

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет энергетический

Кафедра автоматизации производственных процессов и электротехники

Направление подготовки: 15.03.04 «Автоматизация технологических
процессов и производств»

Направленность (профиль) программы «Автоматизация технологических
процессов и производств в энергетике»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой
_____ А.А. Остапенко
(подпись)

«___» _____ 2017 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Автоматическая система определения положения солнца для позицио-
нирования солнечной батареи

Исполнитель

студент группы 341 – зсб _____ С.Н. Меркурьев
(подпись, дата)

Руководитель

доцент, канд. техн. наук _____ А. Н. Рыбалев
(подпись, дата)

Нормоконтроль

старший преподаватель _____ О. В. Скрипко
(подпись, дата)

Благовещенск 2017

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет энергетический

Кафедра автоматизации производственных процессов и электротехники

Направление подготовки: 15.03.04 «Автоматизация технологических процессов и производств»

Направленность (профиль) программы «Автоматизация технологических процессов и производств в энергетике»

УТВЕРЖДАЮ

Заведующий кафедрой

_____ А.А. Остапенко
(подпись)

«___» _____ 2017 г.

З А Д А Н И Е

К выпускной квалификационной работе студента:

гр. 341-зсб Меркурьева Сергея Николаевича

1. Тема выпускной квалификационной работы: Автоматическая система определения положения солнца для позиционирования солнечной батареи.

2. Срок сдачи студентом законченного проекта 08.02.2017

3. Исходные данные к выпускной квалификационной работе:

На основе микроконтроллера и лабораторной солнечной энергоустановки разработать устройство, выполняющее следующие функции и возможности:

- автоматическое определение положения солнца;
- выработку задания следящим системам лабораторной установки;
- предоставление информации об уровне освещённости;
- сопряжения системы с аппаратными программными комплексами лабораторной солнечной энергоустановки;
- расширения за счет размещения нескольких устройств.

4. Содержание выпускной квалификационной работы:

- 1) Описание объекта автоматизации и постановка задачи.
 - 2) Разработка функциональной схемы.
 - 3) Выбор оборудования.
 - 4) Изготовление и сборка системы.
 - 5) Разработка программной части.
5. Перечень материалов приложения:
- 1) Техническое задание на разработку автоматической системы
 - 2) Исходный код программы микроконтроллера

6. Дата выдачи задания: 13.04.2016

Руководитель выпускной квалификационной работы: Рыбалев Андрей Николаевич, доцент, канд. тех. наук.

Задание принял к исполнению (дата) 13.04.2016
(подпись студента)

РЕФЕРАТ

Выпускная квалификационная работа содержит 89 страниц, 37 рисунков, 3 приложения, 5 таблиц, 25 источников.

АВТОМАТИЗАЦИЯ, ДАТЧИКИ, КОНТРОЛЛЕР, ПЛК-73, ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ, ОСВЕЩЕННОСТЬ, СОЛНЕЧНАЯ ЭНЕРГЕТИКА, ФОТОРЕЗИСТОР, ARDUINO NANO.

В выпускной квалификационной работе спроектирована и реализована система автоматическая система определения положения солнца для позиционирования солнечной батареи. Данная система относится к классу неподвижных систем определения положения солнца, и позволяет без дополнительных движений приводов быстро и точно определить координаты солнца. Система может использоваться как для промышленной эксплуатации, так и в научных целях.

Основой методологии разработки стала теоретическая база знаний, полученная в процессе обучения, а также практические навыки, необходимые для реализации проекта.

На основании полученных знаний разработана функциональная схема автоматизации, электрическая принципиальная схема. Изготовлена автоматическая система и разработано программное обеспечение к ней. Система протестирована на работоспособность.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 ОБЪЕКТ АВТОМАТИЗАЦИИ И ПОСТАНОВКА ЗАДАЧИ	10
1.1 Лабораторная солнечная энергоустановка	10
1.2 Структура системы управления солнечной энергоустановки	14
1.3 Постановка задачи	15
2 РАЗРАБОТКА АППАРАТНОЙ ЧАСТИ	16
2.1 Функциональная схема объекта автоматизации	16
2.2 Выбор основы изделия	17
2.3 Выбор фоточувствительного элемента	18
2.4 Выбор вычислительного устройства	22
2.5 Выбор элемента индикации	30
2.6 Выбор мультиплексора	31
2.7 Дополнительные конструктивные элементы	35
2.8 Подготовка основания	39
2.9 Разработка печатной платы	42
2.10 Изготовление печатной платы	44
2.11 Монтаж элементов на печатную плату	48
2.12 Установка и подключение дисплея	50
2.13 Окончательная сборка	52
3 РАЗРАБОТКА ПРОГРАММНОЙ ЧАСТИ	55
4 ПЕРСПЕКТИВЫ РАЗВИТИЯ И МОДЕРНИЗАЦИИ СИСТЕМЫ	65
ЗАКЛЮЧЕНИЕ	68
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	69
ПРИЛОЖЕНИЕ А	72
ПРИЛОЖЕНИЕ Б	83
ПРИЛОЖЕНИЕ В	86

ВВЕДЕНИЕ

Солнечная энергетика – направление альтернативной энергетики, основанное на непосредственном использовании солнечного излучения для получения энергии в каком-либо виде. Солнечная энергетика использует возобновляемые источники энергии и является «экологически чистой», то есть не производящей вредных отходов во время активной фазы использования. Одна из основных тенденций современного мира – увеличение производства электроэнергии с использованием альтернативных источников, в том числе с использованием солнечной энергии. Так, в 1985 году все установленные мощности солнечной генерации мира составляли 0,021 ГВт, в 2012 году общая мощность мировых гелиоэнергетических установок превысила 100 ГВт, а в 2014 году общая мощность таких установок оценивалась уже в 186 ГВт, что составило 0,8% от мирового производства всей электроэнергии. Ежегодный прирост генерации солнечной энергии за последние пять лет составлял в среднем около 50 %. Лидером по установленной мощности является Евросоюз, среди отдельных стран – Китай. Как полагают эксперты Международного энергетического агентства (IEA), солнечная энергетика к 2050 году, при соответствующем уровне распространения передовых технологий, будет вырабатывать около 20–25 % всего необходимого электричества, и это обеспечит сокращение выбросов углекислого газа на бмлрд. тонн ежегодно [6].

Солнечная энергия имеет огромный потенциал, а ее запасы превышают все существующие сырьевые ресурсы, но увеличение мощностей солнечной генерации происходит достаточно медленно. Основными сдерживающими факторами являются:

- зависимость от погоды и времени суток;
- сезонность в средних широтах и нерентабельность в высоких широтах;
- необходимость аккумуляции энергии;

- высокая стоимость конструкции, связанная с применением редких элементов;
- необходимость периодической очистки отражающей/поглощающей поверхности от загрязнения;
- нагрев атмосферы над электростанцией;
- необходимость использования больших площадей;
- несмотря на экологическую чистоту получаемой энергии, сами фотоэлементы могут содержать ядовитые вещества, что ставит под вопрос экологическую чистоту производства и утилизации батарей.

В связи с этим солнечная генерация в значительных масштабах рассматривается сегодня только в виде способа экономии небольшой части традиционного ископаемого топлива в дневное время. Солнечная энергетика пока не в силах полностью взять на себя всю нагрузку, и уменьшить число АЭС, угольных, газовых и гидроэлектростанций [6].

Однако развитие солнечной энергетики является перспективным направлением, так как [1]:

- стоимость ископаемого топлива неуклонно растет по мере уменьшения его запасов;
- разумная государственная политика делает использование солнечных электростанций выгоднее;
- прогресс не стоит на месте, КПД солнечных электростанций повышается, разрабатываются новые технологии в генерировании и аккумулировании электроэнергии.

Следовательно, необходимо проводить научные исследования, позволяющие снизить влияние негативных факторов и увеличить выгоду от использования солнечной энергии. В настоящее время наиболее активно проводятся исследования по нескольким направлениям, рассмотренным ниже.

Исследования, направленные на увеличение эффективности фотоэлементов и модулей. Использование новых материалов и технологий при изготовле-

нии солнечных элементов позволяет снизить нагрев солнечных элементов и повысить их КПД. Повышение КПД также достигается в случае использования систем слежения и систем с изменяемым углом наклона фотоэлектрических модулей, так как эффективность солнечных элементов максимальна при попадании на них прямых солнечных лучей. Однако системы слежения требуют дополнительных затрат на их приобретение, а ввиду постоянных атмосферных воздействий механических операций возникает необходимость регулярного обслуживания и ремонтов.

Исследования, направленные на эффективное распределение выработанной энергии. На данный момент проблема решается использованием эффективных электрических аккумуляторов, преобразованием в другие виды энергии, например, постройкой гидроаккумулирующих станции, а также созданием единых энергетических систем, которые перераспределяют вырабатываемую и потребляемую мощность.

Исследования, направленные на снижение стоимости обслуживания. Поверхность фотоэлементов и зеркал нужно очищать от пыли и других загрязнений. В случае крупных фотоэлектрических станций, при их площади в несколько квадратных километров это может вызвать затруднения, но применение отполированного стекла на современных солнечных батареях практически решает эту проблему. В случае использования систем слежения также необходимо искать способы снижения обслуживания приводов.

Исследования, направленные на снижение экологических рисков. Из-за ограничений в преобразовании спектра в полезную энергию требуется использование больших площадей земли под электростанции. Солнечные концентраторы вызывают большие по площади затенения земель, что приводит к экологическим и, возможно, климатическим изменениям. Проблема нахождения больших площадей земли частично может быть решена использованием солнечных электростанций пригодных как для наземного, так и для морского и высотного базирования. Вышедшие из строя фотоэлементы могут содержать компоненты, которые недопустимо утилизировать обычным способом, необходи-

модополнительное расширение индустрии по их утилизации[6].

Из вышесказанного можно сделать вывод, что в промышленном масштабе актуальна задача повышения КПД солнечных батарей, – это повысит их эффективность на квадратный метр и, следовательно, уменьшит площадь их размещения; также необходимо уменьшать стоимость обслуживания таких систем.

Цель данной выпускной квалификационной работы– автоматизировать процесс наведения солнечной установки; разработать систему предоставляющую информацию, на основе которой будет возможно вырабатывать оптимальные решения о позиционировании солнечной батареи, в том числе с учетом интенсивности светового потока. Такая система позволит снизить стоимость обслуживания систем слежения, за счет уменьшения количества движения приводов, и в то же время сохранить их преимущество перед простым наращиванием площади батарей.

1 ОБЪЕКТ АВТОМАТИЗАЦИИ И ПОСТАНОВКА ЗАДАЧИ

1.1 Лабораторная солнечная энергоустановка

Объектом автоматизации является процесс позиционирования солнечной батареи входящей в состав лабораторной солнечной энергоустановки, общий вид которой представлен на рисунке 1.1, входит:

1. солнечная батарея – фотоэлемент (10Вт);
2. электроприводы;
3. каркас;
4. системы измерения и контроля положения;
5. шкаф управления с ПЛК 73.



Рисунок 1.1 – Общий вид установки

Лицевая панель и компоновка приборов шкафа управления показаны на рисунке 1.2.



Рисунок 1.2 – Шкаф управления

Преобразование энергии в фотоэлементе основано на испускании электронов веществом под действием света или любого другого электромагнитного излучения. Фотоэффект возникает в неоднородных полупроводниковых структурах при воздействии на них солнечного излучения. Неоднородность структуры фотоэлемента может быть получена легированием одного и того же полупроводника различными примесями, путём соединения различных полупроводников с неодинаковой шириной запрещённой зоны, или же за счёт изменения химического состава полупроводника, приводящего к появлению градиента ширины запрещённой зоны. Возможны также различные комбинации перечисленных способов [3]. Солнечная батарея построена с использованием защитного стекла, которое пропускает ультрафиолетовые лучи.

Корпус изготовлен из алюминиевого каркаса, что обеспечивает достаточную прочность и легкость установки, а также более низкую стоимость монтажа конструкции. Он обладает такими преимуществами, как:

- простая установка;
- использование на открытом воздухе;
- устойчивость к коррозии и влажности;
- защита от воды и пыли.

Рабочими органами системы позиционирования солнечной батареи явля-

ются электроприводы линейного перемещения постоянного тока напряжения 12В производства ECO-WORTHY (Китай), обеспечивающие повороты батареи вокруг вертикальной (по азимуту) и горизонтальной (по высоте) осей. Приводы снабжены встроенными концевыми выключателями. Полный ход первого привода обеспечивает изменение положения батареи по азимуту на 120° . Вторым приводом на полный ход не задействован и выполняет поворот по высоте от 0 до 90 градусов. При этом в положении 0° отключение привода осуществляется встроенным концевым выключателем, а в положении 90° – системой управления, так как дальнейшее движение ограничивается элементами конструкции и привод начинает работать на упор.

Для измерения положения батареи по высоте и азимуту применены бесконтактные индуктивные датчики LA12M – 5P1. Они формируют импульсы, количество которых пропорционально значениям соответствующих углов. Датчики положения и концевые выключатели реализованы в конструкции, показанной на рисунке 1.3.

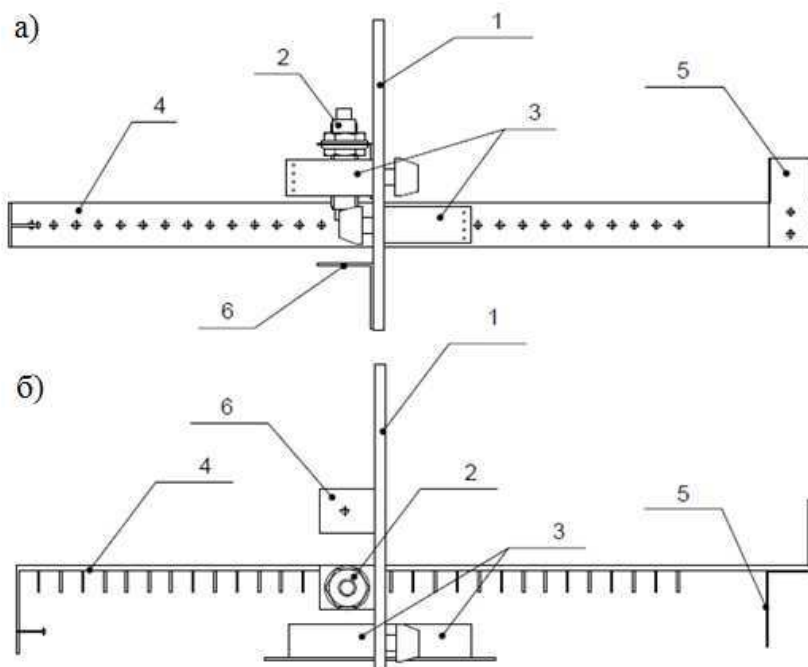


Рисунок 1.3 – Конструкция датчика положения и концевых выключателей:

а) вид сбоку; б) вид сверху

Через фигурное отверстие в алюминиевой пластине 1 движется алюминиевая рейка 4, увлекаемая штоком привода, с концом которой она связана с

одной стороны. На пластине установлены индуктивный датчик 2 и концевые выключатели 3. В рейку с шагом 1 см вкручены металлические шпильки диаметром 3 мм. Датчик срабатывает при прохождении очередной шпильки непосредственно под ним. Свободный конец рейки изогнут на угол 90° , на изгибе располагается винт, воздействующий на один из концевых выключателей. На другой концевой выключатель воздействует деталь в форме уголка, прикрепленная к концу рейки, связанному со штоком привода.

Количество металлических шпилек на рейке датчиков поворота вокруг вертикальной и горизонтальной осей равно 30 и 19 соответственно, что обеспечивает позиционирование с абсолютной погрешностью не более $\pm 6^\circ$ по азимуту и $4,8^\circ$ по высоте.

В качестве контроллера использован ПЛК 73 фирмы ОВЕН. Функциональные возможности ПЛК 73 хорошо подходят для управления установкой. В частности, контроллер снабжен дисплеем и клавиатурой, которые позволяют вводить необходимые параметры и непосредственно наблюдать результаты работы системы без подключения к ПК. Особенностью использования контроллера в системе является задействованный аналоговый вход, ввиду нехватки дискретных входов. В контроллере задействованы [18]:

- 8 дискретных входов (2 – для подключения индуктивных датчиков положения солнечной батареи, 2 – для определения направления движения приводов в ручном режиме и 4 – для сигналов концевых выключателей);
- 4 дискретных выхода для управления реверсивными электроприводами;
- 3 аналоговых входа для определения режима управления, измерения напряжения и тока солнечной батареи.

1.2 Структура системы управления солнечной энергоустановки

Структура системы управления показана на рисунке 1.4.

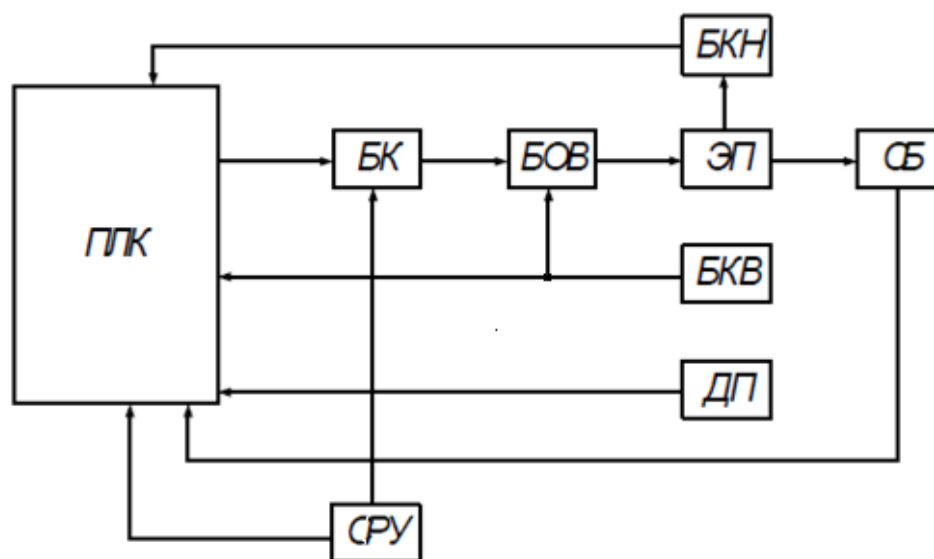


Рисунок 1.4 – Структурная схема системы управления

На схеме присутствуют следующие обозначения[18]:

СБ – солнечная батарея;

ЭП – электроприводы;

БОВ – блок ограничения высоты. Автоматически отключает электропривод поворота по высоте при достижении горизонтального положения батареи (высоты, равной 90°);

БК – блок коммутации. Осуществляет коммутацию электроприводов в автоматическом и ручном режимах;

СРУ – система ручного управления. Состоит из переключателя режима ручной/автомат и четырьмя кнопками ручного управления приводами;

ПЛК – программируемый логический контроллер. Измеряет и индицирует текущее положение солнечной батареи, в автоматическом режиме управляет процессом позиционирования, измеряет и индицирует напряжение и ток солнечной батареи;

БКН – блок контроля направления. Определяет и сообщает ПЛК направление движения приводов в ручном режиме для корректной работы программного элемента, ответственного за измерение положения батареи;

БКВ – блок конечных выключателей. Сигнализирует ПЛК о достижении электроприводами конечных положений и формирует сигнал запрета движения для блока ограничения высоты;

ДП – датчики положения импульсного типа.

