



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ–СОФИЯ



ФАКУЛТЕТ ПО ЕЛЕКТРОННА ТЕХНИКА И ТЕХНОЛОГИИ

ДИПЛОМНА РАБОТА

ТЕМА: Интелигентна система за активно осветление

ДИПЛОМАНТ: Диан Милчев Илиев

РЪКОВОДИТЕЛ: Доц. Емил Димитров

София, 2013

Съдържание

1	Увод	4
2	Литературен обзор	5
3	Избор на метод и блокова схема.....	7
3.1	Избор на метод	7
3.2	Блокова схема	9
4	Оразмеряване на системата	11
4.1	Проектиране на блоковете от блоковата схема.....	11
4.1.1	Сериен интерфейс:.....	11
4.1.2	Инфрачервен port за дистанционно управление	12
4.1.3	Стартираща схема:.....	13
4.1.4	Сензори за температура:	13
4.1.5	Часовник.....	15
4.1.6	Памет за данни.....	16
4.1.7	Интерфейс за светодиодите:.....	16
4.1.8	Контролери на светодиодите:.....	18
4.1.9	Клавиатура	19
4.1.10	Сензори за осветеност и напрежение:.....	20
4.1.11	Управление на мощността.....	22
4.1.12	Сензор за движение с PIR датчик.....	22
4.1.13	Захранване.....	24
4.1.14	Бутони	24
4.2	Интеграционни параметри	26
5	Проектиране на програмното осигуряване	27
5.1	Блок-алгоритъм на програмата	27
5.2	Програмен алгоритъм.....	28
5.2.1	Сериен интерфейс за комуникация с персонален компютър (UART).....	28
5.2.2	Вътрешно модулен интерфейс (I^2C).....	29
5.2.3	Таймерна система (Timer0_A3)	32
5.3	Инструкции за програмиста	38
5.4	Инструкции за потребителя	39
5.4.1	Настройки на серийният port:	39
5.4.2	Потребителски команди за работа със системата през хипертерминал: ..	39
6	Конструктивна част	40
6.1	Принципна схема	40
6.2	Печатна платка	41
6.2.1	Разположение на компонентите:	41
6.2.2	Горен слой:.....	42
6.2.3	Долен слой:	43
6.3	Спецификация на елементите	43
6.4	Модел на осветително тяло	44
6.4.1	Модел	44
6.4.2	Проектиране на осветителното тяло	44
6.4.3	Печатна платка.....	46
7	Заключение - Анотация.....	47
7.1	Насоки за бъдещо развитие:.....	47
8	Използвана Литература.....	49
9	Приложения.....	50

Списък на фигурите

фигура 1 - Блокова схема на устройството	9
фигура 2 - Сериен интерфейс	11
фигура 3 - Инфрачервен Порт	12
фигура 4 - Стартерираща схема	13
фигура 5 - Термосензори	14
фигура 6- Часовник	15
фигура 7 - Памет за данни	16
фигура 8 - Светодиоден интерфейс	17
фигура 9 - Светодиоден контролер	18
фигура 10 - Клавиатура.....	20
фигура 11 - Сензори за осветеност.....	21
фигура 12 - Схема на свързване на сензора за осветеност	21
фигура 13 - Управление на мощността	22
фигура 14 - PIR Сензори.....	23
фигура 15 - Захранващ блок	24
фигура 16 - Бутони.....	25
фигура 17 - Блок-алгоритъм на програмата.....	27
фигура 18 - Разположение на елементите.....	41
фигура 19 - Горен слой	42
фигура 20 - Долен слой.....	43
фигура 21 - Модел на осветително тяло.....	46

Списък на Таблиците

Таблица 1- Хардуерни интеграционни параметри	26
Таблица 2 - Системни тактови сигнали	26

1 Увод

Светлината има много важна роля и е от фундаментално значение за биологичния цикъл на по-голямата част от растителните и животинските видове. Особено голямо е влиянието и върху човека, тъй като зренietо е едно от най-добре развитите ни сетива. Редица изследвания за влиянието на светлината върху човека показват тази чувствителност. Излагането на естествена светлина значително подобряват съня и го правят по-здрав и по-продължителен. Освен на съня, подходящото количество светлина може да помогне за подобряването на настроението, когнитивните функции, повишаване производителността, предизвикването на определени психологически ефекти и др. От друга страна лишаването от светлина за продължително време може да доведе до депресия, рязкото преминаване от слабо към силно осветление може да доведе до дискомфорт и дори главоболие. Всички тези и много други фактори се взимат под внимание когато е необходимо изграждането на осветлението в домовете, офисите и улиците ни. Използването на осветление за подобряване различни желани ефекти се използва все по-масово във всички сфери на човешката дейност.

Осветяването по своята същност е преднамерено използване на светлина с цел постигане на определени практически или естетичен ефект. То включва използването както на изкуствени (като лампи и прожектори) така и на естествени източници посредством подходящо улавяне и насочване на слънчевата и лунната светлина. Дневната светлина понякога се използва и като основен източник на осветление в сградите през деня, като за целта се използват прозорци, капандури, оберлихтове и т.н. Този похват може да спести енергия вместо използването на изкуствена светлина, която представлява и основен компонент в енергоразхода на сградите.

2 Литературен обзор

Разнообразието на съвременните системи за осветление е изключително голямо. Разделянето им на групи може да стане по множество критерии:

Вид на източника на светлина - лампи с нажежаема жичка; луминесцентни тела; светодиоди;

Управляемост - Този критерий дефинира възможност за управление на интензитета на светене и разделя осветителните тела на две големи групи - управляеми и неуправляеми

Енергоефективност

Друг важен аспект на системите за осветление са контролните механизми.

По своята същност те се свеждат до две групи - ръчно и автоматично управление.

Поради нарастващата необходимост от енергия в световен мащаб, все повече се налага изискването за максимална енергоефективност на осветителните системи. Това налага разработването на нови автоматизирани системи за осветление. Обществените места все по-често се съражават със светодиодни осветителни тела захранени с помощта на фотоволтаици и управлявани от прости сензори за движение (най-често P.I.R. датчици). Това безспорно значително намалява енергоразхода за обществено осветление, но поради харектера и простотата на сензорите тези системи са неподходящи за използване в домашни условия. Използването на автоматизирана система за осветление в дома би изисквала интелигентен контрол. Системата трябва да е в състояние да разпознае основни типове активност на хората в стаята за да може да прецени колко светлина е необходима. Така например ако човека е заспал системата може да изключи осветлението. Ако работи на работното си място - да му осигури максимално осветление. Ако гледа телевизия или си почива на дивана - да намали осветеността за да не предизвика

дискомфорт. Когато става въпрос за интелигентност на системите неизменно въвеждаме и понятието изкуствен интелект.

Изкуственият интелект е наука за концепциите, които позволяват на компютрите да правят неща, които за хората изглеждат разумни. Изкуственият интелект има способност да анализира окръжаващата го среда и да предприема действия, които увеличават възможността за постигане на определени цели.

Има две основни концепции за реализиране на интелигентни системи: Генетични алгоритми (невронни мрежи) и предефинирани алгоритми.

Невронната мрежа е модел за обработка на информация, вдъхновен от изучаването на биоелектричните мрежи в мозъка на человека и животните, образувани от неврони и техните синапси.

Най-ценното при използването на този метод е възможността за самообучение на системата - самопрограмиране в зависимост от поведението и специфичните особености на человека който я използва.

Но като недостатък трябва да се отбележи сложността на този тип алгоритми и необходимостта от извършването на множество сложни математически обработки (висока изчислителна мощност на изпълняващия процесор).

Предефинираните алгоритми са модели на обработка на информация, при която се избира една от няколко предварително зададени възможности с помощта на прости математически операции.

Сами по себе си тези алгоритми са относително лесни за реализиране и не изискват сложни изчисления от изпълняващия процесор. Но предлагат ограничен набор от възможности.

3 Избор на метод и блокова схема

3.1 Избор на метод

Най-важния компонент в системите за осветление е източникът на светлина. Неговите характеристики до голяма степен определят изискванията към самата система и дефинират обхвата на възможностите ѝ. Изборът на източника на светлина в съвременните системи за осветление се диктува от три основни параметъра: Енергоефективност, време на живот, и управляемост.

Когато става въпрос за осветление на интериорни помещения съвременните решения са най-вече в областта на луменисцентните лампи за бяла светлина. Техния живот е относително дълъг (около 8000 часа), инсталацията им върху съществуващите инфраструктури е лесна, и макар цената им да е по-висока от тази на източниците с нажежаема жичка тя е по-ниска от тази на светодиодите. Тези лампи обаче имат и някой недостатъци недостатък - те са практически неуправляеми, стартират се относително бавно, а цената на цветните тела е значително по-висока от тази на светодиодите.

От своя страна светодиодните осветителни системи все по-бързо набират популярност и стопяват разликата в цената спрямо луминесцентните лампи. Мощните светодиоди са управляеми, времето на живот е в порядъка на няколко десетки хиляди часа, като излъчването им не е съпроводено с производство на вредни лъчения.

Поради тези причини за текущата система се избират източници на светлина от светодиоден тип. Светодиодният тип дава възможност да се използват схеми за управление с ниско напрежение които са лесно интегрируеми в микропроцесорни системи.

От своя страна микропроцесорната система ще осигурила голяма гъвкавост във възможностите за управление и контрол на източниците на

светлина, което е зададено като задължително условие. Тя ще осигури и нивото на интелигентност на системата.

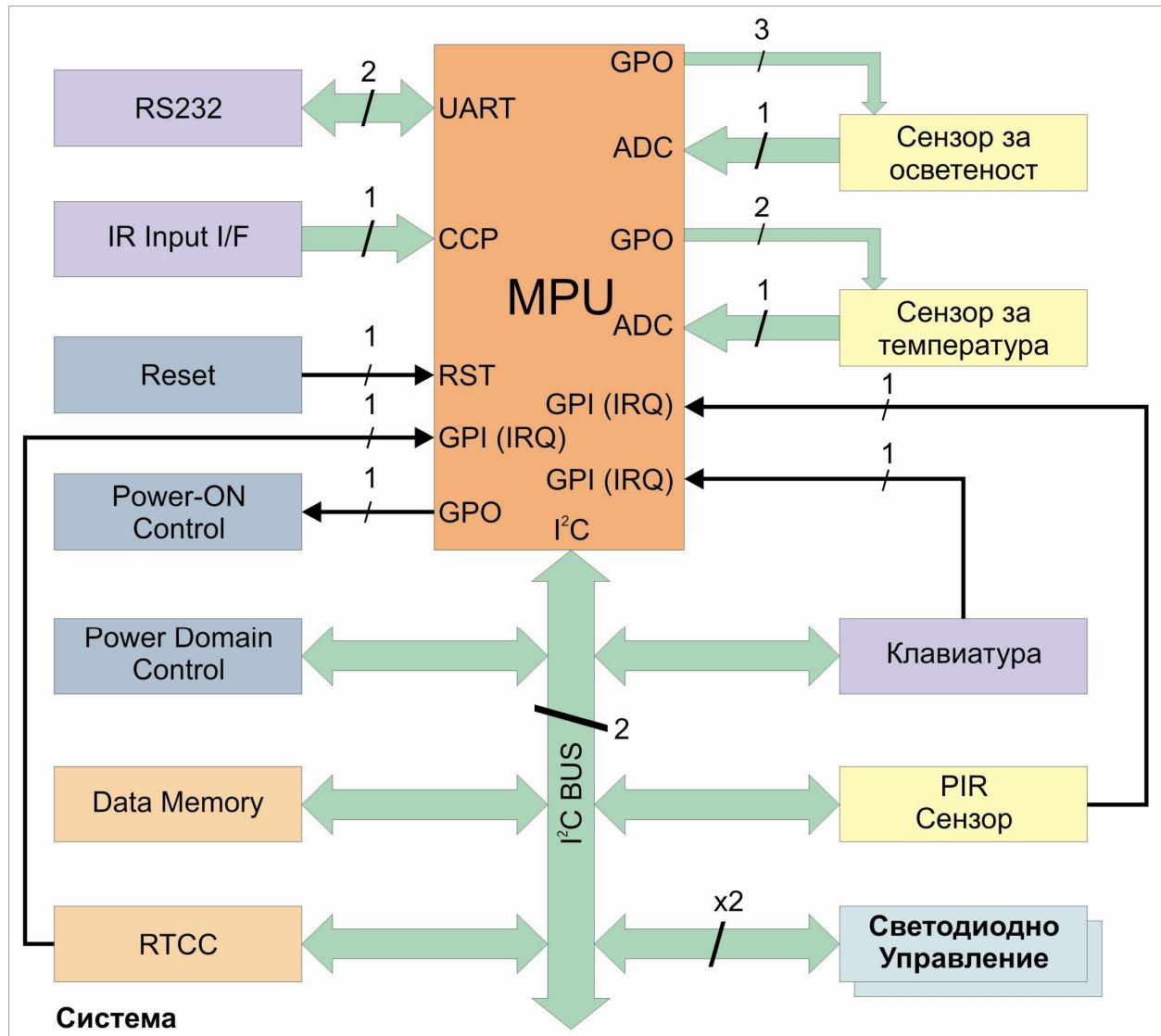
За да се намалят изискванията към изчислителната мощност на микропроцесора, нивото на интелигентност на системата ще бъде изградено на базата на предефинирани алгоритми.

