за шагом / В.С. Яценков, - СПб.: БХВ-Петербург, 2018 – 304с.
 22. Evans, B.W. Arduino Programming Notebook / B.W.Evans, - Creative Commons: San Francisco, 2007. – 40p.
 23. Margolis, M. Arduino Cookbook, Second Edition / Michael Margolis - O'Reilly Media, Inc., 2012. – 724p.
 24. Schmidt, M. Arduino: A Quick-Start Guide / M. Schmidt - The Pragmatic Bookshelf – Dallas, USA: 2011. – 284p.
 25. Schwartz M. Интернет вещей с Arduino Yún / M. Schwartz – 100с.
 26. Wiley, J. Exploring Arduino: Tools and Techniques for Engineering Wizardry / J. Wiley, - Indianapolis, Canada, 2020 – 491p.