1.3 Постановка задачи

В соответствии с техническим заданием, приведенным в приложении А, требуется разработать систему автоматического определения положения солнца для позиционирования солнечной батареи, которая будет вырабатывать задание следящим системам лабораторной установки, и осуществить сопряжение этой системы с аппаратными программными комплексами этой установки.

В системе необходимо предусмотреть решение, позволяющее предоставлять информацию об уровне освещённости датчика в зависимости от его расположения по отношению к солнцу. Эта информация позволит проводить дальнейшие исследования и разрабатывать алгоритмы принятия оптимального решения о позиционировании солнечной батареи, в том числе с учетом интенсивности светового потока. Также необходимо разрабатывать подсистему с возможностью расширения за счет размещения её блоков измерения на удалении от центра управления и друг от друга.

2 РАЗРАБОТКА АППАРАТНОЙ ЧАСТИ

2.1 Функциональная схема объекта автоматизации

На функциональной схеме автоматизации изображена солнечная энергоустановка с необходимым для выполнения поставленной задачи блоком измерения освещенности и определения оптимального положения, а также щит управления, в составе которого находится ПЛК 73, и другие элементы цепи. Функциональная схема автоматизации представлена в приложении Б.

На схеме присутствуют следующие обозначения:

NS – переключатель электрических цепей;

H – ручное управление;

SB1-SB4 – кнопки управления электроприводами;

NS – устройство коммутации;

K1-K6 – промежуточные реле;

SA1 – переключатель управления ручное/автоматическое;

GS – концевые выключатели;

GE – датчики положения;

EE – электрический сигнал напряжения батареи;

M1, M2 – электропривод ECO-WORTHY

LE – группа фоточувствительных элементов;

EY – вычислительное устройство ArduinoNano;

EI – экран TFT;

AI – аналоговые входы ПЛК;

DI – дискретные входы ПЛК;

DO – дискретные выходы ПЛК;

RS-485 – асинхронный интерфейс.

Датчики GS встроены в корпус электроприводов, сигналы с них поступают на дискретные входы контроллера. Они предназначены для автоматического отключения электроприводов при достижении крайнего положения, чтобы избежать перегрузки. Датчики GE располагаются на электроприводах, а сигналы

с них поступают на дискретные входы ПЛК. ПЛК с помощью дискретных выходов может управлять приводами, посылая сигналы на переключатель, и если тот включен в положение автоматического управления, то сигналы идут на устройства коммутации.

Группа фоточувствительных элементов LE размещена определенным образом на четверти сферы и преобразует сигналы освещенности, в зависимости от угла падения солнечных лучей на эти элементы, в электрические сигналы разной мощности, которые передаются на вычислительное устройство EY. Устройство EY вычисляет оптимальный угол и передает его на ПЛК 73 посредством асинхронного интерфейса RS-485 [5], а также отображает карту освещенности на экране EI.

2.2 Выбор основы изделия

В качестве основы изделия была выбрана половина пенопластовой полусферы диаметром 16 сантиметров.



Рисунок 2.1 – Пенопластовая сфера диаметром 16 сантиметров

Преимущества пенопласта:

- незначительный вес;
- простота обработки и монтажа;

- устойчивость к перманентному воздействию влаги;
- устойчивость к гниению и грибку.

Недостатки:

- легко воспламеняется;
- пенопласт подвержен механическим воздействиям;
- пенопласт не выдерживает температуру свыше 90°C;

Сфера из пенопласта – удобная основа для изделия, легко поддается обработке, декорируется акриловыми красками на водной основе.

2.3 Выбор фоточувствительного элемента

Фоторезистор – это резистор, управляемый световым потоком Φ . Действие его основано на внутреннем фотоэффекте [2]. Фоторезистор изменяет величину своего сопротивления при облучении светом. Не имеет р-п перехода, поэтому обладает одинаковой проводимостью независимо от направления протекания тока [7]. Условное графическое обозначение фоторезистора изображено на рисунке 2.2. Два внешних металлических контакта фоторезистора идут через керамический материал основания к светочувствительной пленке, которая по своей геометрии и свойству материала определяет электрические свойства сопротивления. Так же как и человеческий глаз, фоторезистор чувствителен к определенному диапазону длины волны света. При выборе фотоэлемента приходится с этим считаться, поскольку в противном случае он может совсем не отреагировать на выбранный источник света.

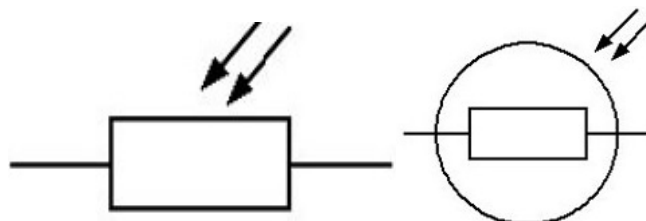


Рисунок 2.2 – Условное графическое обозначение фоторезистора

Для изготовления фоторезисторов используют полупроводниковые материалы с шириной запрещенной зоны, оптимальной для решаемой задачи. Так, для регистрации видимого света используются фоторезисторы из селенида и сульфида кадмия. Для регистрации инфракрасного излучения используются гер-

маний, чистый или легированный примесями, кремний, сульфид свинца и другие соединения [7]. В процессе изготовления полупроводник наносят в виде тонкого слоя на стеклянную или кварцевую подложку или вырезают в виде тонкой пластинки из монокристалла. Слой или пластинку полупроводника снабжают двумя электродами и помещают в защитный корпус.

Фотодиод – это германиевый или кремниевый диод, обратный ток р-п-перехода которого зависит от освещённости перехода. Работа его основана на вентильном фотоэффекте. Фотодиоды могут работать с внешним источником и без него. При работе без внешнего источника питания фотогальваническая ЭДС достигает 0,1 – 0,2В и более [2]. Фотодиод как приёмник оптического излучения преобразует попавший на его фоточувствительную область свет в электрический заряд за счёт процессов в р-п-переходе. Условное графическое обозначение фотодиода изображено на рисунке 2.3.

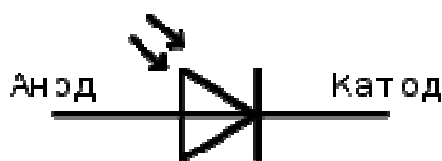


Рисунок 2.3 – условное графическое обозначение фотодиода

Быстродействие фотодиода определяется скоростью разделения носителей полем р-п-перехода и ёмкостью р-п-перехода. Фотодиод, работа которого основана на разделении электронов и дырок в р- и п-области, за счёт чего образуется заряд и ЭДС, называется солнечным элементом. Кроме р-п фотодиодов, существуют и р-і-п фотодиоды, в которых между слоями р и п находится слой нелегированного полупроводника і. р-п и р-і-п фотодиоды только преобразуют свет в электрический ток, но не усиливают его, в отличие от фототранзисторов [8].

Фотодиод может работать в двух режимах:

- фотогальванический режим – без внешнего напряжения;
- фотодиодный режим – с внешним обратным напряжением.

Особенности:

- простота технологии изготовления и структуры;
- сочетание быстродействия и фоточувствительности;
- малое сопротивление базы;
- малая инерционность.

Фототранзистор—оптоэлектронныйполупроводниковыйприбор, вариант биполярного транзистора. Отличается от классического варианта тем, что область базы доступна для светового облучения, за счёт чего появляется возможность управлять усилением электрического тока с помощью оптического излучения. Условное графическое обозначение фототранзистора изображено на рисунке 2.4. Недостатком фототранзисторов является большая инерционность, что ограничивает их применение в качестве быстродействующих выключателей[9].

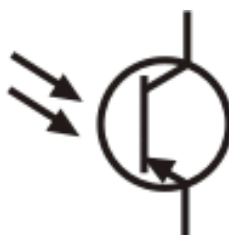


Рисунок 2.4 – условное графическое обозначение фототранзистора

При выборе фоточувствительного элемента выбор сделан в пользу фоторезисторов, поскольку для системы автоматического определения положения солнца для позиционирования солнечной батареи нет необходимости в высоком быстродействии. Кроме того современные фоторезисторы обеспечивают достаточную точность и устойчивость характеристик, в то время как выбор фотодиода или фототранзистора хоть и обеспечивает повышенную точность, требует усложнения конструкции системы и вызывает её удорожание. Выбор конкретной модели фоторезистора делался из серии GL55, т.к. фоторезисторы данной серии распространены на рынке, недороги и имеют хорошие характеристики для данного проекта.

- максимальное напряжение при 25°C: 150В;
- рассеиваемая мощность при 25°C: 100 мВт;

- пик максимальной чувствительности в спектре при 25°C: 540 нм;
- допустимый диапазон температур –от минус 30°Cдо плюс70°C;
- время отклика: повышение 20 мс, понижение 30 мс [24];

Поскольку диапазон значимых измерений от 5000 до 100000 люкс, и коэффициента гамма 0,8, то можно приближенно рассчитать диапазон значимых сопротивлений по формуле (1.1):

$$\gamma = \frac{\log_{10} \frac{R_{10}}{R_{100}}}{\log_{10} \frac{100}{10}} = \log_{10} \frac{R_{10}}{R_{100}} \quad (1.1)$$

где R10 – известное сопротивление при освещении 10 люкс, R100 – неизвестное сопротивление при освещении 100 люкс.

Проведем приближительные расчеты, для модели GL5539, изображенной на рисунке 2.5, с характеристиками:

- сопротивление в темноте (0 люкс): 5,0 МОм;
- световое сопротивление (10 люкс): 50..100 кОм;
- гамма 0,8;
- размеры 4,3x5,1x2,1 мм (ширина, длина, высота).

$$R_{100} = \frac{R_{10}}{10^\gamma} = 0,1585 \cdot R_{10} \quad (1.2)$$

$$R_{100} = 15850 \text{ Ом} \quad (1.3)$$

$$R_{1000} = 0,1585 \cdot R_{100} = 2512 \text{ Ом} \quad (1.4)$$

$$R_{10000} = 0,1585 \cdot R_{1000} = 398 \text{ Ом} \quad (1.5)$$

$$R_{100000} = 0,1585 \cdot R_{10000} = 63 \text{ Ом} \quad (1.6)$$

Поскольку, как мы увидим далее, логика используемого микроконтроллера 5В, а точность его АЦП 1024 значения, то шаг АЦП будет приблизительно 0,005 В. При выборе второго плеча делителя напряжения, например, на 220 Ом мы сможем достаточно точно снимать показания в диапазоне 1000..100000 люкс. Так как даже при грубом линейном расчете изменения сопротивления на один люкс:

$$\text{дельта } 1000..10000 = \frac{5}{\left(\frac{2512+220-398}{10}\right)} = 0,02 \text{ В} \quad (1.7),$$

$$\text{дельта } 10000..100000 = \frac{5}{\left(\frac{398+220-63}{10}\right)} = 0,01 \text{ В} \quad (1.8),$$

Полученная разность потенциалов вполне удовлетворяет заданной точности в 0,005 В.



Рисунок 2.5 – Фоторезистор GL5539

2.4 Выбор вычислительного устройства

Arduino – торговая марка аппаратно-программных средств, разрабатываемая для построения простых систем автоматики и робототехники. Программная часть состоит из бесплатной программной оболочки для написания программ, их компиляции и программирования аппаратуры. Аппаратная часть представляет собой набор смонтированных печатных плат, продающихся как официальным производителем, так и сторонними производителями. Полностью открытая архитектура системы позволяет свободно копировать или дополнять линейку продукции Arduino[11]. Платформа Arduino пользуется огромной популярностью во всем мире благодаря удобству и простоте языка программирования, а также открытой архитектуре и программному коду. Устройство программируется через USB без использования программаторов. Устройства на базе Arduino могут получать информацию об окружающей среде посредством различных датчиков, а также могут управлять различными исполнительными устройствами. Под торговой маркой Arduino выпускается несколько плат с микроконтроллером и платы расширения. Большинство плат с микроконтрол-

лером снабжены минимально необходимым набором обвязки для нормальной работы микроконтроллера, это такие элементы как стабилизатор питания, кварцевый резонатор, цепочки сброса ит.п. [21].

Конструктивное исполнение. Arduino и Arduino-совместимые платы спроектированы таким образом, чтобы их можно было при необходимости расширять, добавляя в устройство новые компоненты. Эти платы расширений подключаются к Arduino посредством установленных на них штыревых разъёмов. Существует ряд плат, допускающих конструктивно жесткое соединение процессорной платы и плат расширения в стопку через штыревые линейки. Кроме того, выпускаются платы уменьшенных габаритов и специальных конструктивов для задач робототехники. Независимыми производителями также выпускается большая гамма всевозможных датчиков и исполнительных устройств, в той или иной степени совместимых с Arduino. Разработчик выбирает метод установки и механической защиты плат самостоятельно. Сторонними производителями выпускаются наборы робототехнической электромеханики, ориентированной на работу совместно с платами Arduino [11].

Микроконтроллер. В линейке устройств Arduino в основном применяются микроконтроллеры Atmel AVR ATmega328, ATmega168, ATmega2560, ATmega32U4, ATTiny85 с частотой тактирования 16 или 8 МГц. Архитектура AVR используется в относительно низкопроизводительных встроенных системах. Один из главных факторов успеха в отрасли микроконтроллеров прост: разместите на микросхеме все, что только может понадобиться, и оформите ее в виде недорогой и компактной микросхемы с небольшим количеством разъемов. Интеграция многих функций позволяет применять микроконтроллер для решения многих задач, а малые размеры и компактность обеспечивают его использование в разных форм-факторах [19]. Микроконтроллеры для Arduino отличаются наличием предварительно прошитого в них загрузчика. С помощью этого загрузчика пользователь загружает свою программу в микроконтроллер без использования традиционных отдельных аппаратных программаторов. Загрузчик соединяется с компьютером через интерфейс USB или с помощью отдельного

переходника UART-USB. Поддержка загрузчика встроена в Arduino IDE и выполняется в один щелчок мыши. На случай затирания загрузчика или покупки микроконтроллера без загрузчика разработчики предоставляют возможность прошить загрузчик в микроконтроллер самостоятельно. Для этого в Arduino IDE встроена поддержка нескольких популярных дешевых программаторов, а большинство плат Arduino имеет штыревой разъем для внутрисхемного программирования[11].

Периферия. Порты ввода-вывода микроконтроллеров оформлены в виде штыревых линеек. Микроконтроллеры питаются от 5В или 3,3В, в зависимости от модели платы. Соответственно порты имеют такой же размах допустимых входных и выходных напряжений. Программисту доступны некоторые специальные возможности портов ввода-вывода микроконтроллеров, например широтно-импульсная модуляция, аналогово-цифровой преобразователь, интерфейсы UART, SPI, I2C. Количество и возможности портов ввода-вывода определяются конкретным вариантом микропроцессорной платы. Помимо портов на платах микроконтроллеров иногда устанавливается периферия в виде интерфейсов USB или Ethernet. Наиболее популярной внешней периферии на модулях расширения является: USB, проводной и беспроводной Ethernet, модуль GSM и другие беспроводные интерфейсы, USB Host, SD card и другие [20]. Сторонние производители выпускают широкую гамму датчиков и исполнительных устройств, подключаемых к Arduino. Например, гироскопы, компасы, гигрометры, термометры, индикаторы, клавиатуры ит.п.

Программирование ведется целиком через собственную программную оболочку, бесплатно доступную на сайте Arduino. В этой оболочке имеется текстовый редактор, менеджер проектов, препроцессор, компиляторы и инструменты для загрузки программы в микроконтроллер. Оболочка написана на Java на основе проекта Processing, работает под Windows, Mac OS X и Linux. Язык программирования Arduino является стандартным C++, с некоторыми особенностями, облегчающими написание программы. Используется компилятор AVR-GCC.

Программы, написанные для Arduino, называются наброски или иногда скетчи (от англ. sketch) и сохраняются в файлах с расширением `.ino`. Эти файлы перед компиляцией обрабатываются препроцессором Ардуино. Также существует возможность создавать и подключать к проекту стандартные файлы C++. Обязательную в C++ функцию `main()` препроцессор Arduino создает сам, вставляя туда необходимые действия. Программист должен написать две обязательные для Arduino функции `setup()` и `loop()`. Первая вызывается однократно при старте, вторая выполняется в бесконечном цикле. Arduino IDE не предлагает никаких настроек компилятора и минимизирует другие настройки, что упрощает начало работы для новичков и уменьшает риск возникновения проблем.

Загрузка программы в микроконтроллер. Закачка программы в микроконтроллер Arduino происходит с помощью специального загрузчика, все микроконтроллеры от Arduino продаются с этим загрузчиком. Загрузчик создан на основе Atmel AVR Application Note AN109. Загрузчик может работать через интерфейсы RS-232, USB или Ethernet в зависимости от состава периферии конкретной процессорной платы. В некоторых вариантах плат для программирования требуется отдельный переходник. Пользователь может самостоятельно запрограммировать загрузчик в чистый микроконтроллер, для этого поддерживается несколько типов популярных дешёвых программаторов [11].