Входните данни ще бъдат осигурени чрез набор от сензори и помощни устройства.

Допълнителни схеми за управление на мощността ще осигурят минималната консумация на системата в изключено състояние.

3.2 Блокова схема

На фигура 1 е показана блоковата схема на устройството.



фигура 1 - Блокова схема на устройството

Системата е изградена от микроконтролер и четири основни групи периферни устройства:

Сензори (в жълто) - осигуряват входния поток от данни на базата на които ще се определят режимите. Дефинирани като задължителни това са сензорите за движение, осветеност и температура.

Управляващите интерфейси (в лилаво) - това са основните начини за управление и настройка на системата. Предвидени са задължителните

интерфейси за дистанционно управление и сериен интерфейс, а като допълнителна опция е прибавена и клавиатура.

Изпълнителни елементи (в синьозелено) - това са основните изпълнителни елементи с помощта на които микроконтролерът ще управлява светодиодните лампи. Те ще се състоят от две основни части - интерфейс (I^2C) за комуникация с микроконтролера и постояннотокови преобразуватели контролиращи тока през светодиодите.

Референтни компоненти (в кремаво) - външна периферия за разширяване обхвата и възможностите на процесора. Състои се от външна памет за данни, която ще съхранява различни настройки, и часовник за реално време, който ще дава възможност на системата да определя коя част на денонаощието е.

Системни компоненти (в сиво) - това са групи компоненти осигуряващи функционирането на системата и контрола върху мощността.

4 Оразмеряване на системата

4.1 Проектиране на блоковете от блоковата схема

4.1.1 Сериен интерфейс:

Серийният интерфейс е заложен като задължителен компонент по който да се осъществи комуникацията с персонален компютър.

Тъй като става въпрос за комуникация с персонален компютър, в случая биха били подходящи USB и RS232 интерфейсите.

Поради спецификата на системата при избора на интерфейс трябва да се вземат в предвид два основни фактора:

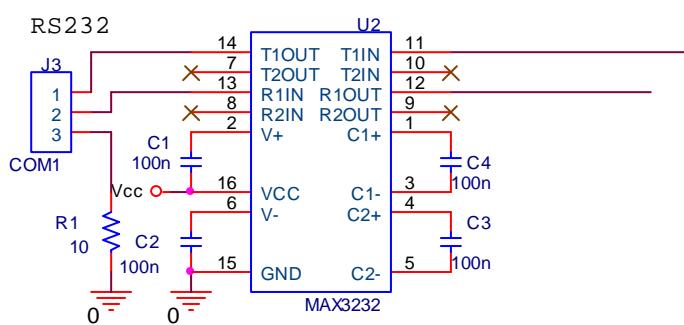
Количество предавани данни

Разстояние за предаване на данните

В случая става въпрос за предаване на малко количество данни към стационарна система. По този критерий и двата интерфейса биха били работоспособни.

Специфична особеност на системата е разстоянието на което тя може да бъде монтирана от управляващия компютър. Тъй като то може да бъде значително (над 5 метра), се налага използването на RS232 интерфейса.

На фигура 2 е показана принципната схема на използвания сериен интерфейс.



фигура 2 - Сериен интерфейс

Изграден е с помощта на CMOS/RS232 логически преобразувател MAX3232.

Изборът на конкретният преобразувател е съобразен с логическите нива на използвания микроконтролер (MSP430G2553) и използваното захранващо напрежение (3,3V).

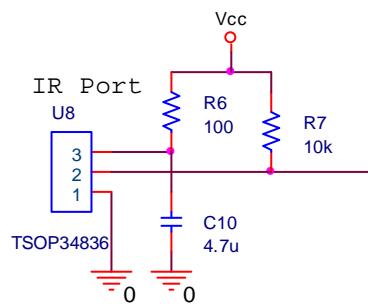
Периферните компоненти са избрани и поставени според препоръките на фирмата производител на чипа.

Добавен е и баластен резистор на масата до конектора (R1), който служи като защитен буфер при поява на потенциална разлика в масите на приемника и предавателя. Такова напрежение може да се породи в следствие на индуциране на паразитни токове от околната среда, тъй като свързващата линия може да бъде дълга.

4.1.2 Инфрачервен port за дистанционно управление

Като задължителен компонент е заложено и дистанционното управление посредством инфрачервен port.

На фигура 3 е показана схемата на свързване на инфрачервения приемник.



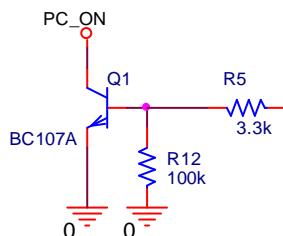
фигура 3 - Инфрачервен Порт

Схемата е реализирана с помощта на интегрална схема TSOP34836. Последната съдържа в себе си всички необходими преобразуватели на входния сигнал, така че в изхода и да се получи разпознаваем сигнал с

CMOS логика. Външно на схемата са включени RC филтър за подтискане на смущенията от захранващата линия изградена според препоръките на производителя.

4.1.3 Стартсираща схема:

На фигура 4 е показана схемата за автоматично включване и изключване на захранването от микроконтролера.



фигура 4 - Стартсираща схема

По същество схемата представлява биполярен транзистор BC107A използван като изход с отворен колектор свързан към микроконтролера. Този тип схема позволява управление на разнообразен тип захранвания - може да се използва за управление на реле, за управление на оптрон или като електронен ключ при използване на по-сложни захранващи блокове.

При всички случаи предназначението на схемата е да осигури контрол на микропроцесора върху мощността. Тази опция би му позволила да сведе до минимум консумацията на енергия в изключено състояние.

Базовият резистор (R5) е избран така, че да ограничи максималният базов ток до 1mA.

Максимален колекторен ток - 100mA

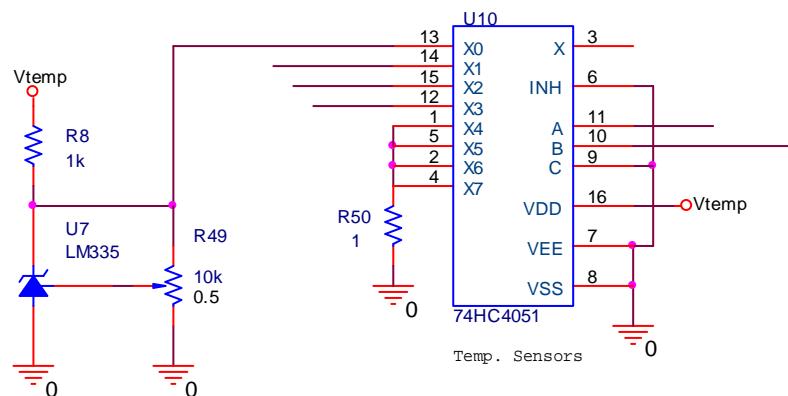
Максимално напрежение колектор-емитер - 50V

За повече информация вижте спецификацията на транзистора.

4.1.4 Сензори за температура:

Термосензорите също са задължителен компонент за системата.

На фигура 5 е показана схемата на свързване на термосензорите към системата.



фигура 5 - Термосензори

За отчитането на информацията от термосензорите е предвиден вход към микроконтролера свързан с АЦП-то му. Предвиждат се входове за 4 термосензора - един локално монтиран на платката за да отчита температурата в кутията на устройството, 1 за отчитане на температурата в стаята и с възможност за свързване на още 2 външни сензора - могат да се използват за следене на температурата на светодиодите в осветителните тела или за триангулиране на стайната температура в зависимост от монтажа им.

Корпуса на микроконтролерите е с ограничен набор от изводи със специализирана функции - в случая порт 1 осигурява мултиплексиран достъп до по-голямата част от периферните устройства на микроконтролера, включително АЦП, UART, I²C и др. Това налага външно мултиплексиране на сигналите от термосензорите. Използвайки аналогияния мултиплексор 74HC4051 се осигурява възможност всички 4 термосензора да бъдат свързани към един единствен извод със специализирана функция на микроконтролера. Управлението на мултиплексора става посредством изводи с общо предназначение от порт 2 на микроконтролера.

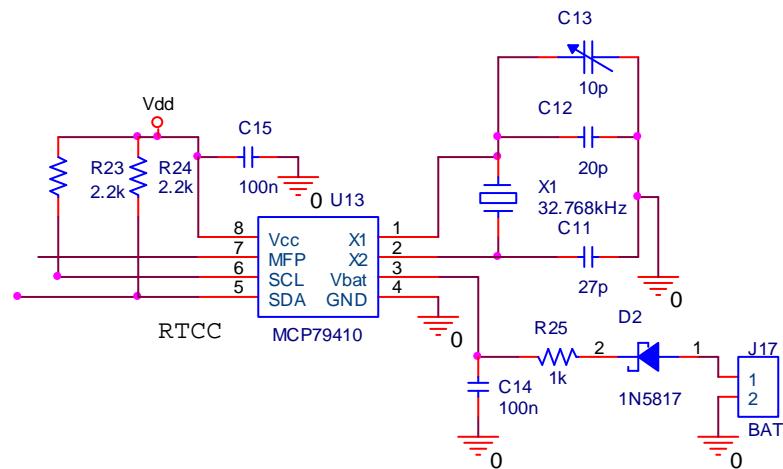
Тъй като аналоговото напрежение през мултиплексора е винаги положително, V_{EE} и V_{SS} захранващите изводи на микроконтролера се свързват към маса.

За термосензори са избрани ИС LM335. Свързан е по схема за калибруем термосензор по указанията на производителя - потенциометър със стойност от $10k\Omega$ е поставен в близост до сензора за да може да се калибрира.

Токоограничаващия резистор (R_8) е избран така, че да осигурява $2mA$ работен ток при захранващо напрежение от $+5V$. Работният ток е избран като компромис между необходима стабилност за работа на сензора, минимална собствена температурна грешка и товароспособност.

4.1.5 Часовник

На фигура 6 е показана схемата на свързване на часовника за реално време.



фигура 6- Часовник

Това опорно периферно устройство дава възможност на системата да включва като фактор в изчислителният процес по избор на режим и периода от денонощието. Освен това осигурява възможност за автоматично включване на осветлението в определен час от денонощието (Аларма).

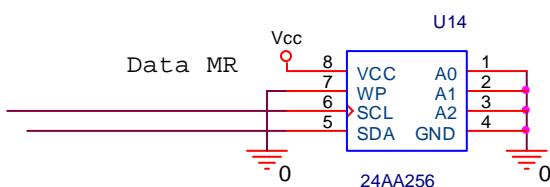
Схемата е изградена посредством ИС MCP79410 според указанията на фирмата производител. Към схемата е добавен тример кондензатор (C_{13}) за фина настройка на тактовата честота на кварцовия осцилатор.

Схемата комуникира с микроконтролера по I^2C интерфейс. Адреса на схемата е даден в секцията Интеграционни параметри на този документ.

Избора на резисторите за I^2C интерфејса е направен по препоръките на стандарта (I2C-bus specification and user manual - Rev. 5 - 9 October 2012) за работа в стандартен режим на работа, и товароспособността на интерфејса (7 периферни устройства) - $R_P = 2.2k\Omega$.

4.1.6 Памет за данни

На фигура 7 е показана схемата на паметта за данни допълнително инсталирана към системата.

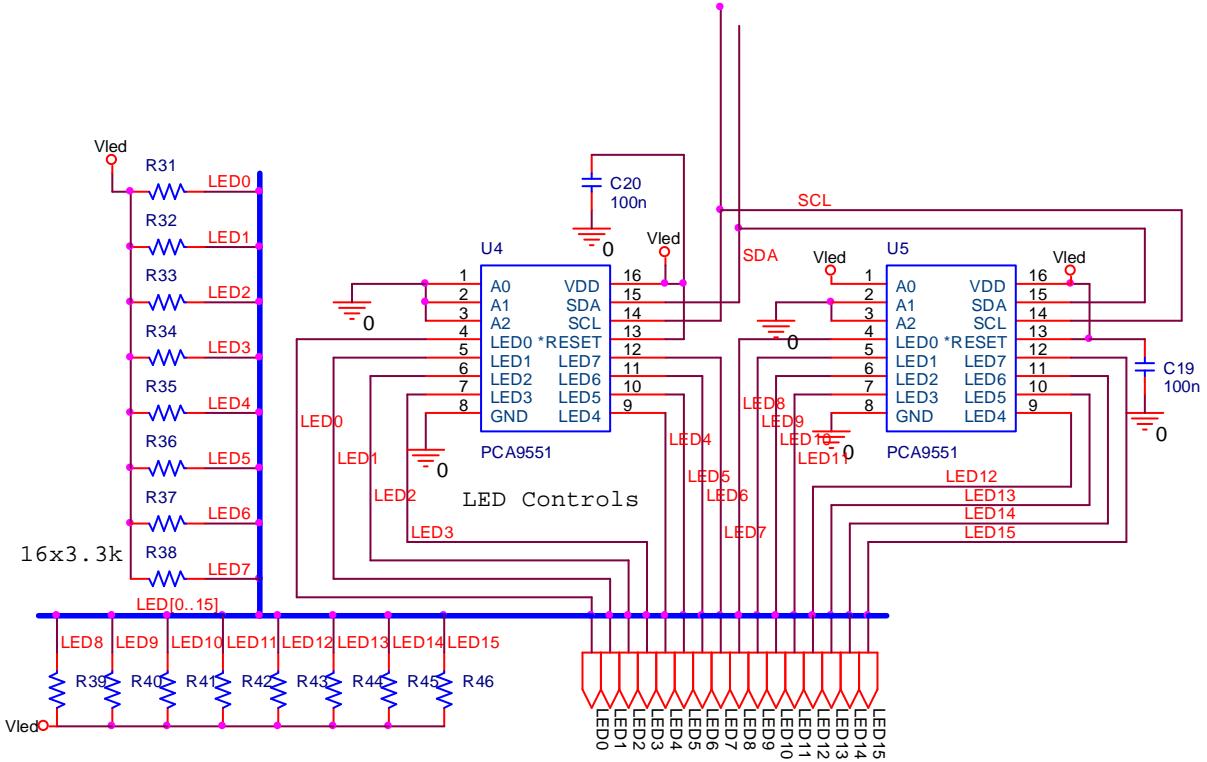


фигура 7 - Памет за данни

Изградена е на базата на ИС 24AA256. Свързана с микроконтролера посредством I^2C интерфејса. По същество тази схема осигурява 256kB пространство за съхранение на потребителски настройки.