При выборе вычислительного устройства практически нет конкурентов платам Arduino, поскольку платы Arduino сочетают в себе надежность, простоту разработки и невысокую стоимость, а также в интернете существует обширное сообщество разработчиков, использующих данные платы. Использование микроконтроллеров, вместо плат Arduino, усложняет изготовление прототипа изделия, и используется, как правило, в промышленных масштабах, а использование других плат, например Raspberry Pi или ПЛК обычно повышает стоимость конечного изделия за счет ненужных, для проекта, вычислительных мощностей и опций.

Для выбора конкретной модели аппаратной платформы рассмотрим наиболее распространенные подходящие модели [21]:

Due—новая плата на базе ARM микропроцессора 32bit Cortex-M3 ARM-SAM3U4E;

Leonardo — последняя версия платформы Arduino на ATmega32u4 микроконтроллере. Отличается разъемом micro-USB, по размерам совпадает с UNO.

Micro— новое компактное решение на базе ATmega32u4.

Uno— самая популярная версия базовой платформы Arduino USB. Uno имеет стандартный порт USB. Имеет новый чип ATmega8U2 для последовательного подключения по USB и новую, более удобную маркировку вход/выходов. Платформа может быть дополнена платами расширения, например, пользовательскими платами с различными функциями.

Duemilanove — является предпоследней версией базовой платформы Arduino USB. Подключение Duemilanove производится стандартным кабелем USB. После подключения она готова к использованию. Платформа может быть дополнена платами расширения, например, пользовательскими платами с различными функциями.

Nano — это компактная платформа, используемая как макет. Nano подключается к компьютеру при помощи кабеля USB mini-B.

Mega2560— новая версия платы серии Mega. Построена на базе Atmega2560 и с использованием чипа ATmega8U2 для последовательного соединения по USB порту.

Mini— самая маленькая платформа Arduino. Прекрасно работает как макетная модель, или, в проектах, где пространство является критическим параметром. Платформа подключается к компьютеру при помощи адаптера Mini USB.

Pro— платформа, разработанная для опытных пользователей, может являться частью большего проекта. Она дешевле, чем Diecimila и может питаться от аккумуляторной батареи, но в тоже время требует дополнительной сборки и компонентов.

Pro Mini— как и платформа Pro разработана для опытных пользователей, которым требуется низкая цена, меньшие размеры и дополнительная функцио-

нальность.

Большинство описанных плат имеют минимум периферии, однако для проекта важно, чтобы плата имела минимальные габаритные характеристики. По этому критерию подходят платы Micro, Nano, Mini, ProMini. Наиболее оптимальным выбором является плата Nano, поскольку имеет встроенный порт для mini-BUSBJack, что упрощает прошивку микроконтроллера, и в то же время не имеет излишних вычислительных мощностей, что удешевляет изделие.

Платформа Nano, изображенная на рисунке 2.6 базируется на микроконтроллере ATmega328 или ATmega168, имеет небольшие размеры и может использоваться в лабораторных работах и небольших проектах. Она имеет схожую с Arduino Duemilanove функциональность. Отличие заключается в отсутствии силового разъема постоянного тока, работе через кабель mini-B USB и габаритных размерах. Для нашего проекта выбрана платформа на микроконтроллере ATmega168.

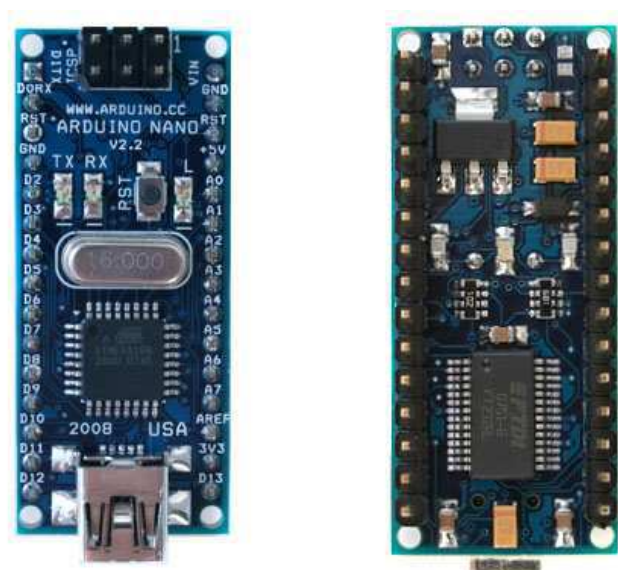


Рисунок 2.6 – ArduinoNano

Arduino Nano может получать питание через подключение mini-B USB, с вывода 30 от нестабилизированного, или с вывода 27 от стабилизированного на 5 вольт внешнего источника питания. Автоматически выбирается источник с самым высоким напряжением. Микросхема FTDI FT232RL получает питание, только если сама платформа запитана от USB.

Краткие характеристики:

- микроконтроллер Atmel ATmega168 или ATmega328;
- рабочее напряжение 5В;
- входное напряжение 6- 20 В;
- 14 цифровых входов/выходов, 6 имеют возможность ШИМ;
- 8 аналоговых выходов;
- постоянный ток через вход/выход 40 мА;
- 16 - 32 Кб флэш-памяти;
- 1 - 2 Кб ОЗУ;
- 512 байт - 1 Кб EEPROM;
- Тактовая частота 16 МГц;
- Размеры 1,85 см x 4,2 см;

Каждый из 14 цифровых выводов Nano, изображенных на рисунке 2.7, используя функции `pinMode()`, `digitalWrite()`, и `digitalRead()`, может настраиваться как вход или выход. Выводы работают при напряжении 5В, каждый вывод имеет отключенный по умолчанию нагрузочный резистор 20-50 кОм и может пропускать до 40 мА.

Некоторые выводы имеют особые функции:

Последовательная шина: 0 (RX) и 1 (TX). Выводы используются для получения (RX) и передачи (TX) данных TTL. Данные выводы подключены к соответствующим выводам микросхемы последовательной шины FTDI USB-to-TTL.

Внешнее прерывание: 2 и 3. Данные выводы могут быть сконфигурированы на вызов прерывания либо на младшем значении, либо на переднем или заднем фронте, или при изменении значения. Подробная информация находится в описании функции `attachInterrupt()`.

ШИМ: 3, 5, 6, 9, 10, и 11. Любой из выводов обеспечивает ШИМ с разрешением 8 бит при помощи функции `analogWrite()`.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Посредством данных выводов осуществляется связь SPI, которая, хотя и поддерживается аппаратной ча-

стью, не включена в язык Arduino.

LED: 13. Встроенный светодиод, подключенный к цифровому выводу 13. Если значение на выводе имеет высокий потенциал, то светодиод горит.

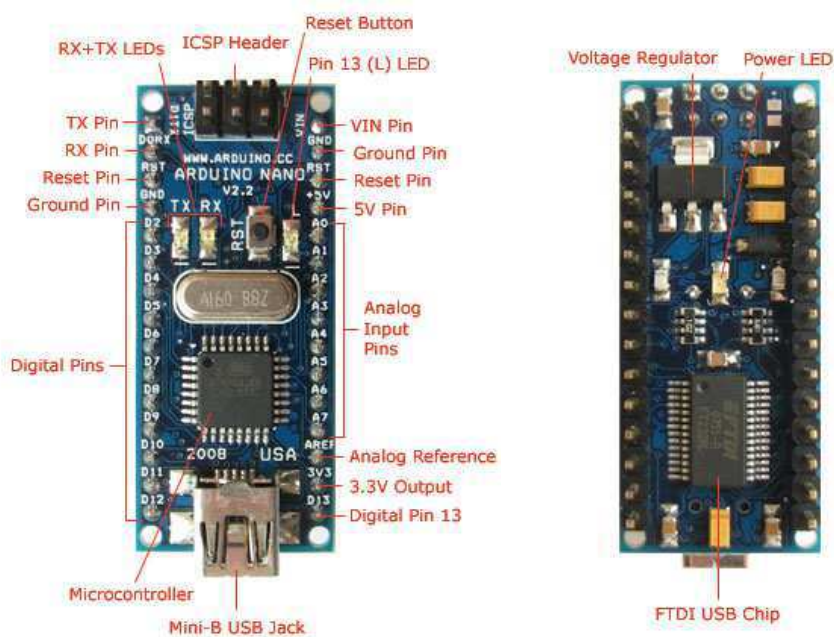


Рисунок 2.7 – Входы и выходы ArduinoNano

На платформе Nano присутствует 8 аналоговых входов, каждый разрешением 10 бит. Стандартно выводы имеют диапазон измерения до 5 В относительно земли, тем не менее, имеется возможность изменить верхний предел посредством функции `analogReference()`. Некоторые выводы имеют дополнительные функции:

I2C: A4 (SDA) и A5 (SCL). Поставляется с библиотекой `Wire`.

Дополнительная пара выводов платформы:

AREF. Опорное напряжение для аналоговых входов. Используется с функцией `analogReference()`.

Reset. Низкий уровень сигнала на выводе перезагружает микроконтроллер. Обычно применяется для подключения кнопки перезагрузки на плате расширения, закрывающей доступ к кнопке на самой плате Arduino.

На платформе Arduino Nano установлено несколько устройств, позволяющих осуществлять связь с компьютером, другими устройствами.

ми Arduino или микроконтроллерами. ATmega168 и ATmega328 поддерживают последовательный интерфейс UART TTL, осуществляемый выводами 0 (RX) и 1 (TX). Установленная на плате микросхема FTDI FT232RL направляет данный интерфейс через USB, а драйверы FTDI, поставляются со средой разработки Arduino, предоставляют виртуальный COM порт программе на компьютере. Библиотекой SoftwareSerial возможно создать последовательную передачу данных через любой из цифровых выводов Nano. ATmega168 и ATmega328 поддерживают интерфейсы I2C (TWI) и SPI.

Платформа программируется посредством среды разработки Arduino. Микроконтроллеры ATmega168 и ATmega328 поставляются с записанным загрузчиком, облегчающим запись новых программ без использования внешних программаторов. Связь осуществляется оригинальным протоколом STK500. Имеется возможность не использовать загрузчик и запрограммировать микроконтроллер через выводы блока ICSP – внутрисхемное программирование. Платформа Nano разработана таким образом, чтобы перед записью нового кода перезагрузка осуществлялась самой программой, а не нажатием кнопки на платформе. Одна из линий FT232RL, управляющих потоком данных, подключена к выводу перезагрузки микроконтроллеров ATmega168 или ATmega328 через конденсатор 100 нФ. Активация данной линии, т.е. подача сигнала низкого уровня, перезагружает микроконтроллер. Программа Arduino, используя данную функцию, загружает код одним нажатием кнопки Upload в самой среде программирования [22].

2.5 Выбор элемента индикации

Для отображения сигналов с фоточувствительных элементов выбран TFT-дисплей совместимый с Arduino, рисунок 2.8. Экран имеет разрешение 128x160 пикселей, ширину 46 и длину 50 мм, управляется драйвером ST7735S, который обеспечивает отображение 18-битного цвета, т.е. 262144 оттенков. На плате устройства также размещается регулятор напряжения 5 – 3,3В и слот для SD карт, который можно использовать для хранения изображений и

последующего их вывода на дисплей. Дисплей имеет неплохое разрешение, качество исполнения и невысокую стоимость.



Рисунок 2.8 – дисплей TFT

Для подключения на плате экрана имеется 16 выводов, три из которых не используются. Выводы 11-14 подключены к разъему SD карты[17]. Назначение выводов дисплея приведено в таблице 2.1.

Таблица 2.1 – Назначение выводов TFTдисплея

Номер вывода	Вывод	Описание
1	GND	земля
2	VCC	питание +5В
3	NC	-
4	NC	-
5	NC	-
6	RESET	сброс
7	A0	линия команд
8	SDA	линия данных
9	SCK	тактовый сигнал
10	CS	выбор устройства
11	SCK	тактовый сигнал SD
12	MISO	данные выход SD
13	MOSI	данные вход SD
14	CS	выбор устройства SD
15	LED+	подсветка +5В
16	LED-	подсветка земля

2.6 Выбор мультиплексора

Мультиплексор – устройство, имеющее несколько сигнальных входов, адресные или управляющие входы и один выход, при этом выбор желаемого входа осуществляется подачей соответствующей комбинации управляющих сигналов. Мультиплексор предназначен для управляемой передачи информации с нескольких источников в один выходной канал. Демультиплексором называется комбинационное логическое устройство, предназначенное для управляемой передачи данных от одного источника информации в несколько выходных каналов. Аналоговые и цифровые мультиплексоры значительно различаются по принципу работы. Первые электрически соединяют выбранный вход с выходом, вторые же не образуют прямого электрического соединения между выбранным входом и выходом, а лишь передают на выход логический уровень с выбранного входа. Аналоговые мультиплексоры иногда называют ключами или коммутаторами. В случае применения аналоговых мультиплексоров выполненных на МДП-транзисторах, одни и те же схемы могут выполнять функции мультиплексора и демультиплексора [15]; такие устройства могут называться коммутаторами.

Схематически мультиплексор можно изобразить в виде коммутатора обеспечивающего подключение одного из нескольких информационных входов к одному выходу устройства. Коммутатор обслуживает управляющая схема, в которой имеются адресные входы и, как правило, разрешающие или по-другому стробирующие. Разрешающие входы используют для расширения функциональных возможностей мультиплексора. Они используются для наращивания разрядности мультиплексора, синхронизации его работы с работой других узлов. Сигналы на разрешающих входах могут разрешать, а могут и запрещать подключение определенного входа к выходу, то есть могут блокировать действие всего устройства. В качестве управляющей схемы обычно используется дешифратор. В цифровых мультиплексорах логические элементы коммутатора и дешифратора обычно объединяются. На рисунке 2.9 изображена обобщённая схема мультиплексора, его условное обозначение и таблица истинности. Входные логические сигналы D_i поступают на входы внутреннего ком-

мутатора мультиплексора и через коммутатор передаются на выход Y . На вход управляющей схемы подаётся слово адресных сигналов X_k . Дополнительно мультиплексор может иметь управляющий вход E , который разрешает или запрещает прохождение входного сигнала на выход Y . В разных типах мультиплексоров при запрещенном состоянии передачи по входу E на выходе Y может быть состояние 0 или 1. Кроме этого, некоторые мультиплексоры могут иметь выход с тремя состояниями: два логических состояния 0 и 1, и третье состояние, отключённый выход –высокоимпедансное или Z -состояние, когда выходное сопротивление велико, выходной внутренней вентиль отключается от выхода специальным внутренним ключом. Перевод мультиплексора в третье состояние производится подачей на вход Output Enable логической 1 или логического 0 – опять же зависит от модели конкретного мультиплексора. На рисунке 2.10 представлена обобщённая схема демультимплексора, его условное обозначение и таблица истинности.

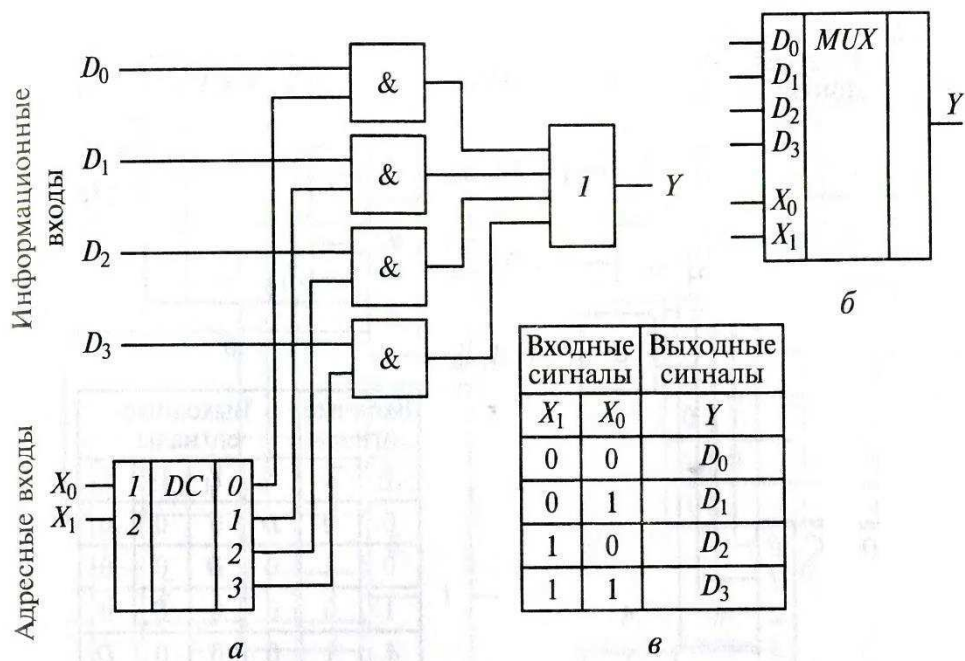


Рисунок 2.9 – Схема мультиплексора (а),
условное обозначение (б), таблица истинности (в)

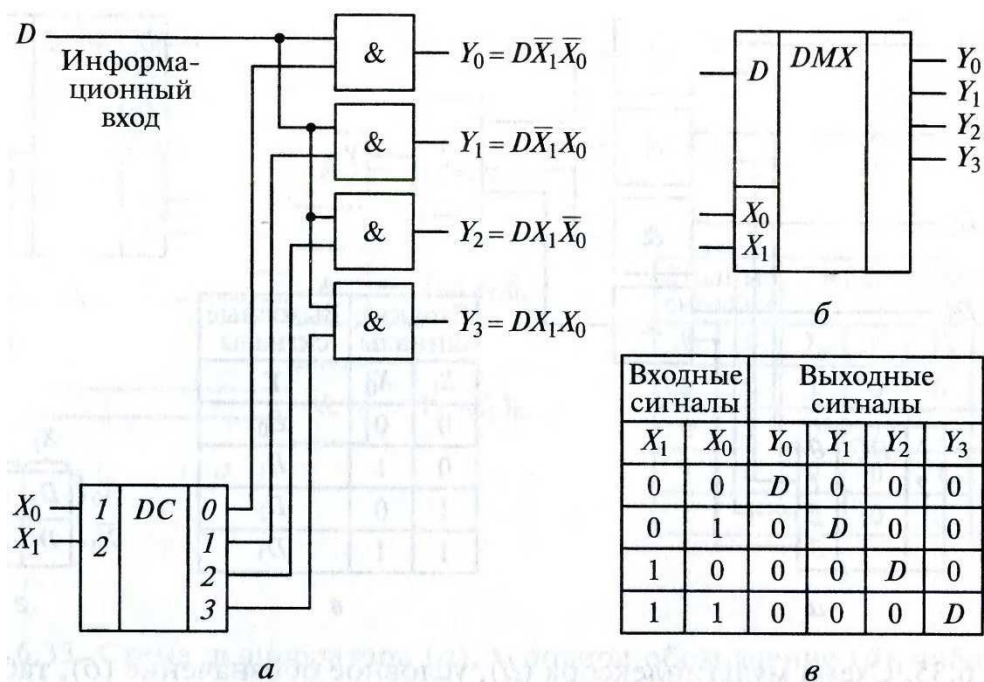


Рисунок 2.10 – Схема демультиплексора (а),
условное обозначение (б), таблица истинности (в)

Для проекта выбран 8-канальный аналоговый мультиплексор, изображенный на рисунке 2.11. Высокоскоростной мультиплексор 74HC4051N совместим со стандартом JEDEC 7A, имеет три управляющих $S_0..S_2$, восемь информационных $Y_0..Y_7$ и разрешающий вход E , активизирующий микросхему при подаче низкого уровня сигнала, также выход Z . Ножки положительного V_{CC} и отрицательного V_{EE} напряжения питания соединяются с источником постоянного напряжения, разница $V_{CC} - V_{EE}$ не должна превышать 10 В. В случае однополярного источника V_{EE} соединяется последовательно входу GND к земле.



Рисунок 2.11 – Внешний вид мультиплексора 74HC4051N

Выбор восьмиканального мультиплексора обоснован наличием большого количества фоторезисторов, с которых необходимо получать данные, и особен-

ностями спроектированной схемы.

2.7 Дополнительные конструктивные элементы

Кроме рассмотренных выше элементов изделия также необходимы:

Конвертер RS485 в TTL на чипе MAX485. Микросхема MAX485 является маломощным трансивером, имеет приёмо-передающий тракт. Каждая часть микросхемы содержит один приёмник и один передатчик. Вход приемника защищён от высокочастотных электромагнитных помех и имеет отказоустойчивую особенность, которое гарантирует высокое качество работы. Все части работают от одного источника питания 5В. Модуль может использоваться как конвертер интерфейсов, для связи через RS-485 между микроконтроллером и другими устройствами. Конвертер приведен на рисунке 2.12.

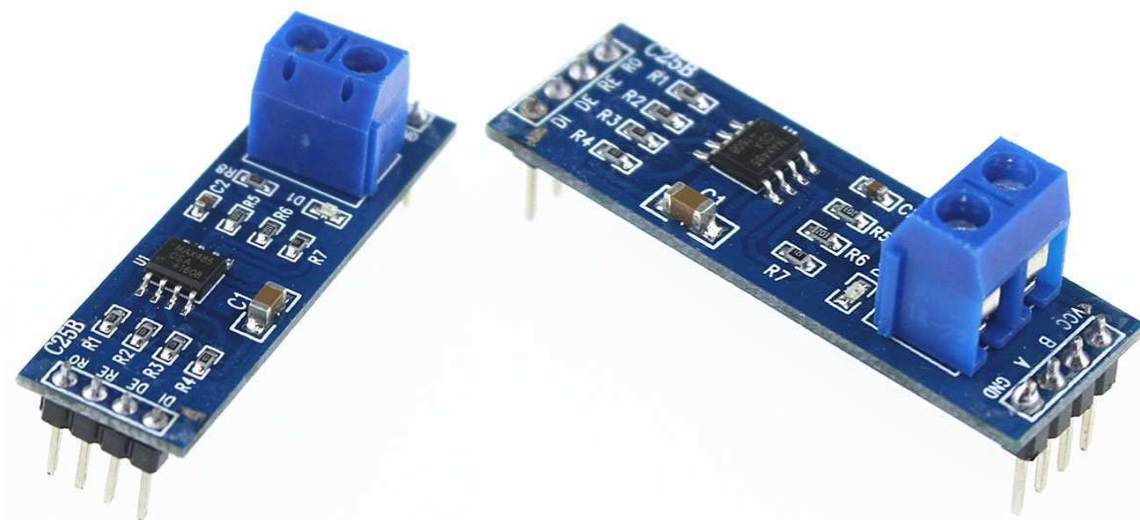


Рисунок 2.12 – Внешний вид конвертера RS485 в TTL

Некоторые характеристики конвертера представлены в таблице 2.2.

Таблица 2.2 – Характеристики конвертера RS485 в TTL

Количество драйверы / приемники	1 / 1
Протокол	RS422, RS485
Напряжение питания	4,75В..5,25 В
Монтажный тип	Для поверхностного монтажа
Температурный диапазон	От минус 40°С до плюс 105°С
Максимальный ток питания	16µА
Монтажный тип	Для поверхностного монтажа
Размеры	44*14 мм
Скорость передачи	2000 Кб/с

RS-485 – стандарт физического уровня для асинхронного интерфейса. Регламентирует электрические параметры полудуплексной многоточечной дифференциальной линии связи типа «общая шина». Стандарт приобрел большую популярность и стал основой для создания целого семейства промышленных сетей, широко используемых в промышленной автоматизации.

Текстолит – электроизоляционный конструкционный материал, применяемый для производства подшипников скольжения, шестерен и других деталей, а также в электро- и радиотехнике. Представляет собой слоистый пластик на основе ткани из волокон и полимерного связующего вещества. Текстолит на основе стеклоткани называется стеклотекстолитом или стеклопластиком, рисунок 2.13. Стеклотекстолит превосходит текстолит по ряду свойств: термостойкость от 140°C до 180°C против 105°C – 130°C у текстолита; удельное сопротивление – 1011 Ом·м против 107 Ом·м; тангенс угла потерь – 0,02 против 0,07. Листовой стеклотекстолит, покрытый медной фольгой, служит основой для изготовления печатных плат [4].



Рисунок 2.13–Двусторонний фольгированный стеклотекстолит

Термоусаживаемые трубки – полые трубки на основе термополимеров, изображены на рисунке 2.14, обладают свойством сжиматься, расширяться, или как-то иначе изменять свои геометрические размеры и форму при нагревании горячим воздухом, открытым пламенем или в горячей воде. Применяются для:

- изоляции в качестве заменителя изоляции;
- восстановления изоляции повреждённых проводов;
- маркировки концов жил кабеля;
- в составе термоусаживаемых кабельных муфт;
- защиты от коррозии;
- механической фиксации тонких деталей в сочетании с клеевым соединением в различных устройствах;
- как упаковочный материал;
- и т.п.



Рисунок 2.14 – Термоусаживаемые трубки

Материал может усаживаться на предметы со сложным профилем, обеспечивая хорошую электрическую изоляцию и механическую защиту, осуществлять внутреннее полимерное покрытие металлических труб для антикислотной и антищелочной защиты, а в некоторых случаях – и защиту от химических и термических воздействий. В Советском Союзе первыми разработали и внедрили в производство, на основе своих авторских свидетельств, термоусаживаемые кабельные муфты и терморасширяющиеся трубки ТРТ в лаборатории Минмонтажспецстроя СССР ВНИИПЭМ ЛенПЭО (1981). Впоследствии лаборатория была преобразована в ЗАО «Термофит». Термоусаживаемые детали изготавливаются преимущественно из полиэтилена низкого или высокого давления той гео-

метрической формы, которую они примут после термоусадки, затем они подвергаются химическому или радиационному воздействию модифицированию. При этом от линейных молекул полимера отщепляются атомы водорода, и молекулы сшиваются между собой, образуя подобную каучуку сетчатую структуру. Затем изделие нагревается до температуры плавления исходного материала и в горячем виде деформируется до нужных размеров и форм, затем охлаждается до комнатной температуры. Полученная деталь приобретает «память формы», и при повторном нагреве в свободном состоянии восстанавливает свою первоначальную геометрическую форму. Существует множество полимеров, из которых изготавливаются термоусаживаемые материалы, и их свойства определяют свойства конкретного изделия. Температурный диапазон эксплуатации термоусаживаемых изделий, в зависимости от материала, может находиться в промежутке от минус 60°C до плюс 260°C. Наиболее распространённые трубки из полиолефинов имеют стандартную рабочую температуру от минус 50°C до плюс 125°C [13].

Кабель mini-USB тип B, рисунок 2.15



Рисунок 2.15 – кабель mini-USB тип B

Гуашь – вид клеевых водорастворимых красок, более плотный и матовый, чем акварель.

Резистор на 220 Ом в качестве одного из плеч делителя напряжения, изображенный на рисунке 2.16, а также провода.



Рисунок 2.16 – Резистор 220 Ом

2.8 Подготовка основания

После выбора всех частей изделия можно приступать к его сборке. Первым делом необходимо произвести разметку выбранной полусферы, изображенной на рисунке 2.17, с целью размещения на ней фоторезисторов.

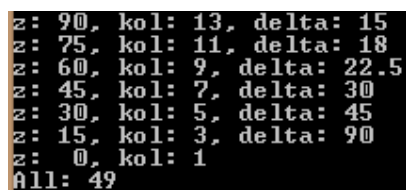


Рисунок 2.17 – Пенопластовая полусфера, основа изделия

Для разметки необходимо подобрать шаг размещения фоторезисторов, что повлияет на будущую точность позиционирования солнечной батареи, а также определит их количество. С данной целью была разработана программа, приведенная ниже. С помощью переменной *interval_z* указывается точность позиционирования солнечной батареи по зениту. С помощью переменной *interval_A* указывается точность позиционирования солнечной батареи по азимуту. После чего программа рассчитывает их общее количество, количество на каждой линии разметки и расстояние в градусах между соседними фотоэлементами. Следует обратить внимание, что, например, при $z = 15^\circ$ мы размещаем всего три фоторезистора с шагом 90° , но разница в угле падения солнечных лучей на эти фоторезисторы 15° , что объясняется тем, что солнце находится практически в зените. В то же время при $z = 75^\circ$ мы размещаем одиннадцать фоторезисторов с шагом 18° , но разница в угле падения солнечных лучей на эти фоторезисторы также 15° , что объясняется тем, что солнце находится практически у горизонта. Таким образом, чем ближе солнце к горизонту, тем большее количество фоточувствительных датчиков требуется

для обеспечения заданной точности позиционирования солнечной батареи. Результат работы программы изображен на рисунке 2.18.

```
#include<iostream>
using namespace std;
#define PI 3.1415926535
int main()
{
double z = PI/2;
double A = PI;
double interval_z = 15*PI/180.0;
double interval_A = 15*PI/180.0;
int lvl_sum = 0;
for (double i = 0; i<z; i = i + interval_z)
    {
cout<< "z: " << 180*(z-i)/PI << ", kol: ";
double max_variation = 2*(z-i);
int lvl = max_variation/interval_A + 1;
lvl_sum+=lvl;
cout<<lvl<< ", delta: " << 180.0/(lvl-
    1)<<endl;
    }
cout<< "z: " << 0 << ", kol: " << 1 <<endl;
cout<< "All: " << lvl_sum+1 <<endl;
return 0;
}
```



```
z: 90, kol: 13, delta: 15
z: 75, kol: 11, delta: 18
z: 60, kol: 9, delta: 22.5
z: 45, kol: 7, delta: 30
z: 30, kol: 5, delta: 45
z: 15, kol: 3, delta: 90
z: 0, kol: 1
All: 49
```

Рисунок 2.18 – Результат работы программы

Для разметки сферы использовался циркуль и гибкая линейка. А для изготовления несквозных отверстий – надфиль круглого сечения диаметром 5 мм. В целях соблюдения глубины 6 мм на надфиль был установлен пенопластовый ограничитель. Также необходимо выполнить сквозные отверстия малого диаметра для выводов фоторезистора. После выполнения отверстий четверть сферы окрашивается в черный цвет, поскольку он обладает наибольшей поглощающей способностью, и попадание на датчики отраженного света сводится к минимуму. Результат подготовки основы изделия изображен на рисунке 2.19.



Рисунок 2.19 – Пенопластовая полусфера, заготовка с отверстиями

После подготовки основы необходимо произвести вклейку фоточувствительных датчиков. Для этого предварительно выводы фоторезисторов были удлинены с помощью скрутки с ними изолированных, и зачищенных на концах, проводов. Места соединения были изолированы с помощью термоусаживаемых трубок. Удлиненные ножки продеваются через сквозное отверстие и укладываются с внутренней стороны полусферы. Вклейка датчиков производится с помощью универсального полимерного клея «Титан». При вклейке датчиков необходимо соблюдать одинаковую величину их углубления. Для этого ножки

резисторов обворачивались кусочкам из пенополистирола, что обеспечивало их более плотную установку, а также одинаковая глубина обеспечивалась их усадкой с помощью винта с пенопластовым ограничителем. Однако без более точных и специальных приспособлений не всегда удастся соблюсти абсолютную точность. Результат вклейки датчиков изображен на рисунке 2.20.



Рисунок 2.20 – Пенопластовая полусфера с вклеенными датчиками

2.9 Разработка печатной платы

Для проекта понадобилась разработка печатной платы, к которой впоследствии будут припаяны выводы фоточувствительных датчиков, мультиплексоры и сопротивление, в качестве плеча делителя напряжения, земля и питание. Данное устройство будет использоваться для выбора конкретного фоторезистора, с помощью управляющих выводов платы Arduino и передачи с него сигнала на вход микроконтроллера. Для этого на плате также предусмотрены шесть

управляющих входов. Для разработки печатной платы был изучен [25] и использовался ProteusDesignSuite – пакет программ для автоматизированного проектирования электронных схем. Пакет представляет собой систему схемотехнического моделирования, базирующуюся на основе моделей электронных компонентов, принятых в PSpice, программе симуляции аналоговой и цифровой техники. Пакет Proteus состоит из двух подпрограмм: ISIS – программасинтезаи моделирования непосредственно электронных схем и ARES – программа разработки печатных плат. Вместе с программой устанавливается набор демонстрационных проектов для ознакомления. Бесплатная ознакомительная версия характеризуется полной функциональностью, но не имеет возможности сохранения файлов. Примечательной особенностью является то, что в ARES можно увидеть 3D-модель печатной платы, что позволяет разработчику оценить своё устройство ещё на стадии разработки[12]. Разработанная электронная схема приведена на рисунке 2.21.

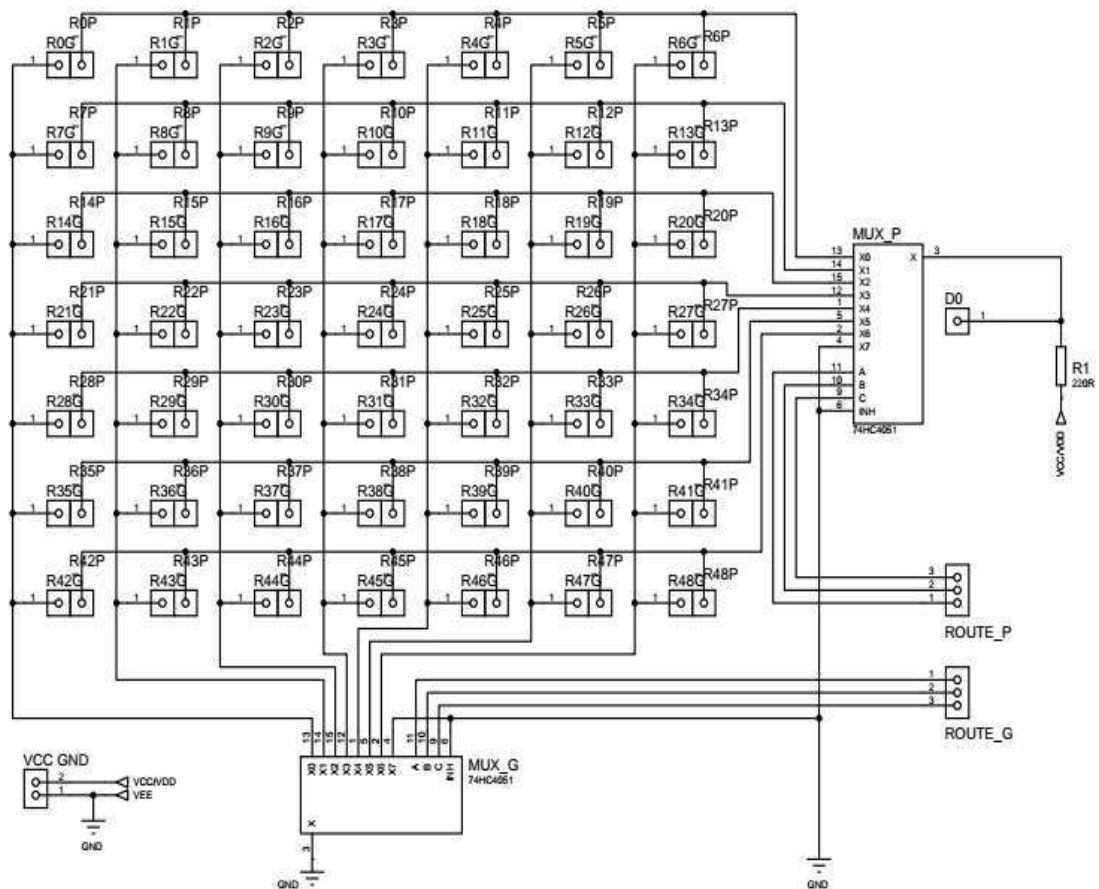


Рисунок 2.21 – Электронная схема выбора датчика

В процессе разработке печатной платы выяснилось, что необходима разработка двухсторонней печатной платы, в связи с большим количеством дорожек. При печати необходимо также учитывать, что в дальнейшем, при переводе печатной платы на текстолит, верхняя часть платы должна быть отражена зеркально, а нижняя в обычном виде. Также желательно рисовать не слишком узкие дорожки, иначе при травлении платы некоторые дорожки могут быть повреждены. Размеры печатной платы – 9,7см длина, 7 см ширина, верхняя часть печатной платы изображена на рисунке 2.22, нижняя часть печатной платы изображена на рисунке 2.23

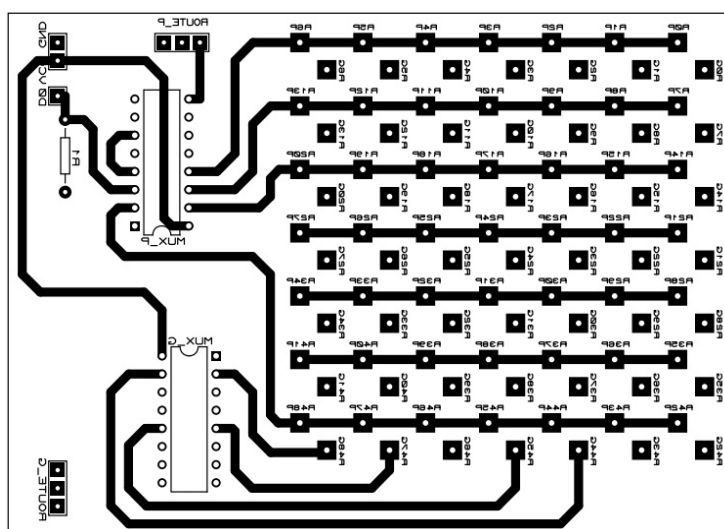


Рисунок 2.22 – Верх печатной платы, зеркальное изображение

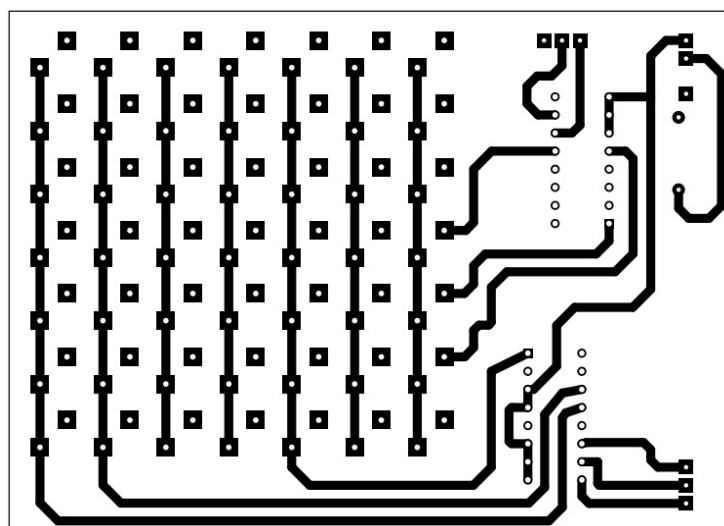


Рисунок 2.23 – Низ печатной платы, обычный вид

2.10 Изготовление печатной платы

При изготовлении печатной платы использовалась технология ЛУТ – ла-

зерно-утюжная технология. Необходимо сказать, что ЛУТ термин не официальный, но достаточно точно отражает суть технологии. Основной принцип данной технологии заключается в переводе рисунка будущей печатной платы с листа бумаги на фольгу текстолита, с помощью утюга. В качестве бумаги для термического переноса использовалась глянцевая фотобумага, результат печати изображен на рисунке 2.24.