4.1.7 Интерфејс за светодиодите:

На фигура 8 е показана схемата на интерфејсните схеми за управление на светодиодите.



фигура 8 - Светодиоден интерфейс

Стандартно за системата се предвижда управление на 9 групи светодиоди:

6 цветни групи: Виолетови, Сини, Зелени, Жълти, Оранжеви и Червени.

Цветните групи са предвидени да осигуряват специфични нюанси на светлината. Работейки самостоятелно или в комбинация тези шест цвята могат да обхванат широк цветови спектър, с което да осигурят подходяща атмосфера за специфична дейност.

2 бели групи: Студено бели и топло/неутрално бели - тази група светодиоди е предвидена като основен източник на бяла светлина. Когато е необходимо тя може да се комбинира и с някои от цветните групи или дори със всички - при нужда от максимална осветеност активирането на всички цветни и бели светодиоди ще отдават максималния поток от бяла светлина на системата.

Инфрачервена група - макар и невидима за човешкото око, много сензори и устройства са чувствителни към този тип лъчения. Причината да бъдат включени в проекта е възможността на системата

да бъде комбинирана с охранителна система. Така при нужда може да се осигури допълнителна светлина за охранителните камери.

За да се осигурят необходимите управляващи изводи с общо предназначение са използвани 2 ИС PCA9551. По същество те представляват 8 извода с общо предназначение управлявани по I²C интерфейс. Като допълнение те поддържат и локална ШИМ схема.

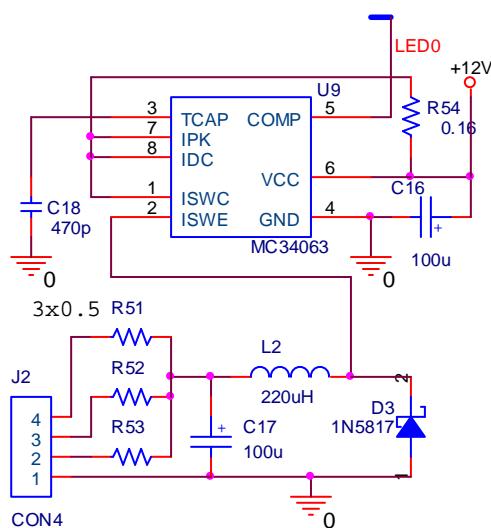
Изходните им буфери са с отворен дрейн. Това дава възможност за използване на външни токоограничаващи резистори, чрез които да се дефинира максималния консумиран ток в режим на работа.

външните резистори са избрани така, че максималният ток през всеки от тях се ограничава до 1mA (при ниско активно изходно ниво).

Свободните 7 управляващи пинове от втората ИС са изведени към куплунг на платката и могат да бъдат използвани като разширителен порт за разнообразни външни устройства съгласно указанията на производителя.

4.1.8 Контролери на светодиодите:

На фигура 9 е показана схемата на свързване на светодиодните контролери.



фигура 9 - Светодиоден контролер

Тя е осъществена с помощта на постояннотоковият преобразувател MC34063. Той е изграден по схема на понижаващ преобразувател съгласно инструкциите на производителя.

В изхода на схемата са добавени 3 извода буферириани с нискоомни резистора. Тези изходи осигуряват захранването на една и съща светодиодна (цветна) група за 3 отделни осветителни тела. Тъй като светодиодите в 3-те групи по същество ще бъдат свързани паралелно, трите буфериращи резистора ще служат за елиминиране на влиянието на производствения толеранс във светодиодите.

Тези резистори се избират с възможно най-малка стойност за да се минимизира загубата на мощност в тях в зависимост от характеристиките на използваните светодиоди.

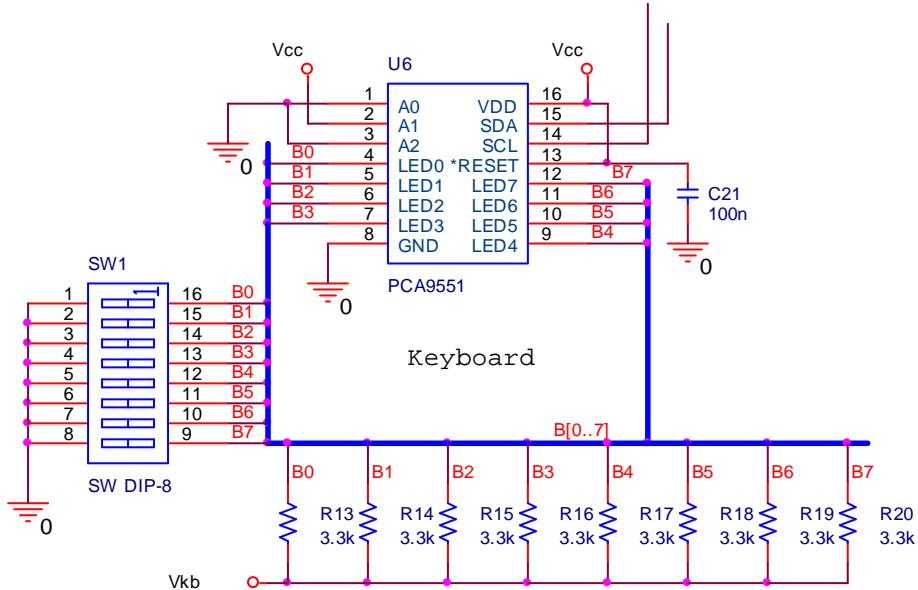
За целите на проекта са използвани 3 последователно свързани мощни диоди от по 1W за всяко от осветителните тела. От параметрите на светодиодите по спецификация следва, че максималният производствен толеранс за напрежението в права посока на тези диоди е $\pm 0,2V$. От където средното квадратично отклонение е $\pm 0,14V$. При 350mA максимален ток през група и максимален пад на напрежение 0,14V върху буферния резистор, се определя неговата стойност от $0,5\Omega$. и мощност от 0,06W.

За трите осветителни тела максималният консумиран ток е $3 \times 0,35A = 1,05A$. Максималната товароспособността на ИС е 1,5A.

Резистора за контрол на тока през конвертора се изчислява според указанията на производителя за изходен ток от 1,05A - съответно $0,3/(2 \times I_{OUTmax}) = 0,16\Omega$ и мощност от 0,18W.

4.1.9 Клавиатура

На фигура 10 е показана схемата на свързване на клавиатурата.



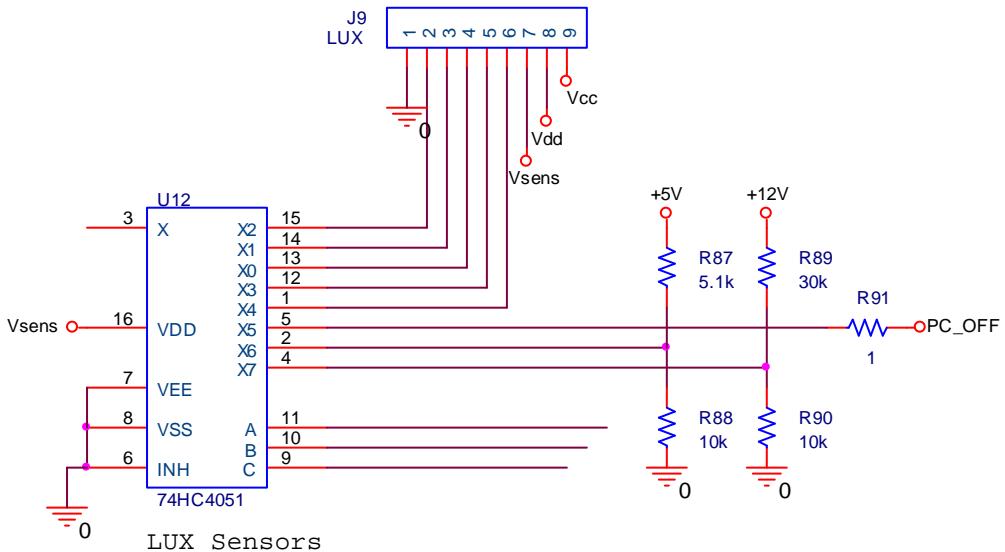
фигура 10 - Клавиатура

Схемата е организирана отново на базата на ИС PCA9551, но този път се използват възможностите и за четене на състоянието на пиновете. Външно са включени позиционни клавиши които определят високото или ниското състояние на входа на всеки извод на ИС. На базата на комбинацията от позициите на ключетата могат да се задават различни комбинации за конфигуриране на системата ръчно. Зареждането на комбинацията става посредством натискането на бутона OK (виж секция бутони тази глава).

Всеки ключ е захранен посредством резистор. Токът през всеки ключ е ограничен до 1 mA, но тъй като схемата е снабдена със система за управление на мощността, напрежение към клавищите се подава едва след като бутона OK е натиснат. По този начин се минимизира средната стойност на използвания ток от клавиатурата.

4.1.10 Сензори за осветеност и напрежение:

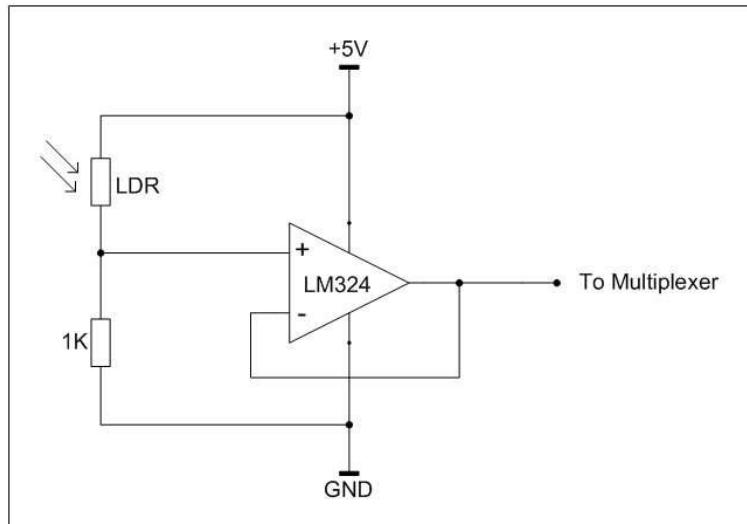
На фигура 11 е показана схемата на свързване на сензорите за осветеност.



фигура 11 - Сензори за осветеност

Както и при термосензорите отчитането на сензорите за осветеност става посредством аналогов мултиплексор и измерване с АЦП-то на микроконтролера. Предвидени са входове за 5 сензора.

Вариант за реализиране на сензор за осветеност е показан на фигура 12



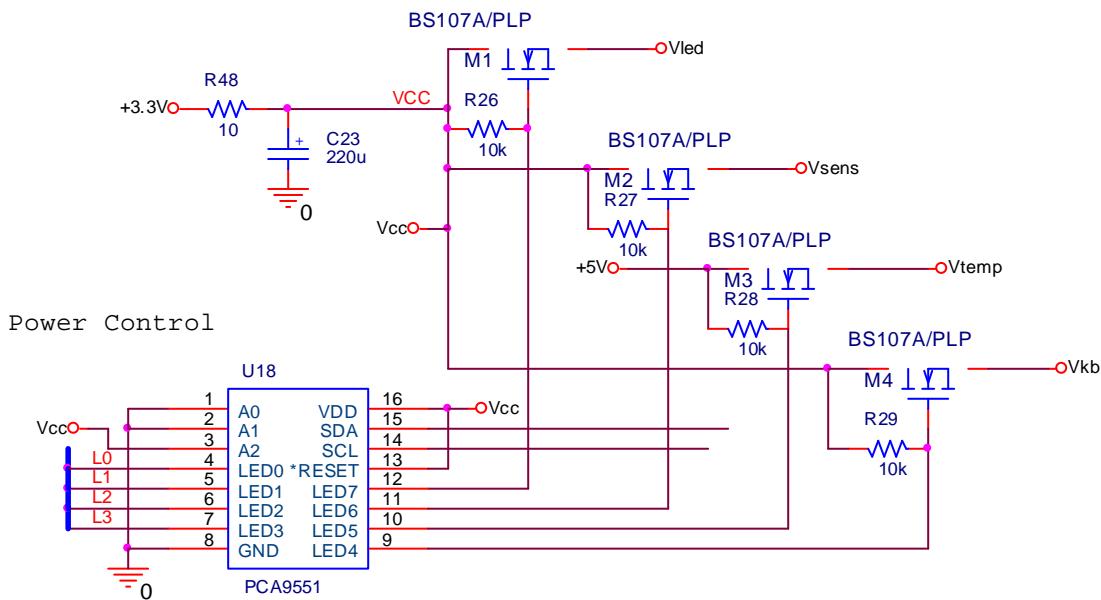
фигура 12 - Схема на свързване на сензора за осветеност

Делителите на напрежение са изградени за да може микроконтролера да следи нивата на захранващите напрежения. Това дава възможност на системата да бъде използвана като част от фотоволтаична система.