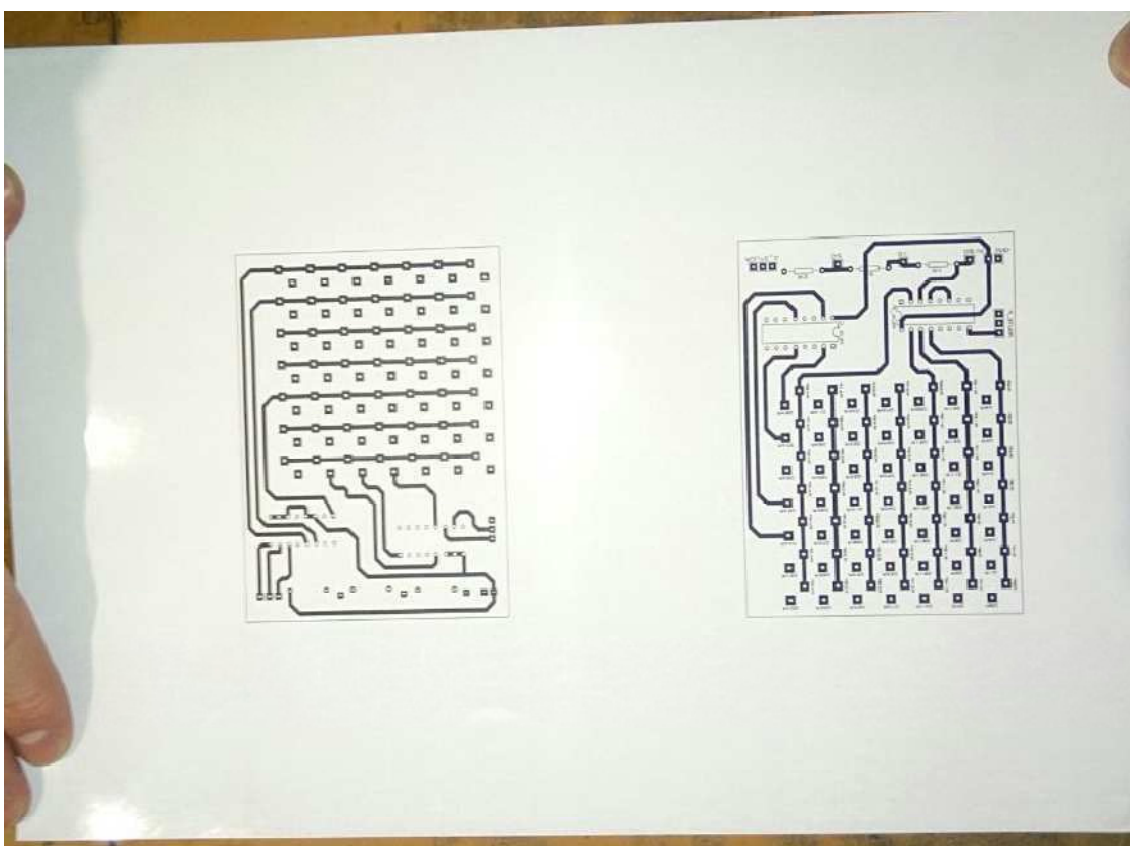


Рисунок 2.24 – Рисунок для термического переноса на стеклотекстолит

Вырезаем рисунок печатной платы, накладываем на текстолит и по линейке обрисовываем. Отрезаем по линиям заготовку фольгированного текстолита, сглаживаем кромок по периметру напильником и очищаем, при необходимости, поверхность фольги мелкой наждачной бумагой. Размещаем кусочек текстолита между слоями газеты. Между утюгом и текстолитом должно быть 2-

3 слоя бумаги. Прогреваем платку около 20 секунд. Теперь помещаем текстолит с рисунком печатной платы, приложенной к нему лицевой стороной, между листами газеты и начинаем проглаживать утюгом с небольшим нажимом в разных направлениях. В случае двухсторонней платы желательно предварительно просверлить несколько отверстий и по ним позиционировать рисунок. Гладим примерно полторы-две минуты. При этом тонер переносится на медную фольгу. Вытаскиваем платку из газеты и даем ей остыть до комнатной температуры. Для того чтобы снять бумагу со стеклотекстолита после переноса изображения наиболее аккуратно, желательно ненадолго опустить плату в ванночку с теплой водой и моющим средством, после чего аккуратно тереть подушечкой пальца до полного удаления бумаги[23].

Для травления платы использовался раствор:

- Лимонная кислота, 30-50 гр.
- Перекись водорода 3%, флакон 100 мл.
- Соль поваренная, 2 чайные ложки, со своей кухни.

Время травления платы среднего размера занимает примерно 15 минут.

Процесс травления платы изображен на рисунке 2.25.

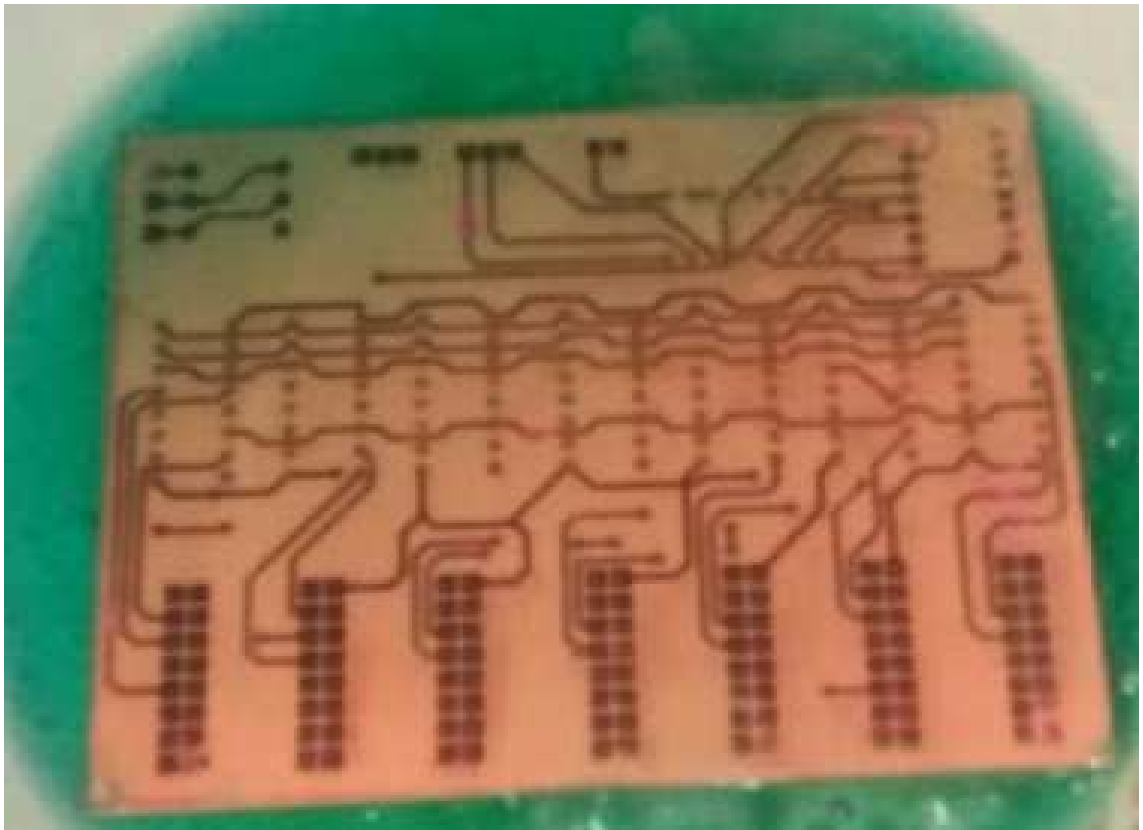


Рисунок 2.25 – Процесс травления печатной платы

После травления остатки тонера и маркера можно смыть ватным тампоном, смоченным ацетоном или растворителем. Мультиметром проверяем целостность дорожек и отсутствие коротких замыканий. Сверлим отверстия сверлом диаметром 1 мм.

Производим лужение печатной платы – одна из операций процесса производства печатных плат или сборки печатных узлов, при которой на металлические паяемые поверхности платы наносится относительно тонкий слой припоя. Данная операция предназначена для подготовки поверхности к последующей пайке. Так же лужение печатной платы является финишным покрытием и входит в состав процесса изготовления платы. В этом случае, кроме подготовки к пайке, оно служит для защиты медных поверхностей от окисления. Лужение выполняется по всей плате после создания проводящего рисунка. Припоем покрывается только открытая часть проводящего рисунка. Покрытие платы должно по возможности соответствовать последующему процессу сборки, поэтому желательно применение того припоя, который будет использоваться при пайке. Защитный слой, нанесенный на печатную плату при помощи данного спосо-

ба, обеспечивает отличные характеристики для пайки изделий и является наиболее распространенным на сегодняшний день финишным покрытием. Он обладает следующими преимуществами:

- доступная цена;
- долговременное сохранение эффекта;
- создание отличного электрического контакта.

К недостаткам данного покрытия относится относительная неравномерность толщины в сравнении с химическими методами и другими более современными методами.

Поскольку технология ЛУТ является «домашним» методом производства печатных плат, то и вариаций их изготовления в интернете достаточно много. Так, в частности, возможно применение другой бумаги и технологии термического переноса, использование различных составов для травления платы, методов её лужения и сверления.

2.11 Монтаж элементов на печатную плату

После изготовления печатной платы производим соединение с ней мультиплексоров, выводов фоторезисторов, резистора на 220 Ом, а также шесть проводов управляющих сигналов, земли и питания, при помощи технологии пайки, рисунок 2.26.

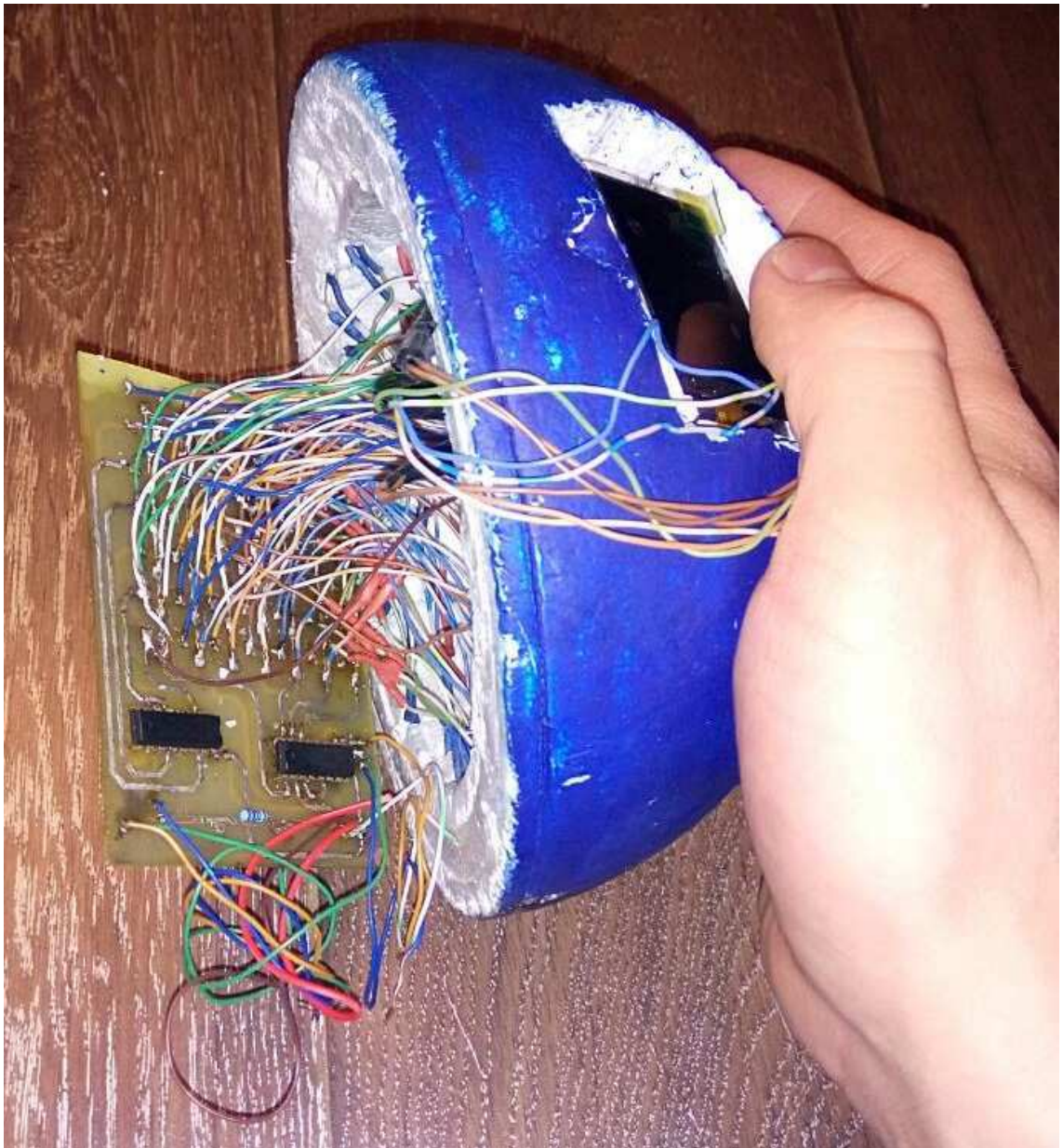


Рисунок 2.26 – Печатная плата с впаянными компонентами

Пайка – технологическая операция, применяемая для получения неразъёмного соединения деталей из различных материалов путём введения между этими деталями расплавленного металла (припоя), имеющего более низкую температуру плавления, чем материал соединяемых деталей. Спаиваемые элементы деталей, а также припой и флюс вводятся в соприкосновение и подвергаются нагреву с температурой температуры плавления припоя, но ниже температуры плавления спаиваемых деталей. В результате припой переходит в жидкое состояние и смачивает поверхности деталей. После этого нагрев прекращается, и припой переходит в твёрдую фазу, образуя соединение. Прочность соединения

во многом зависит от смачиваемости припоем соединяемых поверхностей. При пайке металлов качество смачивания обычно зависит от чистоты поверхности – на ней не должно быть окислов металлов или органических жиров и масел. Для удаления загрязнений, понижения поверхностного натяжения и улучшения растекания припоя применяют ультразвуковые методы активации поверхности. При пайке неметаллических поверхностей или легкоплавкими припоями химические флюсы не помогают смачиванию, поэтому применяют ультразвуковую активацию поверхности.

Для пайки электронных компонентов следует использовать выпускаемый промышленностью оловянно-свинцовый припой с содержанием олова около 61 %, если не указано иное в технологической карте. Припой с таким содержанием олова обладает наименьшей температурой плавления (190°C), и наименьшей прочностью. Для пайки электронных компонентов следует использовать флюсы, не вызывающие коррозию и не обладающие электропроводностью. Такие флюсы имеют надпись «коррозионно-пассивен» и/или «не требует отмывки». Хорошо себя зарекомендовали флюсы в виде геля на канифольной основе. Активные флюсы, например, хлористый цинк, не используются для пайки электронных компонентов и проводов, поскольку флюс остается в порах припоя, проникает в материал платы, под изоляцию провода и его невозможно полностью удалить при отмывке. Современная радиоэлектронная промышленность встала перед фактом организации сбора и удаления отходов, имеющих в своем составе тяжелые металлы, и переходит на технологии с применением материалов, не содержащих свинца [10].

2.12 Установка и подключение дисплея

Предварительно, перед подключением TFT дисплея к Arduino Nano, необходимо произвести его размещение на четверти полусферы, свободной от фоточувствительных датчиков. Для этого с помощью канцелярского ножа вырезаем нишу под экран, сквозное отверстие для соединительных проводов и производим установку экрана, результат приведен на рисунке 2.27. После окончательной проверки работоспособности изделия дисплей будет окончательно за-

креплен, края зашпаклеваны и окрашены.