Допълнителният извод PC_OFF е предвиден като опция за сигнализиране от външна система при необходимост за изключване на устройството.

4.1.11 Управление на мощността

На фигура 13 е показана реализацията на схемата за управление на мощността.



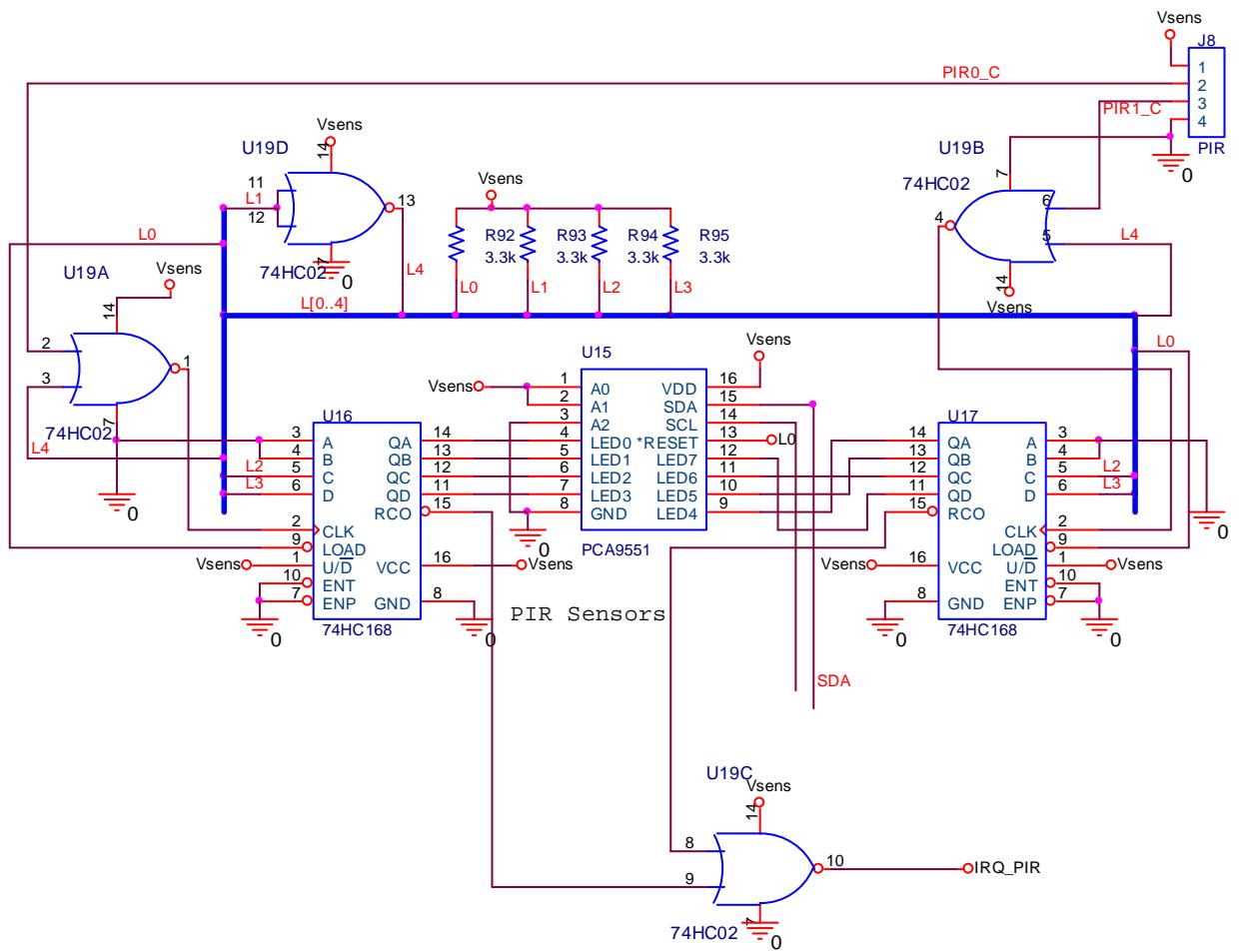
фигура 13 - Управление на мощността

С помощта на половината изводи от ИС PCA9551 се управляват гейтовете на 4 MOS транзистора свързани като ключове. Целта им е да ограничат консумацията на мощност от отделни групи периферни устройства на системата.

*другата половина на ИС се използва за програмиране на PIR броячите - виж секция *Сензор за движение с PIR датчик*

4.1.12 Сензор за движение с PIR датчик

На фигура 14 е показана схемата осигуряваща отчитането на сигнала от PIR сензорите.



фигура 14 - PIR Сензори

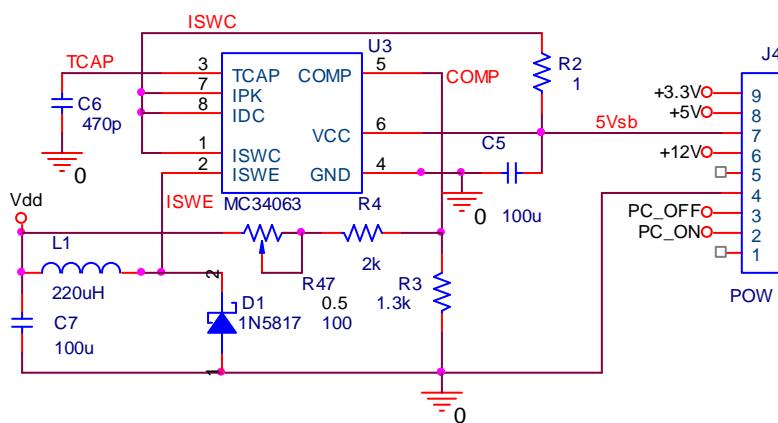
По същество схемата представлява брояч който отчита броя на импулсите подадени от PIR сензора в следствие регистрация на движение в охраняваната област. Стойността на брояча се отчита периодично от микроконтролера посредством прочитане на стойността през входовете на ИС PCA9551. Предвидени са входове за два PIR сензора.

Изходите за пренос на броячите са свързани посредством логическа схема ИЛИ с вход за външно прекъсване на микроконтролера. По този начин в случай на препълване (твърде интензивно движение), микроконтролера ще бъде реконфигуриран да скъси цикъла на проверка на сензора. В случаите когато е включен режим на охрана, този ефект може да предизвика задействане на охранителната система.

С помощта на втората половина от ИС PCA9551 използвана в схемата за управление на мощността след всяко прочитане сензора може да се нулира или да се зареди със стойност.

4.1.13 Захранване

На фигура 15 е показана схемата осигуряваща захранване на ядрото на системата (V_{dd} домейна). Той осигурява захранването на микроконтролера и часовника. Необходимо е той да подава непрекъснато захранване за да може системата да функционира правилно.

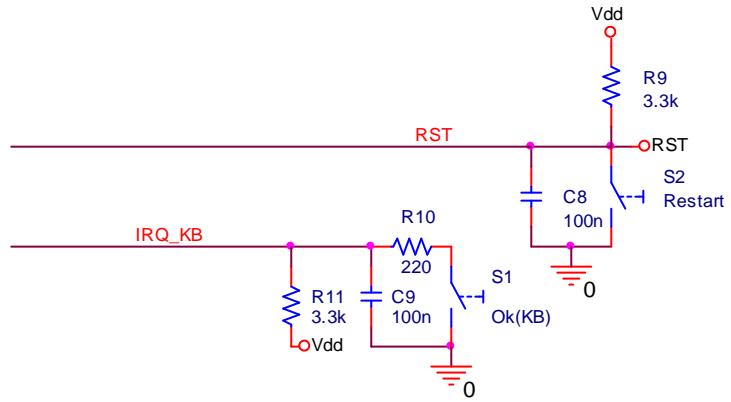


фигура 15 - Захранващ блок

Тъй като източниците на захранване за системата могат да бъдат разнообразни, а постоянното захранващо напрежение което те осигуряват в режим на готовност може да варира значително (от +5V при някои импулсни захранвания, до 12 и 24V при използване на акумулатори в соларни системи), схемата е изградена с помошта на понижаващ постояннотоков преобразувател. Максималният изходен ток е ограничен до 150mA.

4.1.14 Бутони

На фигура 16 е показано изграждането на двета бутона на системата - OK и Reset.



фигура 16 - Бутони

Изградени са с помощта на микробутони. RC веригата осигурува работен ток от 1mA и филтрация на шумовете на бутоните. Към веригата на бутона OK е добавен и последователен резистор за защита на изходния буфер (в случай на неправилно конфигуриране като изход).

4.2 Интеграционни параметри

В таблица 1 са показани хардуерните интеграционни параметри на системата.

Таблица 1- Хардуерни интеграционни параметри

Модул	Използвана ИС	Захранваща линия	Напрежение	Адрес
RS232 интерфейс	MAX3232	Vcc	+3,3V	-
IR интерфейс	TSOP34836	Vcc	+3,3V	-
Бутони		Vdd	+3,3 V _{SB}	-
Power-ON Control	BC107	PC_ON	-	-
Power Domain Control	PCA9551	Vcc	+3,3V	1100100b
Data Memory	24AA256	Vcc	+3,3V	1010000b
RTCC EEPROM SRAM	MCP79410	Vdd	+3,3 V _{SB}	1010111b
				1101111b
LED Control 1	PCA9551	Vled	+3,3V	1100000b
LED Control 2	PCA9551	Vled	+3,3V	1100001b
PIR сензор	PCA9551	Vsens	+3,3V	1100011b
Клавиатура	PCA9551	Vkb	+3,3V	1100010b
Температурен сензор	LM335	Vtemp	+5V	-
Сензор за осветеност		Vsense	+3,3V	-
Микроконтролер	MSP430G2553	Vdd	+3,3 V _{SB}	Master

Адресите в таблицата са адресите на подчинените устройства на I²C шината.

В таблица 2 са дадени софтуерните настройки на системните тактови сигнали.

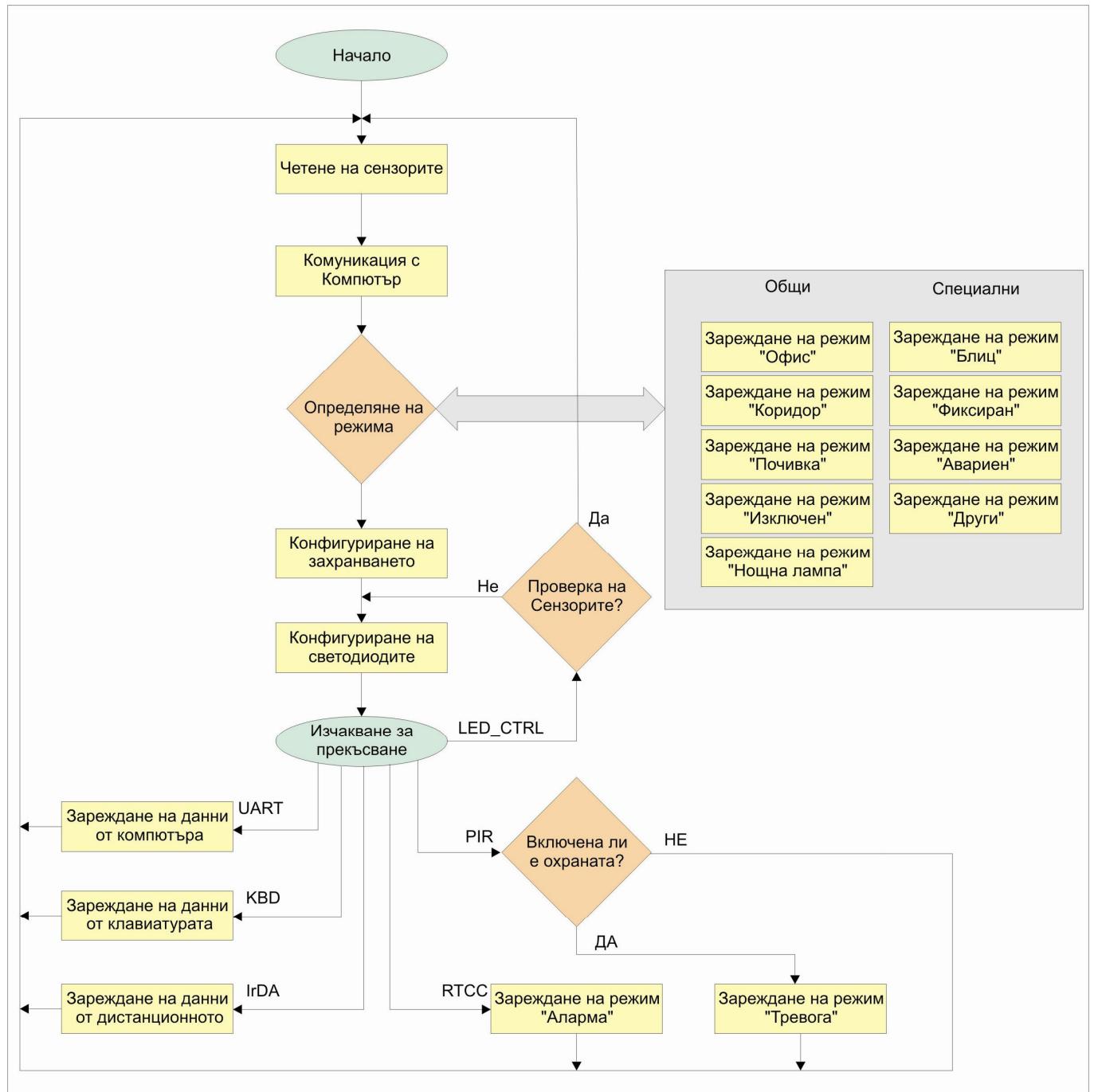
Таблица 2 - Системни тактови сигнали

Тактов сигнал	Честота	Делител	Свързани устройства
MCLK	8 MHz	-	CPU; SMCLK
SMCLK	8 MHz	/1	I ² C; UART; Timer0;
SCL (I ² C)	100 kHz	/80	PCA9551; 24AA256; MCP79410
Timer0 Clock	4 MHz	/2	LED_Drivers
UART	9600 bps	/833	RS232

5 Проектиране на програмното осигуряване

5.1 Блок-алгоритъм на програмата

На фигура 17 е представен блок-алгоритъмът на програмата.