Существует два основных варианта подключения Arduino и дисплея через SPI. В первом случае, можно использовать любые из GPIO выводов Arduino, но при этом будет использован медленный программный SPI драйвер. Arduino также имеет встроенный аппаратный SPI драйвер, имеющий более высокую скорость работы. На Arduino Nano для этого используются выводы с 10 по 13[17]. Соответствие выводов TFTэкрана и выводов платы Arduino, а также номинал резисторов, через которые производится подключение для ограничения тока, указаны в таблице 2.3. Производим соединение TFTдисплея и ArduinoNанов соответствии с этой таблицей.



Рисунок 2.27 – Установка дисплея

Таблица 2.3 – соответствие выводов TFTдисплея и выводов платы Arduino для аппаратного SPIрежима

Вывод TFT	Вывод Arduino	Номинал резистора
GND	GND	-
VCC	+5 В	-
RESET	8	1кОм
A0	9	1кОм
SDA	11	1кОм
SCK	13	1кОм
CS	10	1кОм
LED+	+5 В	400м
LED-	GND	-

Для управления данным дисплеем воспользуемся библиотеками `Adafruit_QDTechiAdafruit_GFX`. Первая непосредственно использует драйвер дисплея, а вторая предоставляет функции для рисования разных геометрических фигур.

2.13 Окончательная сборка

Окончательно подготовим внешнюю поверхность полусферы, для этого необходимо:

- снабдить поверхность отсеками для хранения USB-кабеля и батареи питания;
- обеспечить доступ к программируемой плате Arduino;
- вывести интерфейс RS-485 для связи установкой;
- в отсеке для батареи питания установить контактную площадку;
- зашпаклевать неровности и окрасить вторую четверть сферы.

На рисунке 2.28 изображен окончательный вид системы автоматического определения положения солнца для позиционирования солнечной батареи со стороны датчиков.

На рисунке 2.29 изображен окончательный вид системы автоматического определения положения солнца для позиционирования солнечной батареи с тыльной стороны.



Рисунок 2.28 – Вид системы с лицевой стороны

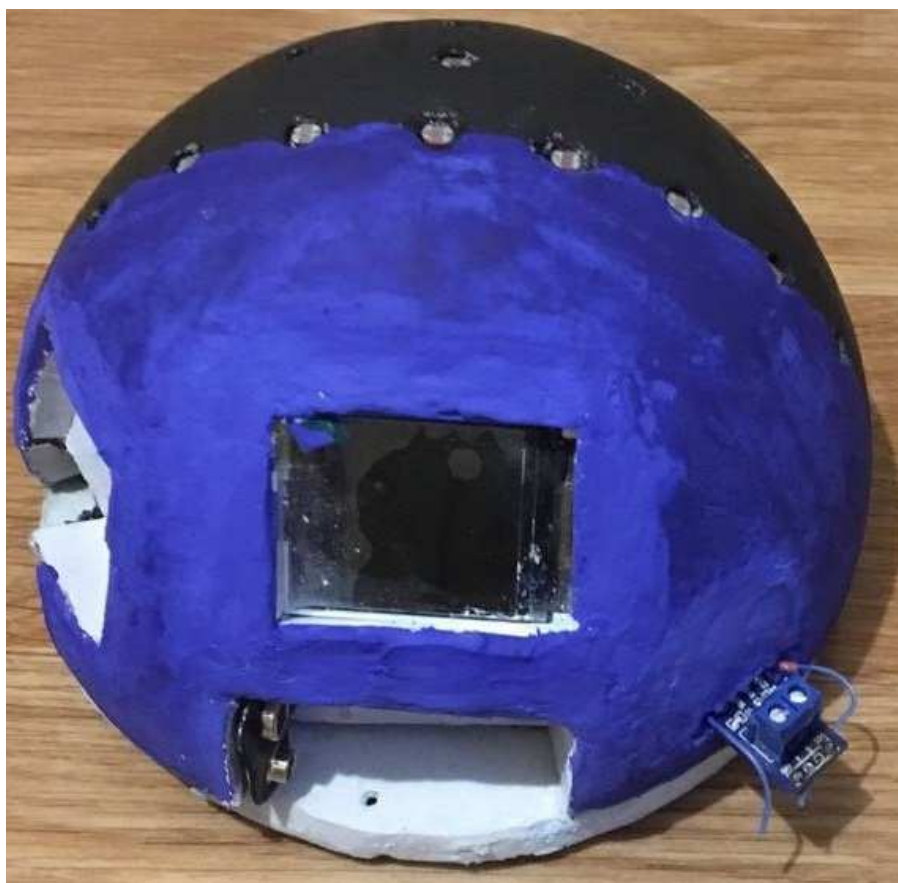


Рисунок 2.29 – Вид системы с тыльной стороны

Теперь мы имеем подготовленную внешнюю часть полусферы. Установленные в неё фоторезисторы соединены с изготовленной платой выбора датчика, а TFT-экран с платой Arduino.

Подключим к Arduino разъем питания для батарейки типа «Крона» в соответствии с таблицами 2.4.

Таблица 2.4 – соединение разъема питания с Arduino

Цвет провода разъема питания	Вывод Arduino
Красный	VIN
Черный	GND

Все остальные части системы питаются от разъема 5В на плате Arduino, т.к. им необходим данный тип напряжения.

Управляющие выводы платы выбора датчика подключаются к Arduino по цифровым входам 2...7, а информационный вывод каналоговому входу 4.

В заключение произведем соединение конвертирующего модуля на базе чипа MAX485 в соответствии с таблицей 2.5. Также стоит отметить особенность данной платы Arduino, при прошивке необходимо отключать вывод TX или RX от конвертирующего модуля, для чего соединяющий провод выведен наружу.

Таблица 2.5 – соединение конвертирующего модуля с Arduino

Конвертирующий модуль на базе MAX485	Вывод Arduino
DI (DriverInput - вход передатчика)	TX1 (Transmit - передать)
RO (Receiver Out - выход приёмника)	RX0 (Receive - получить)
DE (DriverEnable - разрешение работы передатчика)	A5 (В режиме Digital)
~RE (ReceiverEnable - разрешение работы приёмника)	A5 (В режиме Digital)
Vcc	Vcc
GND	GND

3 РАЗРАБОТКА ПРОГРАММНОЙ ЧАСТИ

Вся программа часть делится на четыре блока:

- предварительные определения;
- настройки контроллера;
- основная программа;
- служебные функции.

Блок предварительных определений. Данный блок включает в себя код по подключению библиотек; объявлению и, при необходимости, определению глобальных переменных; объявлению функций. В процессе отладки выяснилось, что библиотека, предназначенная для работы с дисплеем несовместима по цветовой гамме с дисплеем, а другие библиотеки либо полностью несовместимы, либо занимают недопустимо много памяти. В связи с этим предопределенные цвета, приведенные в листинге блока ниже, устанавливались экспериментально.

```
// *****  
// ===== ПРЕДВАРИТЕЛЬНЫЕ ОПРЕДЕЛЕНИЯ =====  
// *****  
// Основная графическая библиотека  
#include<Adafruit_GFX.h>  
// Железозависимая библиотека для TFT экрана  
#include<Adafruit_QDTech.h>  
// Библиотека для работы по SPI протоколу  
#include<SPI.h>  
  
// Назначение выводов TFT  
#define sclk13 // Не изменять  
#define mosi11 // Не изменять  
#define cs10 // CS  
#define dc 9 // Data
```

```

#define rst 8 // Reset

// Определяем цвета
#define RED_LIGHT 84000
#define RED_MIDDLE 23200
#define RED_DARK 136900
#define BLUE_LIGHT 79500
#define BLUE_MIDDLE 114900
#define BLUE_DARK 59700
#define GREEN_LIGHT 27700
#define GREEN_MIDDLE 77300
#define GREEN_DARK 38700
#define BLACK 102700
#define WHITE 69600

// Подключаем дисплей
Adafruit_QDTechTft = Adafruit_QDTech(cs, dc, rst);
// Массив данных об освещенности
int LightArray[7][13];

// *** ПРЕДОПРЕДЕЛЕНИЯ ФУНКЦИЙ *** //
// Функция очищает массив 7x13
void clearLightArray();

// Функция отрисовывает прямоугольник на tft экране
void
draw_iRect(int x0, int y0, int value, int L, int H);

// Функция помещает значение
// с фоторезистора в массив 7x13

```



```
voidputToLightArray(int n1, int n2, int value);
```

Блок настройки контроллера. Функция `setup()` вызывается, когда стартует программа. Используется для инициализации переменных, определения режимов работы выводов, запуска используемых библиотек и т.д. Функция `setup` запускает только один раз, после каждой подачи питания или сброса платы Arduino. В листинге ниже в данном блоке настраивается дисплей, входы и выходы контроллера.

```
// *****  
// ===== НАСТРОЙКИ КОНТРОЛЛЕРА =====  
// *****  
voidsetup() {  
    // Инициализация tft дисплея  
tft.init();  
tft.invertDisplay(true);  
tft.setRotation(1); // 0 - Портрет, 1 - Ландшафт  
tft.fillScreen(BLACK);  
tft.setTextWrap(true);  
tft.setTextSize (1);  
delay(500);  
  
    // Установка цифровых выходов  
    // управляющих выбором фотоэлемента  
pinMode(2, OUTPUT);  
pinMode(3, OUTPUT);  
pinMode(4, OUTPUT);  
pinMode(5, OUTPUT);  
pinMode(6, OUTPUT);  
pinMode(7, OUTPUT);
```

Блок основной программы. Исходя из названия понятно, что в данной части выполняется основная подпрограмма, функция *voidloop()* зациклена и постоянно повторяется. В подпрограмме выполняется объявление и инициализация локальных переменных. Далее происходит поочередный опрос фоторезисторов с помощью циклической конструкции, для этого использован приём побитового сравнения переменной цикла и номера вывода в двоичном виде. Одновременно с опросом значений датчика мы записываем его в массив и производим определение максимального и минимального значений освещенности. После цикла производим обход по массиву и отображаем на экране графическое представление данных с датчиков, записанных ранее в массив.

```
// ***** //
// ===== ОСНОВНАЯ ПРОГРАММА ===== //
// ***** //
voidloop() {
    // === 1. Инициализация переменных и массива === //
    // Значение освещенности
    int valLight;
    // Максимальная освещенность
    intmaxLight = 0;
    // Минимальная освещенность
    intminLight = 0;
    // Номер резистора с максимальной освещенностью
    intnumPR = 0;
    // Очищаем массив
    clearLightArray();

    // === 2. Проход по фоторезисторам === //
    for (int i = 0; i<=6; ++i) {
        // Установка на конкретный фоторезистор
```

```

        digitalWrite(4,B001&i);
digitalWrite(3,B010&i);
digitalWrite(2,B100&i);
    // ...
for (int j = 0; j<=6; ++j) {
digitalWrite(7,B001&j);
digitalWrite(6,B010&j);
digitalWrite(5,B100&j);
// ...
// Фоторезистор определён, начало обработки
valLight = 1023-analogRead(4);
// Считываем значение с фоторезистора
putToLightArray(i, j, valLight);
// Записываем значение в массив 7x13
    //
    // Проверка на максимум
if (valLight>maxLight) {

maxLight = valLight;
numPR = i;
    }
    // Проверка на минимум
if (valLight<minLight) {
minLight = valLight;
}
    }
}

// === 3. Отрисовка массива на экране === //
tft.fillScreen(BLACK);

```

```

for (inti = 0; i < 7; i++) {
for (int j = 0; j < 13; j++) {
if (LightArray[i][j] == -1) continue;
draw_iRect(j,i,LightArray[i][j], minLight, maxLight);
}
}
delay(1000);
}

```

Блок служебных функций. Содержит код, вынесенный из основной программы и оформленный в виде отдельных функций:

clearLightArray() выполняет очистку массива освещенности, необходима при новом цикле основной программы, чтобы очистить данные оставшиеся от предыдущего прохода.

putToLightArray() помещает значение с фоторезистора в массив. Также происходит отображение данных с платы размером 7x7, т.к. именно в двух циклах, внешнем и вложенном, происходит перебор резисторов, на внешнее их расположение на сфере формой 7x13. При этом каждая из 7 строк заполняется в соответствии с расположением датчиков на сфере, т.е. первая сверху строка имеет одно значение, вторая три, третья пять и т.д.

draw_iRect() принимает номер строки, столбца и значение текущего элемента массива, минимальный и максимальный элемент массива. На основании полученных данных отображает элемент таблицы – ячейку, значение внутри неё и производит заливку цветом ячейки. Заливка цветом производится по принципу деления текущего диапазона значений на более мелкие диапазоны. Отдельным цветом окрашивается максимальный элемент.

```

// ***** //
// ===== СЛУЖЕБНЫЕ ФУНКЦИИ ===== //
// ***** //
// ФУНКЦИЯ clearLightArray //
// ОЧИЩАЕТ МАССИВ 7x13 //

```

```

voidclearLightArray()
{
for (inti = 0; i < 7; i++)
for (int j = 0; j < 13; j++)
LightArray[i][j] = -1;
}

// ФУНКЦИЯputToLightArray //
// ПОМЕЩАЕТ ЗНАЧЕНИЕ С ФОТОРЕЗИСТОРА НА ПЛАТЕ 7x7 В
МАССИВ 7x13 //
// int n1 - "строка" на плате //
// int n2 - "столбец" на плате //
// intarr[][13] - массив 7x13 //

voidputToLightArray(int n1, int n2, int value) {
    // Ряд 6
if(n1 == 0) LightArray[6][n2] = value;
else if (n1 == 1 && n2 < 6) LightArray[6][7 + n2] =
value;

    // Ряд 5
else if (n1 == 1 && n2 == 6) LightArray[5][0] = value;
else if (n1 == 2) LightArray[5][1+n2] = value;
else if (n1 == 3 && n2 < 3) LightArray[5][8+n2] =
value;

    // Ряд 4
else if (n1 == 3 && n2 >= 3) LightArray[4][n2-3] =
value;
else if (n1 == 4 && n2 < 5) LightArray[4][4+n2] =

```

```

value;

    // Ряд 3
else if (n1 == 4 && n2 >= 5) LightArray[3][n2-5] =
value;
else if (n1 == 5 && n2 < 5) LightArray[3][2+n2] =
value;

    // Ряд 2
else if (n1 == 5 && n2 >= 5) LightArray[2][n2-5] =
value;
else if (n1 == 6 && n2 < 3) LightArray[2][2+n2] =
value;

    // Ряд 1
else if (n1 == 6 && n2 >= 3 && n2 < 6)
LightArray[1][n2-3] = value;

    // Ряд 0
else if (n1 == 6 && n2 == 6) LightArray[0][0] = value;
}

// ФУНКЦИЯdraw_iRect//
voiddraw_iRect(int x0, int y0, int value, int L, int H)
{
    // Форматируемстроку
    String strtmp, strValOne, strValTwo;
    strtmp = String(value,DEC);
    intdopNull = 4-strtmp.length();
    for (int k = 1; k<=dopNull; k++) {

```

```

strtmp = "0"+strtmp;
}
strValOne = strtmp.substring(0,2);
strValTwo = strtmp.substring(2);

// Выводимнаэкран
intcolorRect;
intcolorFont = WHITE;
int Range = (H-L)/4;
if (value <= L+Range) colorRect = BLUE_DARK;
else if ((value >= L+Range) &&(value < L+2*Range))
colorRect = BLUE_LIGHT;
else if ((value >= L+2*Range) &&(value < L+3*Range))
colorRect = RED_LIGHT;
else if ((value >= L+3*Range) &&(value <=L+4*Range))
colorRect = RED_MIDDLE;
if (value == H) colorRect = RED_DARK;

tft.fillRect(13*x0-x0,19*y0-y0,13,19,colorRect);
tft.drawRect(13*x0-x0,19*y0-y0,13,19,QDTech_BLACK);
tft.setTextColor(colorFont);
tft.setCursor(13*x0-x0+1,19*y0-y0+1+1);
tft.println(strValOne);
tft.setCursor(13*x0-x0+1,19*y0-y0+1+9);
tft.print(strValTwo);
}

```

Программа прошивается в плату ArduinoNanoи после перезагрузки отображает показания фоторезисторов на экране, результат работы программы на готовом устройстве можно увидеть на рисунке 3.1, на снимке не видно цифровых значений. Полный текст программы приведён в приложении В.

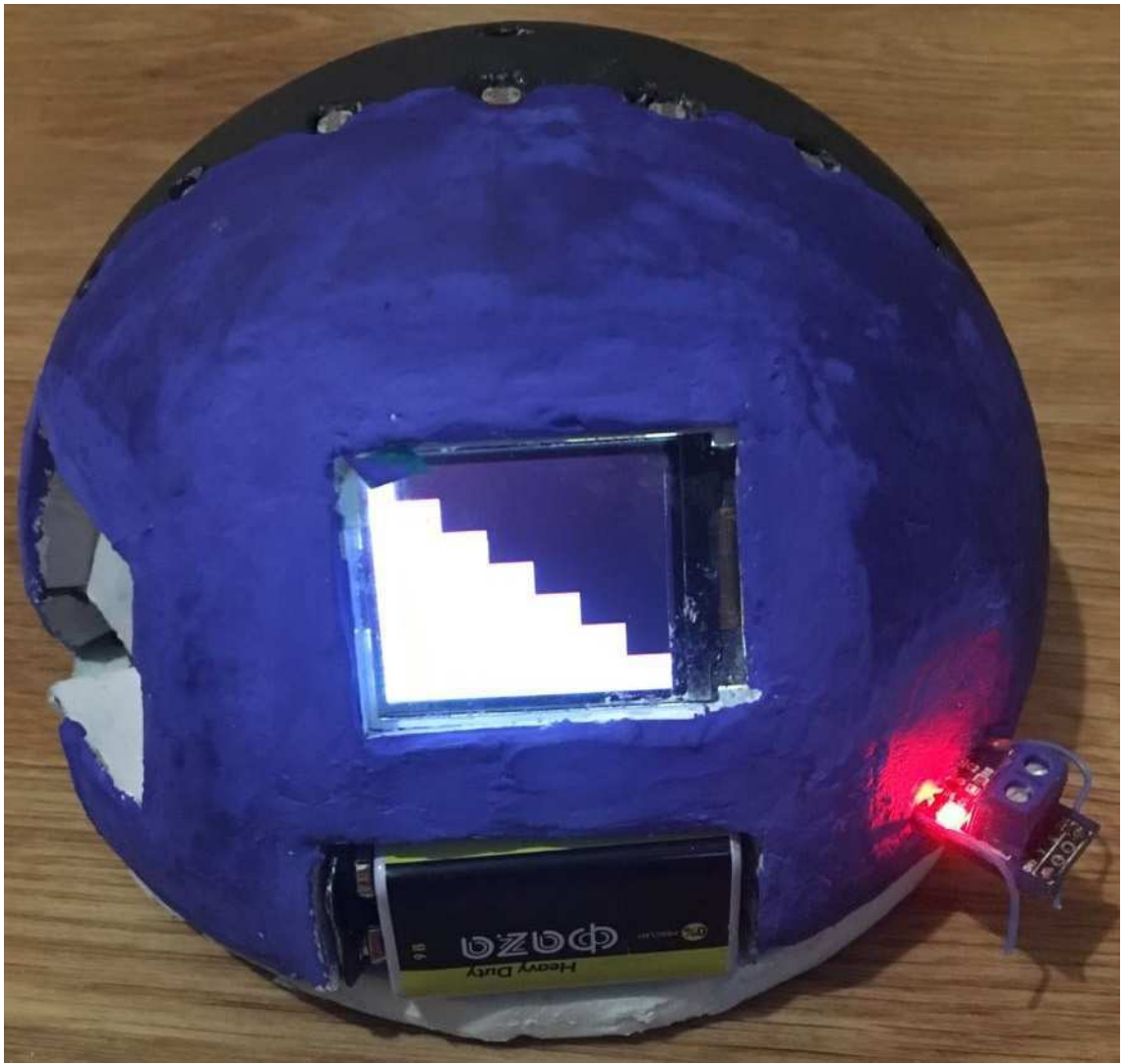


Рисунок 3.1 – АСУВ процессе работы

4 ПЕРСПЕКТИВЫ РАЗВИТИЯ И МОДЕРНИЗАЦИИ СИСТЕМЫ

В соответствии с техническим заданием, приведенным в приложении А, автоматизированная система должна реализовывать возможность дальнейшей модернизации, как программного обеспечения, так и комплекса технических средств. В частности в процессе разработки и тестирования системы выдвинуты предложения по развитию и модернизации системы, описанные ниже.

Увеличить надежность и качество системы. В процессе тестирования системы было обнаружено, что при опросе конкретного резистора, на показания несколько влияют и другие датчики, вероятно по причине недостаточного сопротивления незадействованных входов. Данная проблема решается заменой мультиплексора или электронной схемы выбора датчика, рисунок 4.1. На рисунке 4.2 изображена верхняя часть печатной платы, на рисунке 4.3, нижняя часть печатной платы. В данной версии электронной схемы происходит прямая коммутация фоторезистора, не пересекаясь с другими линиями, однако её недостатком является необходимость использовать 8 мультиплексоров.

Для улучшения надёжности при промышленной эксплуатации также можно подобрать фоторезисторы с более широким температурным диапазоном работы.

Снизить себестоимость и увеличить точность системы. Для снижения себестоимости необходимо уменьшить количество датчиков, при этом, если собрать некоторый объем данных об измерениях, можно улучшить алгоритм программы, и аппроксимировать показания с датчиков, что улучшит точность системы.

Для снижения себестоимости использования системы можно подобрать элемент питания, подключаемый напрямую, без стабилизации напряжения.

При массовом производстве значительно снизит себестоимость использование схемы только с микроконтроллером, без платы Arduino.

Увеличение скорости работы, достижимо при использовании более быстродействующих элементов и совершенствования кода программы.

Также можно заменить дисплей на более качественный и большего разрешения.

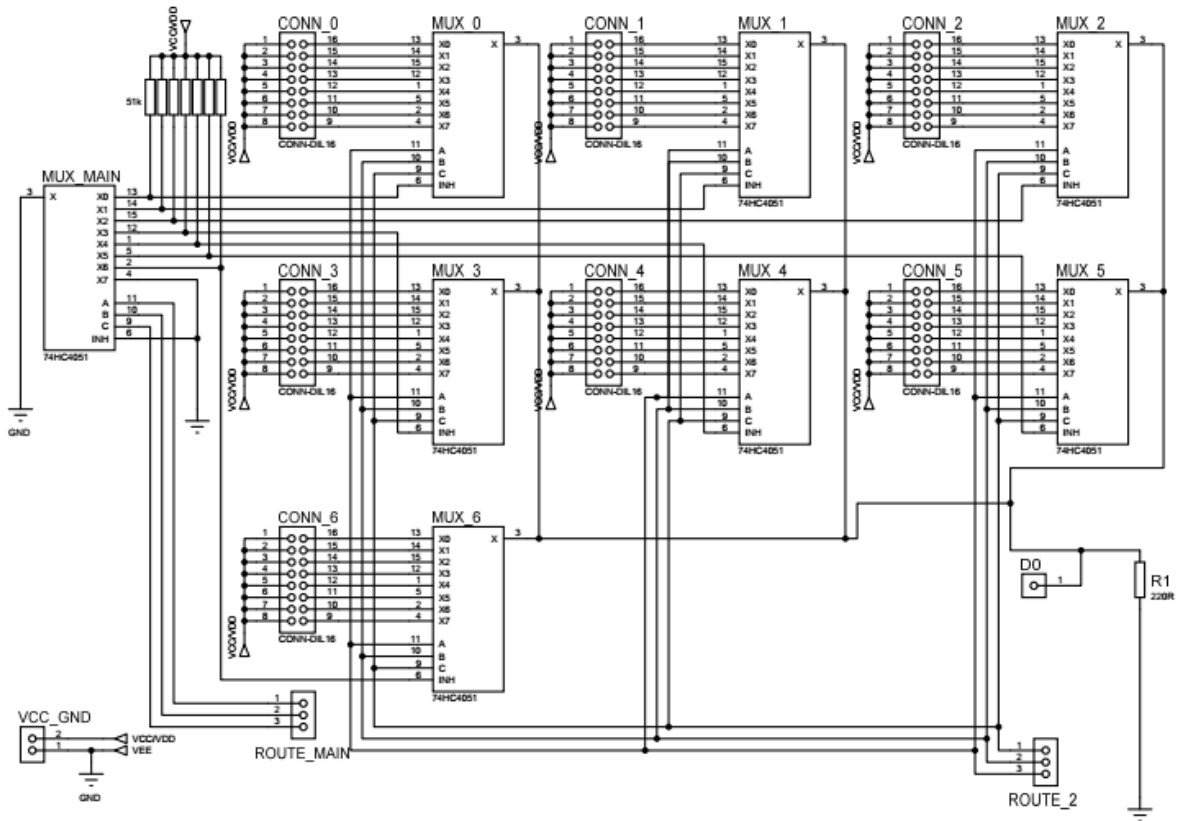


Рисунок 4.1 – Переработанная электронная схема выбора датчика

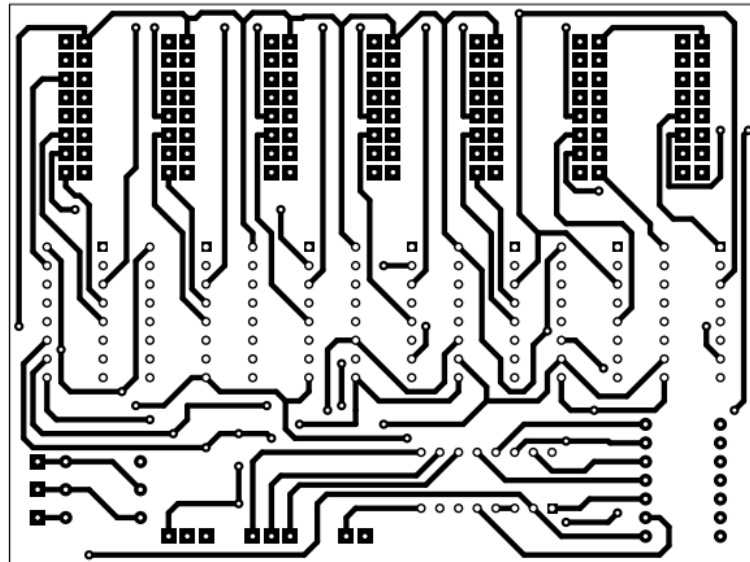


Рисунок 4.2 – Верх переработанной печатной платы, зеркальное изображение

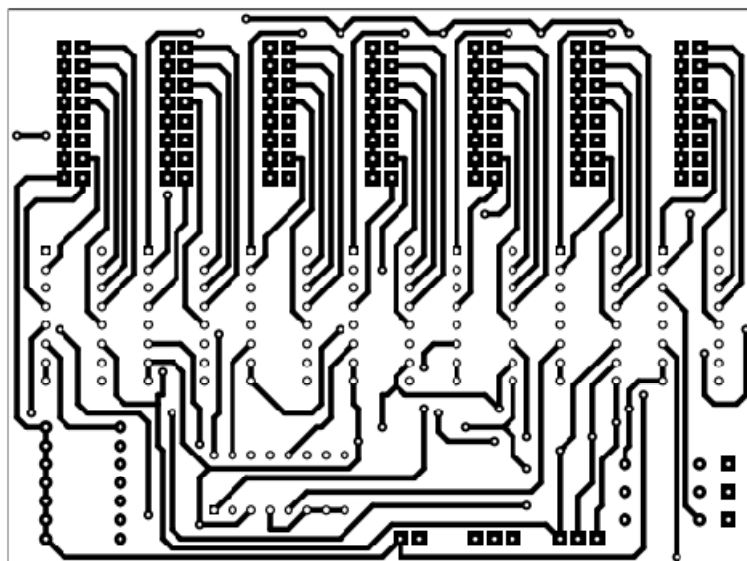


Рисунок 4.3 – Низ переработанной печатной платы, обычный вид

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе на основе микроконтроллера и лабораторной солнечной энергоустановки разработана автоматическая система определения положения солнца для позиционирования солнечной батареи, выполняющая следующие функции:

- автоматическое определение положения солнца;
- выработку задания следящим системам лабораторной установки;
- предоставление информации об уровне освещённости;
- сопряжения системы с аппаратными программными комплексами лабораторной солнечной энергоустановки;
- расширения за счет размещения нескольких устройств.

В результате выполнения работы, удалось составить функциональную схему, схему сборки системы, выбрать оборудование, которое смогло обеспечить для системы надежные показатели точности и качества управления, изготовить систему, удовлетворяющую поставленным задачам. Данная система автоматизирует процесс наведения солнечной установки; предоставляет информацию, на основе которой будет возможно вырабатывать оптимальные решения о позиционировании солнечной батареи, в том числе с учетом интенсивности светового потока. Такая система в перспективе позволит снизить стоимость обслуживания систем слежения, за счет уменьшения количества движения приводов, и в то же время сохранить их преимущество перед простым наращиванием площади батарей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Альтернативная энергия [Электронный ресурс] : офиц. сайт. – Режим доступа : <https://alternativenergy.ru/solnechnaya-energetika/507-solnechnaya-energetika-elektrostantsii-perspektivy.html>. – 10.05.2016
2. Бессонов, Л.А. Теоретические основы электротехники. Электрические цепи : учебник / Л.А. Бессонов. – 11-е изд., М.: Гардарики, 2007. – 701с.
3. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа : <https://ru.wikipedia.org/wiki/Фотоэлемент>. – 01.08.2016
4. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа : <https://ru.wikipedia.org/wiki/Текстолит>. – 12.11.2016
5. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа : <https://ru.wikipedia.org/wiki/RS-485>. – 20.08.2016
6. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа : https://ru.wikipedia.org/wiki/Солнечная_энергетика. – 01.05.2016
7. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа : <https://ru.wikipedia.org/wiki/Фоторезистор>. – 21.08.2016
8. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа : <https://ru.wikipedia.org/wiki/Фотодиод>. – 25.08.2016
9. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа : <https://ru.wikipedia.org/wiki/Фототранзистор>. – 25.08.2016

10. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа :
<https://ru.wikipedia.org/wiki/Пайка>. – 10.12.2016
11. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа :
<https://ru.wikipedia.org/wiki/Arduino>. – 19.09.2016
12. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа :
[https://ru.wikipedia.org/wiki/Proteus \(система автоматизированного проектирования\)](https://ru.wikipedia.org/wiki/Proteus_(система_автоматизированного_проектирования)). – 15.11.2016
13. Википедия – свободная энциклопедия [Электронный ресурс] : офиц. сайт. – Режим доступа :
[https://ru.wikipedia.org/wiki/Термоусаживающиеся материалы](https://ru.wikipedia.org/wiki/Термоусаживающиеся_материалы). – 12.11.2016
14. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы – Введ. 1990–01–01.
15. Лаврентьев, Б.Ф. Схемотехника электронных средств : учеб. пособие для студ. высш. учеб. заведений / Б. Ф. Лаврентьев. – М. : Издательский центр «Академия», 2010. – 336 с.
16. Оформление выпускных квалификационных и курсовых работ (проектов) – Утв. 01.04.2011 / АмГУ / СТО СМК 4.2.3.05 – 2011
17. Популярная робототехника [Электронный ресурс] : офиц. сайт. – Режим доступа :
<http://www.poprobot.ru/home/Arduino-TFT-SPI>. – 28.10.2016
18. Рыбалев, А.Н. Система позиционирования солнечной батареи : статья / А.Н. Рыбалев. – Благовещенск, 2016. – 12 стр.
19. Таненбаум, Э. Архитектура компьютера / Таненбаум Э., Остин Т. – 6-е изд. – СПб.: Питер, 2013. – 816 с.
20. Arduino.cc [Электронный ресурс] : офиц. сайт. – Режим доступа :

- <https://www.arduino.cc/en/Main/Products>. – 15.10.2016
21. Arduino.ru :Аппаратная платформа [Электронный ресурс] : офиц. сайт.
– Режим доступа :<http://arduino.ru>. – 12.09.2016
22. Arduino.ru :Аппаратная платформа [Электронный ресурс] : офиц. сайт.
– Режим доступа :
<http://arduino.ru/Hardware/ArduinoBoardNano>. – 25.10.2016
23. Schem.net: Сайт паяльник [Электронный ресурс] : офиц. сайт. – Режим
доступа :
<http://csem.net/master/45.php>. – 23.11.2016
24. DatasheetsPDF [Электронный ресурс] : офиц. сайт. – Режим доступа :
<http://datasheetspdf.com/datasheet/GL5539.html>. – 03.09.2016
25. PROTEUS [Электронный ресурс] : офиц. сайт. – Режим доступа :
<http://proteus321.narod.ru>. – 13.11.2016

ПРИЛОЖЕНИЕ А

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ

Техническое задание разработано с требованиями ГОСТ 34.602–89 [14].

1 ОБЩИЕ СВЕДЕНИЯ

- Техническое задание составлено на разработку подсистемы автоматического определения положения солнца для позиционирования солнечных батареи.
- Заказчик: ФГБОУ ВПО Амурский государственный университет
- Исполнитель: Меркурьев С.Н.
- Система разрабатывается на основании следующих документов:
 - ФГОС направления подготовки бакалавров 15.03.04 АТПиП;
 - учебный план направления подготовки бакалавров 15.03.04 автоматизации технологических процессов и производств;
- Плановый срок начала работ по разработке подсистемы автоматического определения положения солнца – 13 апреля 2016 года.
- Плановый срок окончания работ по созданию автоматической системы позиционирования солнечной батареи 08 февраля 2017 года.
- Таблица А.1 со сроками результатов

Таблица А.1 – Таблица со сроками результатов

Задание	Дата выполнения
Функциональная схема	10.08.2016
Техническое задание	10.09.2016
Физическая реализация	01.01.2017
Разработка ПО	15.01.2017
Тестирование и разработка документации	08.02.2017

2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Система автоматического определения положения солнца предназначена для:

- автоматического определения положения солнца;
- выработки задания следящим системам лабораторной установки;

Продолжение ПРИЛОЖЕНИЯ А

- исследований распределения освещенности при различных метеорологических условиях
- разработки алгоритмов оптимального наведения солнечной установки, в том числе алгоритмов принимающих решение о целесообразности движения при различной интенсивности светового потока.

2.2 Цели создания системы:

- разработка устройства, позволяющего как использовать его в промышленных масштабах, так и проводить исследования в области солнечной энергетики;
- совершенствование лабораторной базы кафедры;
- получение навыков построения систем управления данного класса;

3 ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ

Объектом автоматизации является процесс позиционирования солнечной батареи входящей в состав лабораторной солнечной энергоустановки – электромеханического устройства, состоящего из металлического основания, на котором установлена вертикальная направляющая конструкция, в верхней части которой закреплена солнечная батарея. Батарея приводится в движение сигналами ПЛК 73, вокруг вертикальной и горизонтальной оси с изменением азимута и угла места, с прямоходными исполнительными механизмами на основе двигателей постоянного тока напряжением 12В. Диапазон изменения азимута 120°С, угла места 90°С.

Объект эксплуатируется в диапазоне температур от минус 20°С до плюс 75°С, при влажности до 80%.

4 ТРЕБОВАНИЯ К СИСТЕМЕ

4.1 Требования к системе в целом

Система управления должна включать следующие элементы:

- блок питания 9 В;
- блок микроконтроллера, вычисляющего оптимальное положение;

Продолжение ПРИЛОЖЕНИЯ А

- блок визуального информирования;
- блок контроля и измерения положения приводов;
- блок измерения освещенности;
- блок коммутации с солнечной установкой.

Блок коммутации с солнечной установкой должен передавать информацию объекту автоматизации.

Блок контроллера предназначен для получения и обработки сигналов датчиков положения и выработки сигналов управления в соответствии с программой управления.

Блок измерения освещенности и определения оптимального положения автоматически определяет положение солнца и выработывает задание следящим системам лабораторной установки.

4.1.1. Требования к структуре и функционированию системы

1) Требования к способам и средствам связи для информационного обмена между компонентами системы. Для информационного обмена между компонентами системы используются цифровые интерфейсы RS-232 и RS-485.