фигура 17 - Блок-алгоритъм на програмата

5.2 Програмен алгоритъм

В тази секция са представени конкретни демонстрационни фрагменти от програмния код на устройството. Целта на тези програмни фрагменти е да изяснят ключови архитектурни компоненти на управляващият алгоритъм.

Целият текст на програмата заедно с всички файлове необходими за компилиране са представени в компакт диска към приложението.

Алгоритъмът на системата се базира на 4 основни структури - комуникация с потребителя, управление на компонентите, автоматично реконфигуриране и снемане на данните от сензорите.

Тези 4 структури се базират на 3 основни микроалгоритъма - управление на серийният интерфейс, управление на вътрешно модулният интерфейс, управление на таймерната система.

5.2.1 Сериен интерфейс за комуникация с персонален компютър (UART)

В този фрагмент е представена подпрограмата осигуряваща комуникацията през серийният интерфейс.

```
#include "msp430g2553.h"

const char string1[] = { "Welcome to ALIS\r\n" };
unsigned int i;

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT

    P1DIR |= (BIT0|BIT5); // Set the LEDs as outputs
    P1OUT = 0x01;

    BCSCTL1 = CALBC1_8MHZ; // Set DCO
    DCOCTL = CALDCO_8MHZ;
    P1SEL = BIT1 + BIT2; // P1.1 = RXD, P1.2=TXD
    P1SEL2 = BIT1 + BIT2; // P1.1 = RXD, P1.2=TXD
    UCA0CTL1 |= UCSSEL_2; // SMCLK
    UCA0BR0 = 56; // 8MHz/9600 = 833
    UCA0BR1 = 3; //
    UCA0MCTL = UCBSR1; // Modulation UCBRSx = 2
    UCA0CTL1 &= ~UCSWRST; // Initialize USCI state machine
    IE2 |= UCA0RXIE; // Enable USCI_A0 RX interrupt

    __bis_SR_register(LPM0_bits + GIE); // Enter LPM0, interrupts enabled
}

#pragma vector=USCIAB0TX_VECTOR
__interrupt void USCI0TX_ISR(void)
```

```

{
    if (IE2&UCA0TXIFG) //Check if the IRQ is from UART
    {
        UCA0TXBUF = string1[i++];
                                // TX next character

        if (i == sizeof string1 - 1)           // TX over?
            IE2 &= ~UCA0TXIE;
                                // Disable USCI_A0 TX
interrupt
    }
}

#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR(void)
{
    if (UCA0RXBUF == 'e')                  // 'e' received?
    {
        i = 0;
        IE2 |= UCA0TXIE;
                                // Enable USCI_A0 TX interrupt
        UCA0TXBUF = string1[i++];
    }

    if (UCA0RXBUF == 'R')
    {
        P1OUT |= BIT0; // Turn on P1.0 red LED when R is received
    }
    else if (UCA0RXBUF == 'r')
    {
        P1OUT &= ~BIT0; // Turn off P1.0 red LED when r is received
    }
    if (UCA0RXBUF == 'G')
    {
        P1OUT |= BIT5; // Turn on P1.6 green LED when G is received
    }
    else if (UCA0RXBUF == 'g')
    {
        P1OUT &= ~BIT5; // Turn off P1.6 green LED when g is received
    }
}

```

По същество функционалността на тази микропрограма се изразява в разпознаване на входни данни получени по серийният интерфейс и реакция в зависимост от данните - да се включи/изключи червеният или зеленият светодиод или да изпрати съобщение към потребителят.

5.2.2 Вътрешно модулен интерфейс (I^2C)

Чрез I^2C интерфейса се осигурява контролирането на всички активни вътрешно модулни компоненти. Системата разполага с 5 контролни модула от тип PCA9551 (GPIO), един от тип MCP79410 (RTCC) и един от тип 24AA256 (памет за данни). Особеност в адресирането има само при данновата памет - тя има нужда от подаване на два адресни байта. Тази

особеност е отразена във функцията за предаване на данни (`i2c_tx()`), когато предстои обръщение към паметта (адрес 0x50).

Работните масиви за приемане и предаване на данните са заложени с голям размер за да може с една операция да се извършва записа на целия набор от данни в паметта (9 цветови стойности и реперни точки за сензорите)

```
#include "msp430g2553.h"

unsigned char txdata[16] = {0x05, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}; //Transmit placeholder"
unsigned char rxdata[14]= {0xCC, 0xAA, 0x00, 0xEE, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}; //Receive placeholder
unsigned char i2c_count;
unsigned char tx_counter;
unsigned char rx_counter;
unsigned char tx_rx;

void i2c_tx(int txf);
void i2c_rx(void);

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT
    BCSCTL1 = CALBC1_8MHZ; // Set DCO
    DCOCTL = CALDCO_8MHZ;

    P1SEL |= BIT6 + BIT7; //Set I2C pins
    P1SEL2|= BIT6 + BIT7;
    UCB0CTL1 |= UCSWRST; //Enable SW reset
    UCB0CTL0 = UCMST + UCMODE_3 + UCSYNC; //I2C Master, synchronous mode
    UCB0CTL1 = UCSSEL_2 + UCSWRST; //Use SMCLK, keep SW reset
    UCB0BR0 = 80;
    //fSCL = SMCLK/12 = ~100kHz
    UCB0BR1 = 0;
    UCB0I2CSA = 0x60; //Slave Address
    UCB0CTL1 &= ~UCSWRST; //Clear SW reset, resume operation
    IE2 |= UCB0TXIE; //Enable TX interrupt
    IE2 |= UCB0RXIE; //Enable RX interrupt
    int i=0;

    i2c_rx();
    tx_rx = rxdata[0];

    while (1)
    {
        for (i=0;i<4;i++)
        {
            txdata[1]=tx_rx<<2;
            txdata[1] |= BIT0;
            i2c_tx(0);
            __delay_cycles(200000);
            tx_rx=txdata[1];
        }
        for (i=4;i>0;i--)
        {
            txdata[1]=tx_rx<<2;
            txdata[1] &= ~BIT0;
            i2c_tx(0);
            __delay_cycles(200000);
        }
    }
}
```

```

        tx_rx=txdata[1];
    }
}

void i2c_tx(int txf)
{
    //Prepare counters
    if (txdata[0]==0x50)    //for R/W to D_MR it should send 2 bytes addr.
    {
        if (txf==0) //if the function is needed to write information
        {
            i2c_count=16; // prepare entire set of data to save
        }
        else          //if the function is needed to read information
        {
            i2c_count=2;
        }
    }
    else //for r/w to PCA9551 and RTCC (MCP79410) just 1 address byte
    {
        if (txf==0) //if the function is needed to write information
        {
            i2c_count=2;
        }
        else          //if the function is needed to read information
        {
            i2c_count=1;
        }
    }

    tx_counter = i2c_count;           // Load TX byte counter
    UCB0CTL1 |= UCTR + UCTXSTT;     // I2C TX, start condition
    __bis_SR_register(CPUOFF + GIE); // Enter LPM0 w/ interrupts

    // Remain in LPM0 until all data is TXed
}

void i2c_rx(void)
{
    i2c_tx(1);

    if (txdata[0]==0x50)    //for read from the data memory it should
receive 14 byte address
    {
        i2c_count=14;
    }
    else //for read from PCA9551 and RTCC (MCP79410) just 1 data byte
    {
        i2c_count=1;
    }

    rx_counter = i2c_count;           // Load RX byte counter
    UCB0CTL1 &= ~UCTR;             // I2C RX
    UCB0CTL1 |= UCTXSTT;            // I2C start condition
    __bis_SR_register(CPUOFF + GIE); // Enter LPM0 w/ interrupts

    // Remain in LPM0 until all data is RX'd
}

#pragma vector=USCIAB0TX_VECTOR
__interrupt void USCI0TX_ISR(void)
{

```

```

if (IFG2&UCB0TXIFG) // (tx_rx == 0)
{
    if (tx_counter > 0)      //Check TX byte counter
    {
        UCB0TXBUF = txdata[i2c_count - tx_counter];
        // Load TX buffer
        tx_counter--;
        //Decrement TX byte counter
    }
    else if(tx_counter == 0)
    {
        UCB0CTL1 |= UCTXSTP;           //I2C stop condition
        while (UCB0CTL1 & UCTXSTP); //Ensure stop condition sent
        IFG2 &= ~UCB0TXIFG;
        //Clear USCI_B0 TX int flag
        __bic_SR_register_on_exit(CPUOFF); //Exit LPM0
    }
}
else if(IFG2&UCB0RXIFG)
{
    if (rx_counter > 0)      //Check RX byte counter
    {
        rxdata[i2c_count - rx_counter] = UCB0RXBUF;
        rx_counter--;      //Decrement RX byte counter
    }
    else if(rx_counter == 0)
    {
        UCB0CTL1 |= UCTXSTP;           // I2C stop condition
        while (UCB0CTL1 & UCTXSTP); // Ensure stop condition sent
        rxdata[i2c_count - rx_counter] = UCB0RXBUF;
        rxdata[i2c_count - rx_counter] = UCB0RXBUF;
        IFG2 &= ~UCB0RXIFG;// Clear USCI_B0 RX int flag
        __bic_SR_register_on_exit(CPUOFF); // Exit LPM0
    }
}
}

```

Функцията на демонстрационната програмата тук е сведена до последователно просветване и изгасяне на 4 светодиода вързани на изхода на PCA9551.

По същество обаче, така изградените функции за предаване и приемане на данни могат да се използват за всички необходими операции на системата.

5.2.3 Таймерна система (Timer0_A3)

По същество това е основният двигателен механизъм чрез който действа системата.

```

#include "msp430g2553.h"

//Pin Definitions
#define ASTOP          BIT6 //Output ASTOP P1.0
#define BTN            BIT3 //BTN Input P1.2
#define RED_LED        BIT0 //Output A P1.3

```

```

#define      GRN_LED           BIT4 //Output B P1.4
#define      BLU_LED           BIT5 //Output B P1.4

// Variable definitions
int automode=0;           //The automode flag;
int temp=0;                //Temperature data;
int lux=0;                 //Lux sensor data;
int pir=0;                 //PIR sensor data;
int mode=0;                //selected mode;
int red=0; //Red color length
int grn=0; //Red color length
int blu=0; //Red color length
int button=0; //button mode TBR
int array[4];
int flag=0;

void Preconfiguration (void); //Initial configuration
//void CompIF (void);           //This function organize the
communication with PC
void Night_Light (void);      //Night Light mode
void Office_Light (void);     //Office light mode
void Rest_Light (void);       //Rest light mode
void Off_light (void);        //Switch-off light mode
void sensdata (void);         //this function checks sensors
void Brain (void);           //This function analize sensors
data and selects mode
void LED_Driver (void);       //This function controls the LEDs

void main (void)
{
    Preconfiguration ();
    while (1) //infinity loop; The work is via interrupts
    {
        //CompIF ();
        Brain ();
        LED_Driver ();
    }
}

void Brain (void)
{
    if (automode==1)
    {
        sensdata ();
    }
    switch (mode)
    {
        case 1 : Off_light(); break;
        case 2 : Night_Light(); break;
        case 3 : Rest_Light(); break;
        default : Office_Light(); break;
    }
    P1OUT &= ~ASTOP;
    button=0;
}

void LED_Driver (void)
{
    //Prepire Timer A
    TACTL |= TACLR;           //Resets TA.
    TACCTL0 &= ~CCIFG;        //Clear Interrupt Flag
}

```

```

TACCTL0 |= CCIE;                                //CCR0 interrupt enabled
TACTL = TASSEL_2 + ID_1;                         //SMCLK/2, 4MHz

//Prepare variables
array[0] = red;
array[1] = grn;
array[2] = blu;
array[3] = 256;
int i,j;
for(i=0;i<4;i++)
{
    for(j=0;j<i;j++)
    {
        if(array[i]<array[j])
        {
            int test=array[i]; //swap
            array[i]=array[j];
            array[j]=test;
        }
    }
}
flag=0;
while (array[flag]==0) //eliminate the zero values
{
    flag++;
    if (flag==3) //can't find nonzero value
    {
        P1OUT |= ASTOP;
        flag=0;
        P1OUT &= ~RED_LED + ~GRN_LED + ~BLU_LED;
        break;
    }
}

if (array[flag]>0)
{
    TACCR0 = (array[flag]*256)-1;                // define counter value

    if (red>0)
    {
        P1OUT |= RED_LED;
    }
    if (blu>0)
    {
        P1OUT |= BLU_LED;
    }
    if (grn>0)
    {
        P1OUT |= GRN_LED;
    }

    TACTL |= MC_2;                               //start timer Continuous mode
}

while(button == 0)                                //wait for BTN interrupt
{}
TACTL |= MC_0;                                    //stop timer

}

// Timer A0 interrupt service routine
#pragma vector=TIMER0_A0_VECTOR

```

```

__interrupt void Timer_A (void)
{
    if (array[flag]==red)
    {
        P1OUT &= ~RED_LED;
    }
    if (array[flag]==grn)
    {
        P1OUT &= ~GRN_LED;
    }
    if (array[flag]==blu)
    {
        P1OUT &= ~BLU_LED;
    }

    if (array[flag]==256)
    {
        if (red>0)
        {
            P1OUT |= RED_LED;
        }
        if (blu>0)
        {
            P1OUT |= BLU_LED;
        }
        if (grn>0)
        {
            P1OUT |= GRN_LED;
        }
        flag=0;
    }
    else flag++;
}

TACCR0 = (array[flag]*256)-1;           // define counter value

TACCTL0 &= ~CCIFG;           //Clear Capture/compare interrupt flag
}

void Night_Light (void)
{
    red=5;
    grn=128;
    blu=10;
}

void Office_Light (void)
{
    red=256;
    grn=256;
    blu=256;
}

void Rest_Light (void)
{
    red=128;
    grn=10;
    blu=5;
}

void Off_light (void)
{
    red=0;
    grn=0;
    blu=0;
}

```

```

}

void sensdata (void)
{
    mode++;
    if (mode>3)
    {
        mode=0;
    }
    P1OUT &= ~RED_LED + ~GRN_LED + ~BLU_LED;
}

void Preconfiguration (void)
{
    WDTCTL = WDTPW + WDTHOLD;      // Stop WDT

    /* Use calibration values to set digitally controlled oscillator
     * to 8 MHz, and set the submain clock to the same
     */
    BCSCTL1 = CALBC1_8MHZ;
    DCOCTL = CALDCO_8MHZ;
    BCSCTL2 &= ~(DIVS_3); //SMCLK devider cleared (goes from "/8" to "/1")

    //Define port directions
    P1DIR |= ASTOP + RED_LED + GRN_LED + BLU_LED; //Defines outputs
    P1OUT = 0x00; //define output value
    _BIS_SR(GIE); //Enable interrupts

    //Set Port1 for Buton IRQ
    P1IES |= BTN;                      // P1.2 Hi/lo edge
    P1IFG &= ~BTN;                    // P1.2 IFG cleared
    P1IE |= BTN;                      // P1.2 interrupt enabled
}

#pragma vector=PORT1_VECTOR
__interrupt void Port_1(void)
{
    CCTL0 &= ~CCIE;    // CCR0 interrupt disabled
    P1IE &= ~BTN;    // P1.2 interrupt disabled
    TACTL |= MC_0;    //stop timer
    buton = 1;        //Change the working mode
    sensdata();       //Check sensors (change mode)
    P1IFG &= ~BTN;    // P1.3 IFG cleared
    P1IE |= BTN;    // P1.2 interrupt enabled
}

```

В дадената микропрограма е демонстриран по опростен начин моделът на работа на системата, а именно:

Посредством различни аппаратни методи се задава режима на работа на системата - в случая това се прави посредством натискането на бутон, който избира между 4 предефинирани алгоритъма на работа. В крайният вариант това ще става на базата на данните от сензорите във функцията sensdata () - тя ще извлича данните от сензорите и ще ги обработва.

След като са обработени, данните се предоставят на подпрограма Brain(), която решава дали ще изпълнява автоматичен режим или ще задава статични данни. След което зарежда данните в съответните променливи.

На подпрограмата LED_Driver () е дадена задачата да сортира данните по ред на изпълнение и да стартира таймера.

Таймера от своя страна започва просветването на светодиодите и изключването им на определени интервали от време според зададените данни.

5.3 Инструкции за програмиста

При разиване на представеният програмен алгоритъм трябва да се вземат под внимание някои характерни особености на системата:

При комуникация с различни чипове по I²C интерфейса е необходимо да се запише адреса на целевия чип в txdata[0:1] преди да се извика функцията за приемане предаване на информация.

Когато се използва функцията за запис на данни по I²C интерфейса директно (извиква се директно функцията i2c_tx(int txf)), като параметър се записва 0: i2c_tx(0). Този параметър служи за обозначаване дали функцията се ползва за предаване на данни или е извикана в процеса на четене (изпраща се само адреса).

Когато се извършва запис на информация в паметта за данни, задължително трябва да се осигури времезакъснение необходимо на паметта за съхраняване на данните, преди отново да се извърши обратъщение към нея.

Важна особеност на PCA9551 чиповете е че те интегрират в себе си 2 ШИМ модула. Те могат да бъдат настройвани и стартирани съгласно инструкциите на фирмата производител, когато това е необходимо.

Системата разполага с апаратен механизъм за ограничаване на консумацията на мощност, поради което голяма част от периферията е изключена когато нейната работа не е необходима. Това трябва да се има в предвид при добавяне на алгоритми боравещи с клавиатурата или сензорите. Управлението на захранването се извършва с помощта на LED[7:4] от PCA9551 на адрес 0x64.

5.4 Инструкции за потребителя

5.4.1 Настойки на серииният порт:

Серииният порт на устройството е настроен за работа със следните параметри:

Baud Rate : 9600

Stop bit: 1

Parity: none

Hardware flow control: none

5.4.2 Потребителски команди за работа със системата през хипертерминал:

Стартиране (бяла светлина) - ASCII: B

Спиране (изключване) - ASCII: E

Извличане на данните от сензорите - ASCII: G

Зареждане на потребителска настройка - ASCII: S

Зареждане на блиц режим - ASCII: F

Зареждане на автоматичен режим - ASCII: A

6 Конструктивна част

6.1 Принципна схема

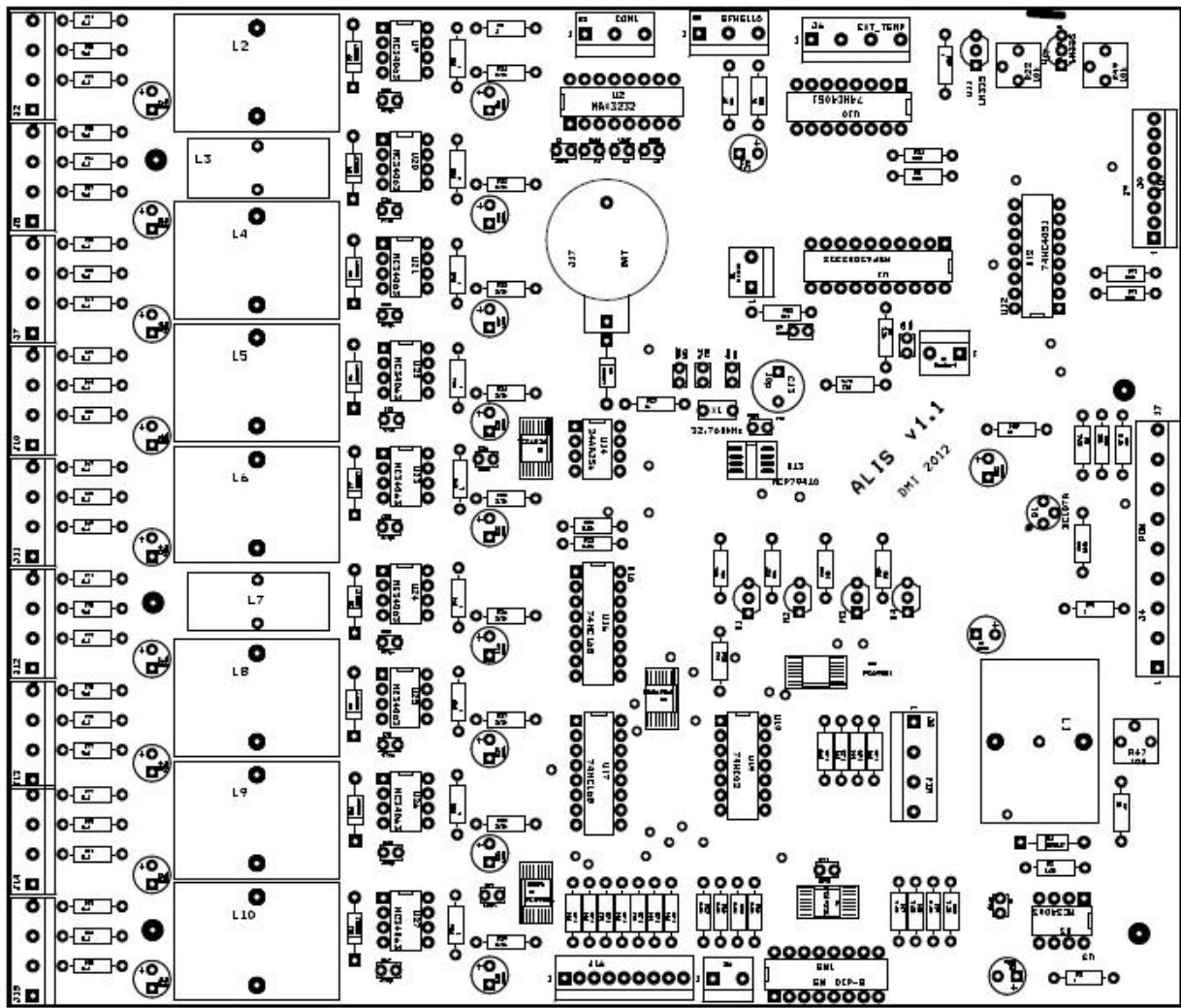
Принципната схема на устройството е представена в два чертежа - принципна схема на основната част и принципна схема на управляващите контролери.

Схемите са представени в приложението!

6.2 Печатна платка

6.2.1 Разположение на компонентите:

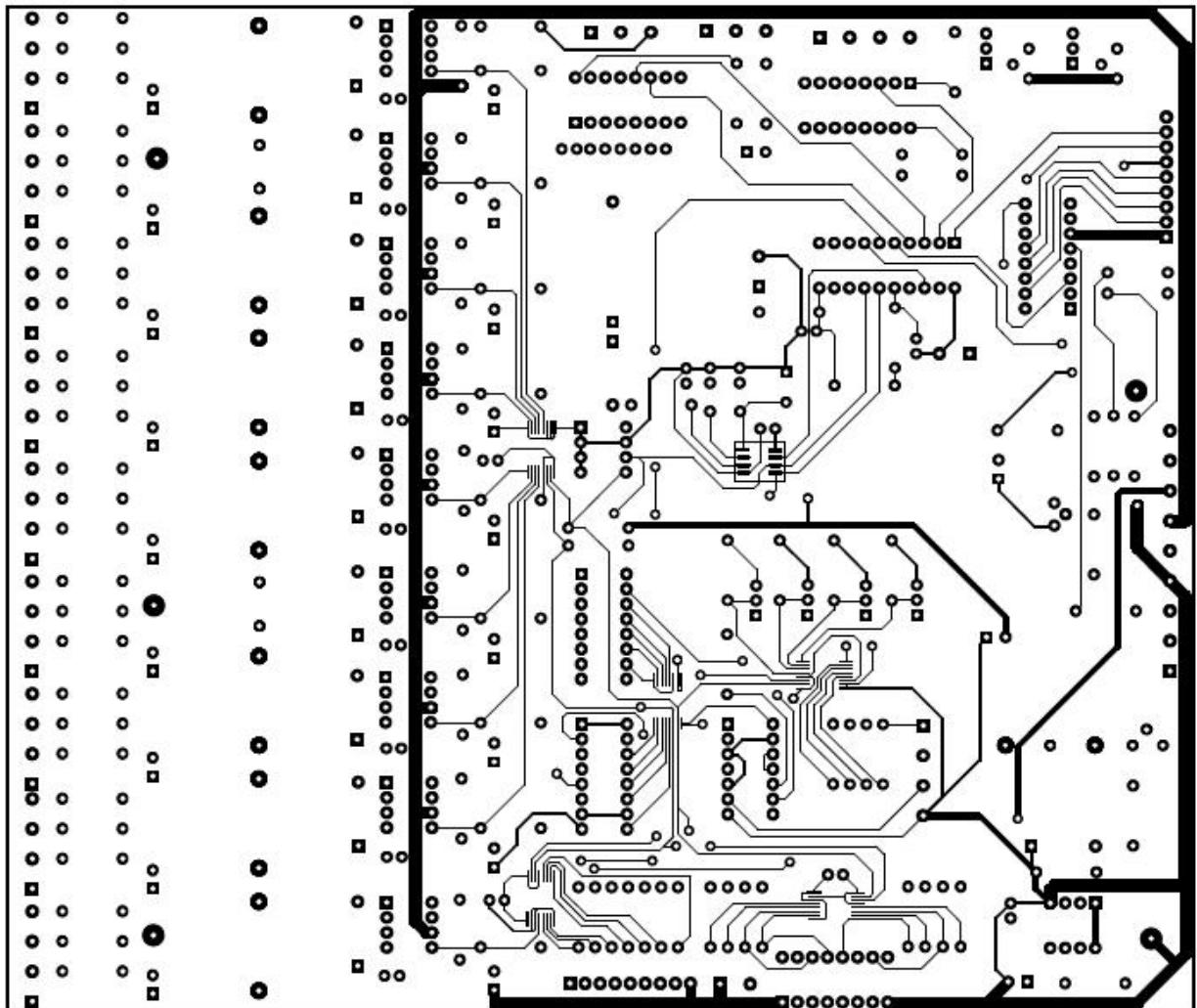
На фигура 18 е показан дизайна на печатната платка в изглед "разположение на елементите"



фигура 18 - Разположение на елементите

6.2.2 Горен слой:

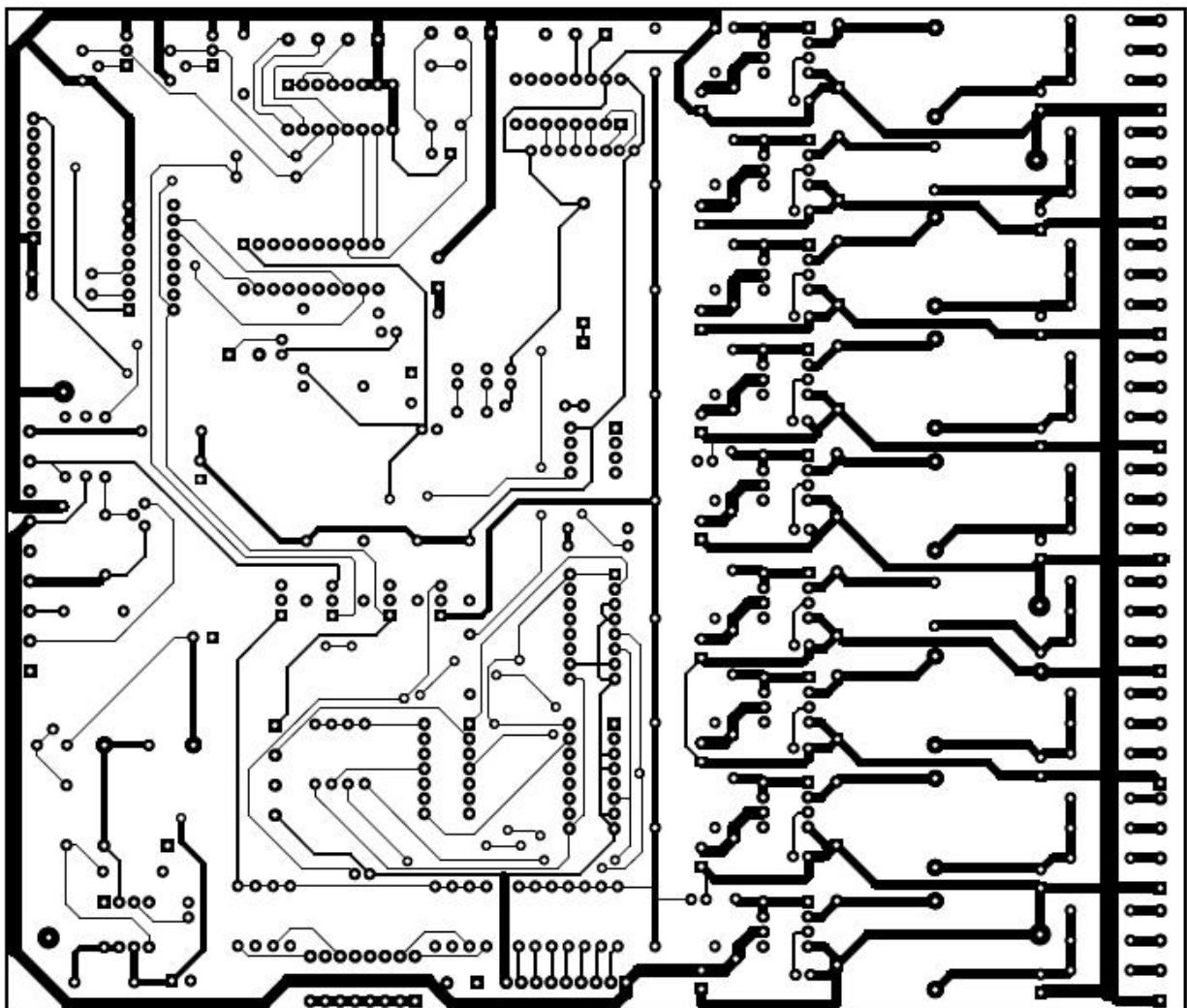
На фигура 19 е показан дизайн на печатната платка в изглед горен слой.



фигура 19 - Горен слой

6.2.3 Долен слой:

На фигура 20 е показан дизайна на печатната платка в изглед долен слой.



фигура 20 - Долен слой

6.3 Спецификация на елементите

Спецификацията на използваните компоненти е представена в приложението!

6.4 Модел на осветително тяло

За целите на проекта е разработен и модел на осветителното тяло работещо със системата.

6.4.1 Модел

За да може осветителното тяло да бъде монтирано лесно в съществуващите интериорни решения за аплици на осветителни тела, то трябва да има подходяща форма.

Най-често срещаните аплици са с форма на полукръг и кръг с радиус 9,5 см. Поради тази причина за форма на осветителното тяло се избира полукръг с радиус 9 см. Това ще позволи на тялото за бъде монтирано в аплик с форма на полукръг, а в аплиците с форма на кръг ще може да се монтират 1 или 2 тела по избор.

6.4.2 Проектиране на осветителното тяло

За да се осигури безопасното и дълготрайно функциониране на тялото е необходимо светодиодите да се охлаждат правилно.

Съгласно правилата за изчисляване на охлаждащи радиатори изчисляваме необходимата максимална разсейвана мощност:

Светодиодни групи = 7; (в случая не се включват виолетовите и инфрачервените диоди тъй като те нямат изискване за охлаждащ радиатор)

Брой светодиоди в група = 3

Максимална разсейвана мощност от всеки светодиод = 1W

Необходима площ на охлаждащия радиатор за 1 светодиод е:

$$S_{PC} = \frac{1200P_C}{T_j - T_A - P_C(R_{tc} + R_{cp})} = \frac{1200 \times 1}{150 - 35 - 1.(10 + 1)} = 11,5 \text{ cm}^2$$

Сумарната площ на радиатора за всички светодиоди е:

$$S_{\text{tot}} = 21 \times 11,5 = 242 \text{ cm}^2.$$

Поради тази причина и поради ограниченото място в аплиците, се избира нестандартна платка изработена от текстолит с алуминиева подложка.

При заложената форма на печатната платка, алуминиевата подложка би осигурила радиаторна площ от около 127 cm^2 .

За останалите 115 cm^2 необходима радиаторна площ се предвижда монтирането на допълнителен охлаждащ радиатор.

За целта са предвидени 4 монтажни отвора в печатната платка, които ще осигурят закрепянето на радиатора.

Самият радиатор трябва да е с височина не повече от 2 см (стандартната дълбочина на аплиците е 2,5 см, без да се включва обема между стъклото и основата.)

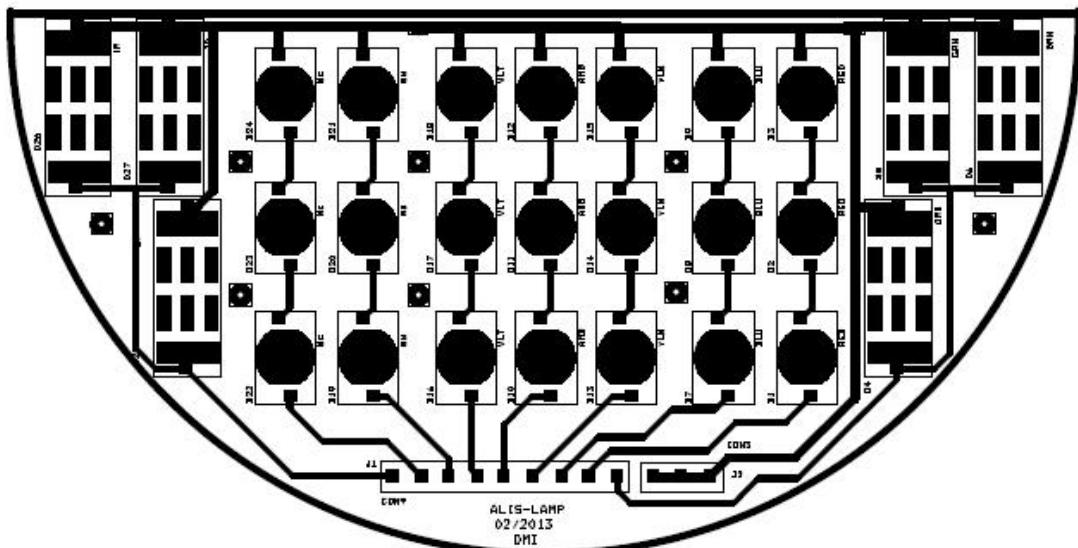
Широчината на радиатора трябва да бъде с максимален размер от 6 см, а дължината - 10 см. Тези параметри са дефинирани от разположението на светодиодите, монтажните отвори и захранващите шини.

Според така дефинираните размери за да се достигне необходимата радиаторна площ е необходимо присъединеният радиатор да има поне 4 ребра - всяко ребро е с охлаждаща повърхност от $6 \times 2,5 \times 2 = 30 \text{ cm}^2$.

Препоръчително е ребрата на радиатора да са разположени вертикално за по-добър конвекционен топлообмен.

6.4.3 Печатна платка

На фигура 21 е показан дизайна на печатната платка за модела на осветителното тяло.



DRILL CHART				
SYM	DIAM	TOL	QTY	NOTE
x	3.499 mm		B	NON-PLATED
	TOTAL		B	

фигура 21 - Модел на осветително тяло

7 Заключение - Анотация

Съвременните тенденции за намаляване на енергоразхода и увеличаване работоспособността на осветителните системи поставят основата на все по-голямо търсене на икономични осветителни системи способни да работят както в конвенционални сгради, така и в съвременни 'зелени' жилища. В комбинация с напредването на технологията и развитието на съвременното технологично общество, се полага идеята за развитието на една по-динамична домашна атмосфера. Хората все по-често искат да си създадат специфична атмосфера в дома си, която да е в унисон с тяхното собствено психологическо и емоционално състояние. Това от своя страна води до развитие и усъвършенстване на системи като представената.

Подобни и многократно по-сложни системи могат да бъдат изградени на базата на съвременните системи върху чип. Такива системи лесно биха управлявали не само осветлението, а цялата електронно управляема техника в къщата. Важно обаче е да се отбележи, че реализирането на отделни функционални системи изпълняващи ограничен набор от специализирани функции могат да бъдат ефективно разработени с помощта на относително прости апаратни и програмни средства. Реализирането на отделните функции посредством отделни апаратни решения осигурява автономност на отделните звена - те имат възможност да бъдат управлявани от централизирана система, но наличието на такава не е задължително условие за тяхното нормално функциониране и пропадането ѝ или това на някоя от останалите системи няма да доведе до неизправност на даденото звено.

7.1 Насоки за бъдещо развитие:

В тази секция са представени някои идеи за бъдещо развитие на системата:

Подходящо развитие на системата е интегрирането на генетични алгоритми в управляващият програмен код. Това ще даде възможност

на системата да се самообучава в зависимост от конкретните предпочтания на потребителя.

Разработката на програмни графични интерфейси за управление на системата посредством мобилни устройства (смартфони, таблети) и/или управление през уеб-браузер.

Серийният интерфейс може да се модифицира за предаване на данни към радио-комуникационен модул за безжична комуникация с устройства.

8 Използвана Литература

C - Практически самоучител - Хърбърт Шилдт - СофтПрес - София 2001г.

Embedded Microprocessor Systems Real World Design - Stuart R. Ball - 2002г

Complete PCB design using OrCAD capture and layout - Kraig Mitzner -
2007г.

MSP430G2x53, MSP430G2x13 Mixed Signal Microcontroller (Rev. H) -

<http://www.ti.com/lit/gpn/msp430g2553>

MSP430x2xx Family User's Guide (Rev. I) -

<http://www.ti.com/litv/pdf/slau144i>

I²C стандарт - http://www.nxp.com/documents/user_manual/UM10204.pdf

9 Приложения

- 1 Списък на елементите в системата
- 2 Принципна схема на системата
- 3 DVD с каталожните данни на елементите

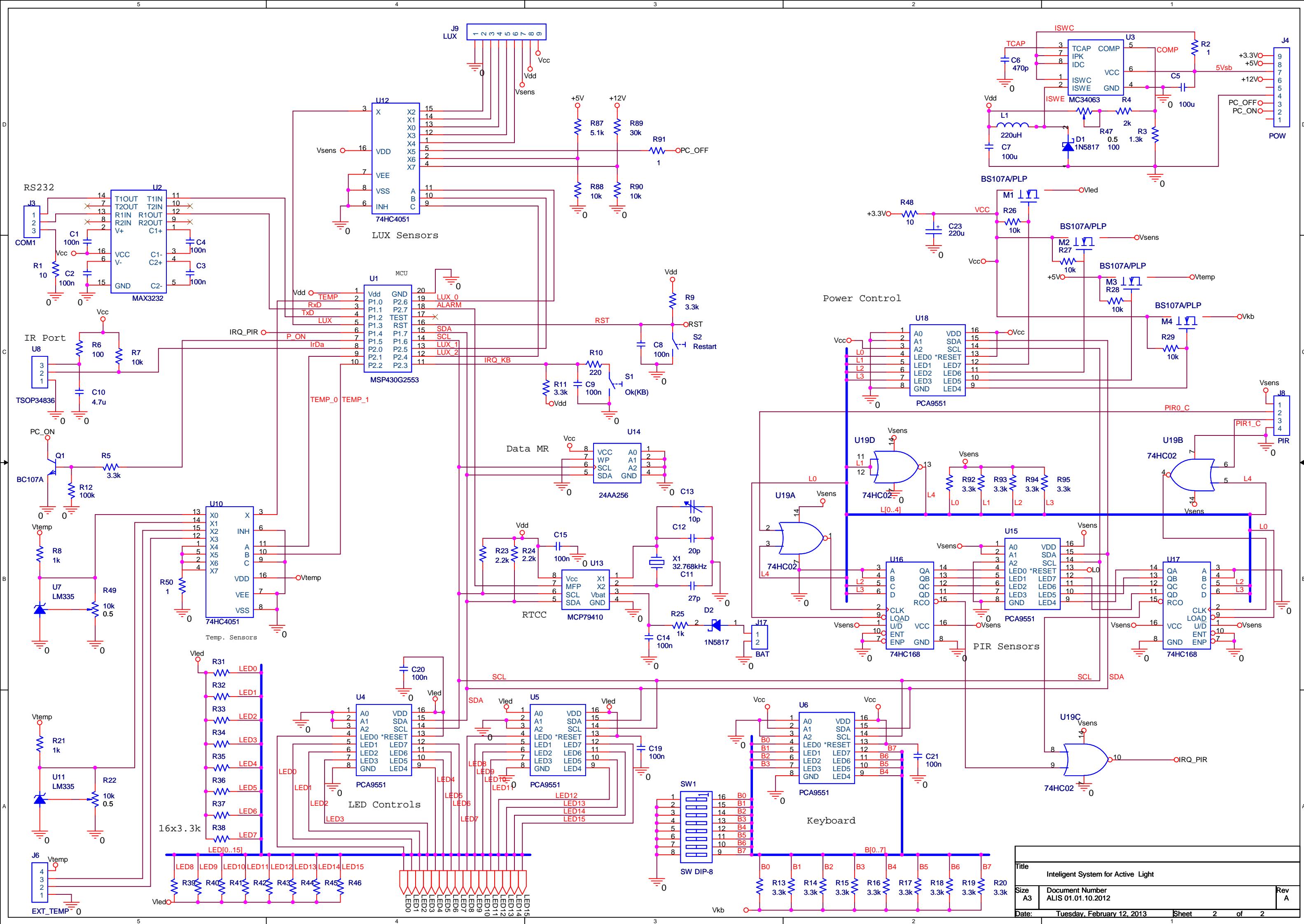
Списък на елементите

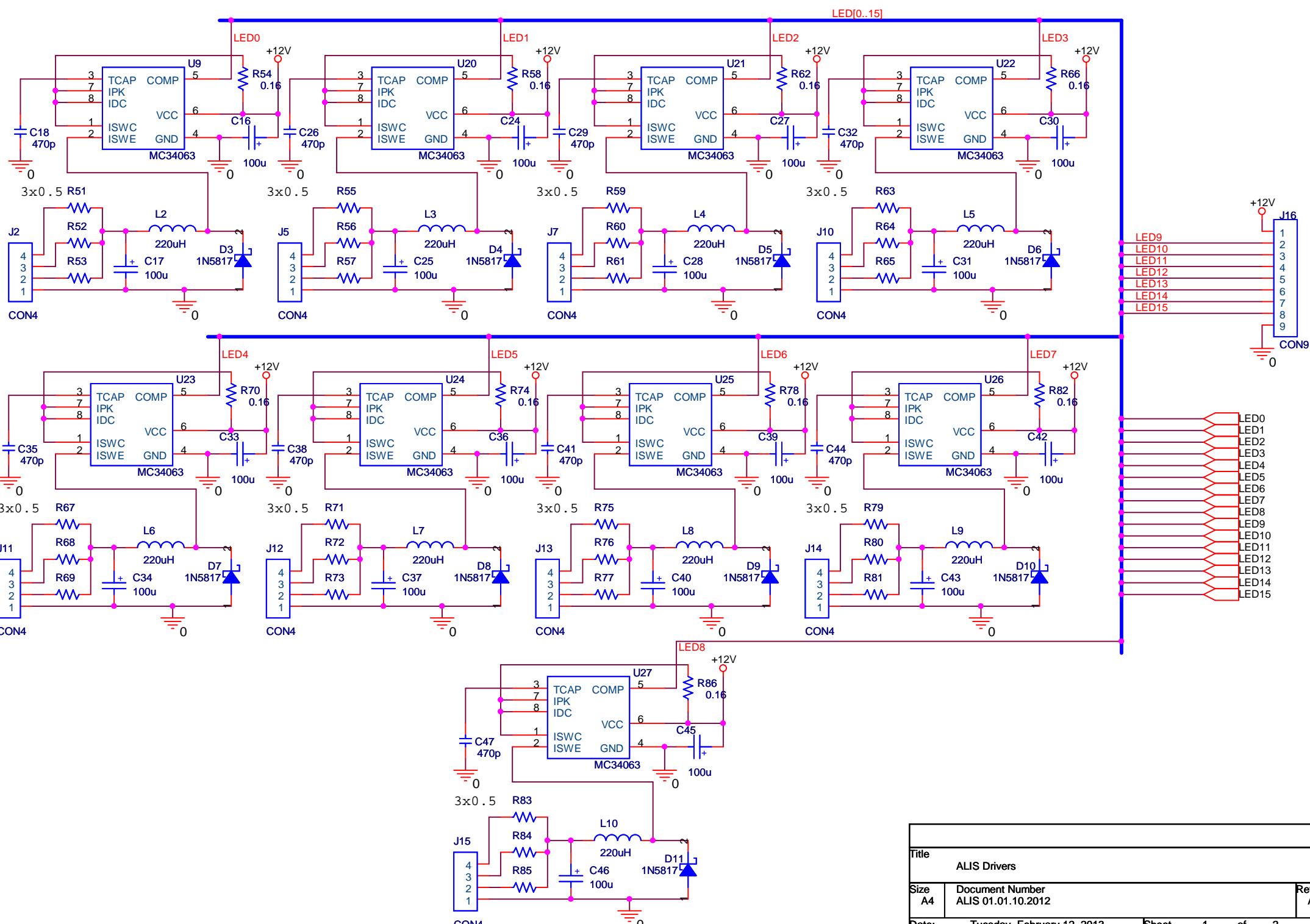
Списък на елементите

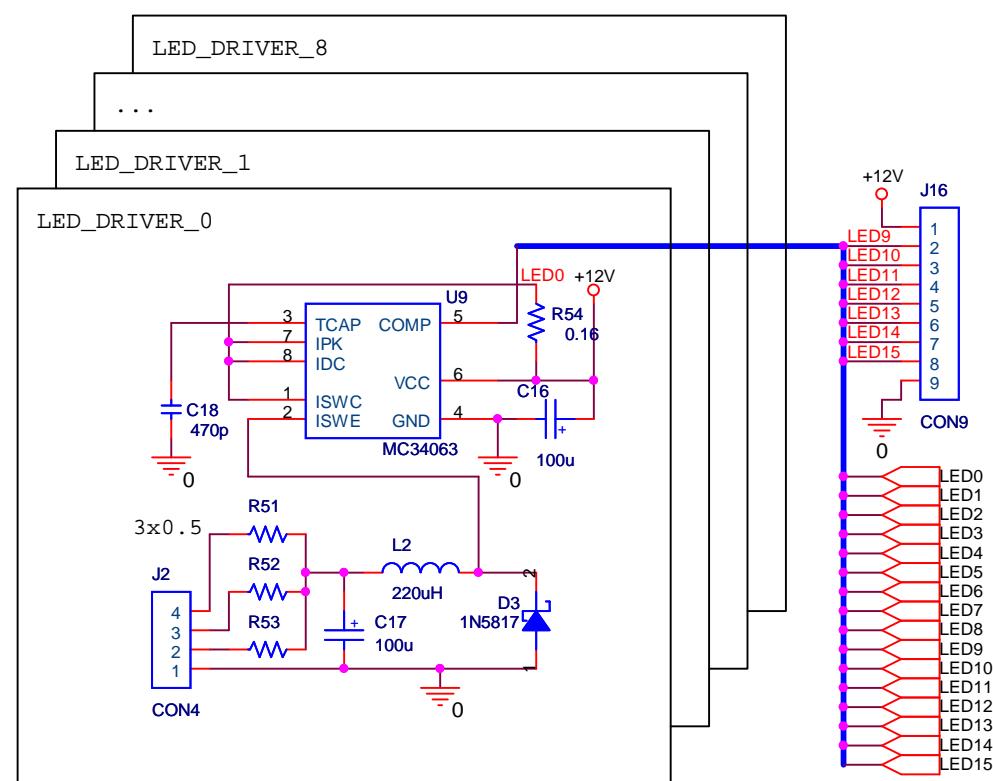
Означение в схемата	Наименование и означение	Кол.	Забележка
R4	Резистор постоянен 2kΩ±5%/0.25W	1	
R5,R9,R11,R13,R14, R15,R16,R17,R18,R19, R20,R31,R32,R33,R34, R35,R36,R37,R38,R39, R40,R41,R42,R43,R44, R45,R46,R92,R93,R94, R95	Резистор постоянен 3.3kΩ±5%/0.25W	31	
R6,R47	Резистор постоянен 100Ω±5%/0.25W	2	
R7,R26,R27,R28 ,R29,R88,R90	Резистор постоянен 10kΩ±5%/0.25W	7	
R22,R49	Резистор постоянен 10kΩ±5%/0.25W	2	
R8,R21,R25	Резистор постоянен 1kΩ±5%/0.25W	3	
R10	Резистор постоянен 220Ω±5%/0.25W	1	
R12	Резистор постоянен 100kΩ±5%/0.25W	1	
R23,R24	Резистор постоянен 2.2kΩ±5%/0.25W	2	
R51,R52,R53,R55,R56, R57,R59,R60,R61,R63, R64,R65,R67,R68,R69, R71,R72,R73,R75,R76, R77,R79,R80,R81,R83, R84,R85	Резистор постоянен 0.5Ω±5%/0.25W	27	
R54,R58,R62,R66, R70,R74,R78,R82, R86	Резистор постоянен 0.16Ω±5%/0.25W	9	
R87	Резистор постоянен 5.1kΩ±5%/0.25W	1	
R89	Резистор постоянен 30kΩ±5%/0.25W	1	
SW1	Ключ електромеханичен DIP-8р	1	
S1, S2	Микробутон DTS63K	2	
U1	Интегрална схема MSP430G2553	1	
U2	Интегрална схема MAX3232	1	
U3,U9,U20,U21, U22,U23,U24,U25, U26,U27	Интегрална схема MC34063	10	

Списък на елементите

Означение в схемата	Наименование и означение	Кол.	Забележка
U4,U5,U6,U15, U18	Интегрална схема PCA9551	5	
U7,U11	Интегрална схема LM335	2	
U8	Интегрална схема TSOP34836	1	
U10,U12	Интегрална схема 74HC4051	2	
U13	Интегрална схема MCP79410	1	
U14	Интегрална схема 24AA256	1	
U16,U17	Интегрална схема 74HC168	2	
U19	Интегрална схема 74HC02	1	
X1	Кварцов резонатор 32.768kHz	1	
ТУ-София	Разработил Инж. Илиев	Наименование, доп.наименование	ALIS 01.000.09.2012
	Одобрил Доц. Димитров	Интелигентна система за активно осветление	Изм. Дата на изд. 12.02.2013
		Език BG	Лист 3/3







Title ALIS Drivers	
Size A4	Document Number ALIS 01.01.10.2012
Date: Saturday, February 16, 2013	Sheet 1 of 2