2) Требования к режимам функционирования системы. Для автоматизированной системы определены следующие режимы функционирования:

- Нормальный режим функционирования;
- Аварийный режим функционирования.

Основным режимом функционирования АС является нормальный режим. В нормальном режиме функционирования системы:

- программное обеспечение и технические средства системы обеспечивают возможность функционирования в течение дня.
- исправно работает оборудование, составляющее комплекс технических средств;
- исправно функционирует системное, базовое и прикладное программное обеспечение системы.

Продолжение ПРИЛОЖЕНИЯ А

Для обеспечения нормального режима функционирования системы необходимо выполнять требования и выдерживать условия эксплуатации программного обеспечения и комплекса технических средств системы, указанные в соответствующих технических документах.

Аварийный режим функционирования системы характеризуется отказом одного или нескольких компонент программного или технического обеспечения. В случае перехода системы в аварийный режим необходимо:

- сообщить об аварии;
- начать выполнение программы по поиску оптимального положения заново.

После этого необходимо выполнить комплекс мероприятий по устранению причины перехода системы в аварийный режим.

3) требования по диагностированию систем:

Автоматизированная система должна предоставлять инструменты диагностики основных процессов системы, мониторинга процесса выполнения программы.

Компоненты должны предоставлять удобный интерфейс для возможности просмотра диагностических событий, мониторинга процесса выполнения программ.

При возникновении аварийных ситуаций, либо ошибок в программном обеспечении, диагностические инструменты должны позволять сохранять полный набор информации, необходимой разработчику для идентификации проблемы (снимки экранов, текущее состояние памяти, файловой системы).

4) перспективы развития, модернизации системы.

АС должна реализовывать возможность дальнейшей модернизации как программного обеспечения, так комплекса технических средств, таких как:

Продолжение ПРИЛОЖЕНИЯ А

- увеличить точность системы;
- снизить себестоимость;
- повысить надежность и качество;
- увеличить скорость работы.

4.1.2 Требования к численности и квалификации персонала системы

Для плановой диагностики АС требуется один человек.

4.1.3 Требования к надежности

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

- при сбоях в системе электроснабжения аппаратной части;
- при ошибках в работе аппаратных средств;
- при ошибках, связанных с программным обеспечением.

4.1.4 Требования к безопасности

Все внешние элементы технических средств системы, находящиеся под напряжением, должны иметь защиту от случайного прикосновения, а сами технические средства иметь защитное заземление в соответствии с ГОСТ 12.1.030-81 и ПУЭ. Общие требования пожарной безопасности должны соответствовать нормам на бытовое электрооборудование. Факторы, оказывающие вредные воздействия на здоровье со стороны всех элементов системы (в том числе инфракрасное, ультрафиолетовое, рентгеновское и электромагнитное излучения, вибрация, шум, электростатические поля, ультразвук строчной частоты и т.д.), не должны превышать действующих норм (СанПиН 2.2.2./2.4.1340-03 от 03.06.2003 г.).

4.1.5 Требования к эргономике и технической эстетике

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса. Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен

Продолжение ПРИЛОЖЕНИЯ А

обеспечивать быстрое отображение экранных форм. Навигационные элементы должны быть выполнены в удобной для пользователя форме. Средства редактирования информации должны удовлетворять принятым соглашениям в части использования функциональных клавиш, режимов работы, поиска, использования оконной системы. Ввод-вывод данных системы, прием управляющих команд и отображение результатов их исполнения должны выполняться в интерактивном режиме. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям системы.

Все надписи экранных форм, а также сообщения, выдаваемые пользователю должны быть на русском языке.

Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях система должна выдавать пользователю соответствующие сообщения.

4.1.6 Требования к транспортабельности для подвижных АС

АС в сложенном виде должна быть компактной, а также должна иметь возможность быстрой и простой сборки.

4.1.7 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Для нормальной эксплуатации разрабатываемой системы должно быть обеспечено бесперебойное питание. При эксплуатации система должна быть обеспечена соответствующая стандартам хранения и эксплуатации.

Периодическое техническое обслуживание используемых технических средств должно проводиться в соответствии с требованиями технической документации изготовителей, но не реже одного раза в год.

Продолжение ПРИЛОЖЕНИЯ А

В процессе проведения периодического технического обслуживания должны проводиться внешний и внутренний осмотр и чистка технических средств, проверка контактных соединений, проверка параметров настроек работоспособности технических средств и тестирование их взаимодействия.

На основании результатов тестирования технических средств должны проводиться анализ причин возникновения обнаруженных дефектов и приниматься меры по их ликвидации.

Восстановление работоспособности технических средств должно проводиться в соответствии с инструкциями разработчика и поставщика технических средств и документами по восстановлению работоспособности технических средств и завершаться проведением их тестирования. Размещение оборудования, технических средств должно соответствовать требованиям техники безопасности, санитарным нормам и требованиям пожарной безопасности.

Все пользователи системы должны соблюдать правила эксплуатации электронной вычислительной техники.

4.1.8 Требования по сохранности информации при авариях

Программное обеспечение АС должно восстанавливать свое функционирование при корректном перезапуске аппаратных средств. Приведенные выше требования не распространяются на компоненты системы, разработанные третьими сторонами и действительны только при соблюдении правил эксплуатации этих компонентов.

4.1.9 Требования к защите от влияния внешних воздействий

Защита от влияния внешних воздействий должна обеспечиваться средствами программно – технического комплекса.

4.1.10 Требования к патентной чистоте

Установка системы в целом, как и установка отдельных частей системы не должна предъявлять дополнительных требований к покупке лицензий на программное обеспечение сторонних производителей, кроме программного обеспечения, указанного в разделе 4.3.4, ГОСТ 34.602–89.

Продолжение ПРИЛОЖЕНИЯ А

4.1.11 Дополнительные требования

Дополнительные требования не предъявляются.

4.2 Требования к видам обеспечения

4.2.1 Требования к математическому обеспечению системы

Математические методы и алгоритмы, а также программное обеспечение, используемые при разработке АС должны быть максимально оптимизированными и понятными для разработчиков.

4.2.2 Требования к информационному обеспечению системы

Состав, структура и способы организации данных в системе должны быть определены на этапе технического проектирования.

Технические средства, обеспечивающие хранение информации, должны использовать современные технологии, позволяющие обеспечить повышенную надежность хранения данных и оперативную замену оборудования.

4.2.3 Требования к лингвистическому обеспечению системы

Все прикладное программное обеспечение системы для организации взаимодействия с пользователем должно использовать русский язык.

4.2.4 Требования к программному обеспечению системы

При проектировании и разработке системы необходимо максимально эффективным образом использовать программное обеспечение.

4.2.5 Требования к техническому обеспечению

Техническое обеспечение системы должно максимально и наиболее эффективным образом использовать существующие технические средства.

4.2.6 Требования к метрологическому обеспечению

Разрабатываемая АС должна обеспечивать следующие метрологические требования: чувствительность солнечной батареи должна быть не ниже 97%, погрешность – не выше 3%, угловое позиционирование солнечной батареи не превышает 0,5°.

Продолжение ПРИЛОЖЕНИЯ А

4.2.7 Требования к организационному обеспечению

Организационное обеспечение системы должно быть достаточным для эффективного выполнения возложенных на него обязанностей при осуществлении автоматизированных и связанных с ними неавтоматизированных функций системы.

5 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

Перечень документов, по ГОСТ 34.201-89, предъявляемых по окончании соответствующих стадий и этапов работ:

Таблица А.2 – Состав и содержание работ по созданию системы

Этап	Содержание работ	Результаты работ
1	Разработка технического обеспечения	Создание структурной схемы, функциональной схемы автоматизации и принципиальной схемы
2	Разработка программного обеспечения	Описание алгоритма, программного обеспечения, написание руководства пользователя, составление перечня входных сигналов и данных
3	Определение потребностей в материалах и оборудовании	Составление ведомости оборудования и материалов и локальный сметный расчет

6 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ

6.1 Виды, состав, объем, методы испытаний системы

Виды, состав, объем, и методы испытаний системы должны быть изложены в программе и методике испытаний АС, разрабатываемой в составе рабочей документации.

6.2 Общие требования к приемке работ по стадиям

Все создаваемые в рамках настоящей работы программные изделия передаются заказчику, как в виде готовых модулей, так и в виде исходных кодов, представляемых в электронной форме на стандартном машинном носителе.

6.3 Статус приемочной комиссии

Статус приемочной комиссии определяется заказчиком до проведения испытаний.

Продолжение ПРИЛОЖЕНИЯ А

7 ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ СИСТЕМЫ В ДЕЙСТВИЕ

В ходе выполнения проекта на объекте автоматизации требуется выполнить работы по подготовке к вводу системы в действие. При подготовке к вводу в эксплуатацию АС заказчик должен обеспечить выполнение следующих работ:

- Обеспечить соответствие помещений и рабочих мест пользователей системы в соответствии с требованиями;
- Обеспечить выполнение требований, предъявляемых к программно-техническим средствам, на которых должно быть развернуто программное обеспечение АС;
- Совместно с исполнителем подготовить план развертывания системы на технических средствах заказчика;
- Провести опытную эксплуатацию АС.

Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие, включая перечень основных мероприятий и их исполнителей должны быть уточнены на стадии подготовки рабочей документации и по результатам опытной эксплуатации.

8 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

Техническая часть:

- 1.) Схема автоматизации
- 2.) Схема принципиальная

Программная часть:

- 1.) Описание алгоритма
- 3.) Описание программного обеспечения
- 4.) Руководство пользователя

Экономическая часть:

- 1.) Ведомость оборудования и материалов
- 2.) Локальный сметный расчет

Продолжение ПРИЛОЖЕНИЯ А

9 ИСТОЧНИКИ РАЗРАБОТКИ

Учебники, учебные пособия, и другие материалы:

– Программирование программируемых логических контроллеров ОВЕН ПЛК 63 и ПЛК 73;

– Контроллер программируемый логический ОВЕН ПЛК73. Руководство по эксплуатации;

ГОСТ 34.201-89 Виды, комплектность, обозначения документов при создании АС.

ПРИЛОЖЕНИЕ Б

ПРИЛОЖЕНИЕ В

```
// *****  
// ===== ПРЕДВАРИТЕЛЬНЫЕ ОПРЕДЕЛЕНИЯ =====  
// *****  
// Основная графическая библиотека  
#include<Adafruit_GFX.h>  
// Железозависимая библиотека для tft экрана  
#include<Adafruit_QDTech.h>  
// Библиотека для работы по SPI протоколу  
#include<SPI.h>  
// Назначение выводов TFT  
#define clk13 // Не изменять  
#define mosi11 // Не изменять  
#define cs10 // CS  
#define dc 9 // Data  
#define rst 8 // Reset  
// Определяем цвета  
#define RED_LIGHT 84000
```

```
#define RED_MIDDLE 23200
#define RED_DARK 136900
#define BLUE_LIGHT 79500
#define BLUE_MIDDLE 114900
#define BLUE_DARK 59700
#define GREEN_LIGHT 27700
#define GREEN_MIDDLE 77300
#define GREEN_DARK 38700
#define BLACK 102700
#define WHITE 69600
```

Продолжение ПРИЛОЖЕНИЯ В

```
// Подключаем дисплей
Adafruit_QDTechTft = Adafruit_QDTech(cs, dc, rst);
// Массив данных об освещенности
int LightArray[7][13];
// *** ПРЕОПРЕДЕЛЕНИЯ ФУНКЦИЙ *** //
// Функция очищает массив 7x13
```

```

voidclearLightArray();
// Функцияотрисовываетпрямоугольникнатftэкране
void
draw_iRect(int x0, int y0, int value, int L, int H);
// Функцияпомещаетзначение
// с фоторезистора в массив 7x13
voidputToLightArray(int n1, int n2, int value);
// ***** //
// ===== НАСТРОЙКИ КОНТРОЛЛЕРА ===== //
// ***** //
voidsetup() {
    // Инициализация tft дисплея
tft.init();
tft.invertDisplay(true);
tft.setRotation(1); // 0 - Портрет, 1 - Ландшафт
tft.fillScreen(BLACK);
tft.setTextWrap(true);
tft.setTextSize (1);
delay(500);

```



```
// Установка цифровых выходов
// управляющих выбором фотоэлемента
pinMode(2, OUTPUT);
```

Продолжение ПРИЛОЖЕНИЯ В

```
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
// ***** //
// ===== ОСНОВНАЯ ПРОГРАММА ===== //
// ***** //
voidloop() {
    // === 1. Инициализация переменных и массива === //
    // Значение освещенности
    intvalLight;
    // Максимальная освещенность
```

```
intmaxLight = 0;
// Минимальная освещенность
intminLight = 0;
// Номер резистора с максимальной освещенностью
intnumPR = 0;
// Очищаем массив
clearLightArray();
    // === 2. Проход по фоторезисторам === //
for (int i = 0; i<=6; ++i) {
// Установка на конкретный фоторезистор
digitalWrite(4,B001&i);
digitalWrite(3,B010&i);
digitalWrite(2,B100&i);
for (int j = 0; j<=6; ++j) {
digitalWrite(7,B001&j);
```

Продолжение ПРИЛОЖЕНИЯ В

```
digitalWrite(6,B010&j);
digitalWrite(5,B100&j);
```

```

// Фоторезистор определён, начало обработки
valLight = 1023-analogRead(4);
// Считываем значение с фоторезистора
putToLightArray(i, j, valLight);
// Записываем значение в массив 7x13
    // Проверка на максимум
if (valLight>maxLight) {
maxLight = valLight;
numPR = i;
    }
    // Проверка на минимум
if (valLight<minLight) {
minLight = valLight;
}
}
}
// === 3. Отрисовка массива на экране === //
tft.fillScreen(BLACK);
for (inti = 0; i < 7; i++) {

```

```

for (int j = 0; j < 13; j++) {
if (LightArray[i][j] == -1) continue;
draw_iRect(j,i,LightArray[i][j], minLight, maxLight);
}
}
delay(1000);
}

```

Продолжение ПРИЛОЖЕНИЯ В

```

// ***** //
// ===== СЛУЖЕБНЫЕ ФУНКЦИИ ===== //
// ***** //
// ФУНКЦИЯ clearLightArray //
// ОЧИЩАЕТ МАССИВ 7x13 //
void clearLightArray()
{
for (int i = 0; i < 7; i++)
for (int j = 0; j < 13; j++)
LightArray[i][j] = -1;
}

```

```

}
// ФУНКЦИЯ putToLightArray //
// ПОМЕЩАЕТ ЗНАЧЕНИЕ С ФОТОРЕЗИСТОРА НА ПЛАТЕ 7x7 В //
// МАССИВ 7x13 //
// int n1 - "строка" на плате //
// int n2 - "столбец" на плате //
// int arr[][13] - массив 7x13 //
void putToLightArray(int n1, int n2, int value) {
    // Ряд 6
    if(n1 == 0) LightArray[6][n2] = value;
    else if (n1 == 1 && n2 < 6) LightArray[6][7 + n2] =
value;
    // Ряд 5
    else if (n1 == 1 && n2 == 6) LightArray[5][0] = value;
    else if (n1 == 2) LightArray[5][1+n2] = value;
    else if (n1 == 3 && n2 < 3) LightArray[5][8+n2] =
value;
    // Ряд 4

```

Продолжение ПРИЛОЖЕНИЯ В

```
elseif (n1 == 3 &&n2 >= 3) LightArray[4][n2-3] =
value;
else if (n1 == 4 && n2 < 5) LightArray[4][4+n2] =
value;
    // Ряд 3
else if (n1 == 4 && n2 >= 5) LightArray[3][n2-5] =
value;
else if (n1 == 5 && n2 < 5) LightArray[3][2+n2] =
value;
    // Ряд 2
else if (n1 == 5 && n2 >= 5) LightArray[2][n2-5] =
value;
else if (n1 == 6 && n2 < 3) LightArray[2][2+n2] =
value;
    // Ряд 1
else if (n1 == 6 && n2 >= 3 && n2 < 6)
LightArray[1][n2-3] = value;
```

```

// Ряд 0
else if (n1 == 6 && n2 == 6) LightArray[0][0] = value;
}
// ФУНКЦИЯdraw_iRect//
voiddraw_iRect(int x0, int y0, int value, int L, int H)
{
// Форматируемстроку
String strtmp, strValOne, strValTwo;
strtmp = String(value,DEC);
intdopNull = 4-strtmp.length();
for (int k = 1; k<=dopNull; k++) {

```

Продолжение ПРИЛОЖЕНИЯ В

```

strtmp = "0"+strtmp;
}
strValOne = strtmp.substring(0,2);
strValTwo = strtmp.substring(2);
// Выводимнаэкран
intcolorRect;

```

```

int colorFont = WHITE;
int Range = (H-L)/4;
if (value <= L+Range) colorRect = BLUE_DARK;
else if ((value >= L+Range) &&(value < L+2*Range))
colorRect = BLUE_LIGHT;
else if ((value >= L+2*Range) &&(value < L+3*Range))
colorRect = RED_LIGHT;
else if ((value >= L+3*Range) &&(value <=L+4*Range))
colorRect = RED_MIDDLE;
if (value == H) colorRect = RED_DARK;
tft.fillRect(13*x0-x0,19*y0-y0,13,19,colorRect);
tft.drawRect(13*x0-x0,19*y0-y0,13,19,QDTech_BLACK);
tft.setTextColor (colorFont);
tft.setCursor (13*x0-x0+1, 19*y0-y0+1+1);
tft.println(strValOne);
tft.setCursor (13*x0-x0+1, 19*y0-y0+1+9);
tft.print(strValTwo);
}

```

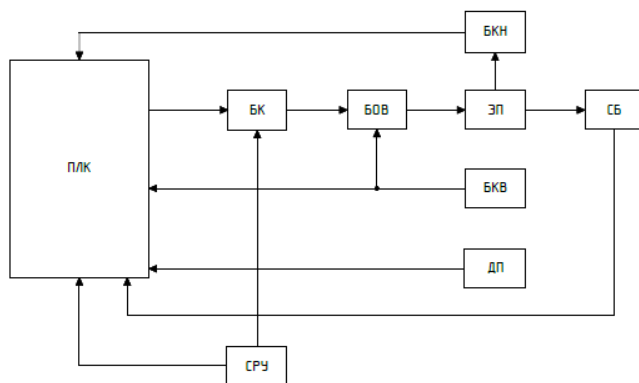

3D-модель



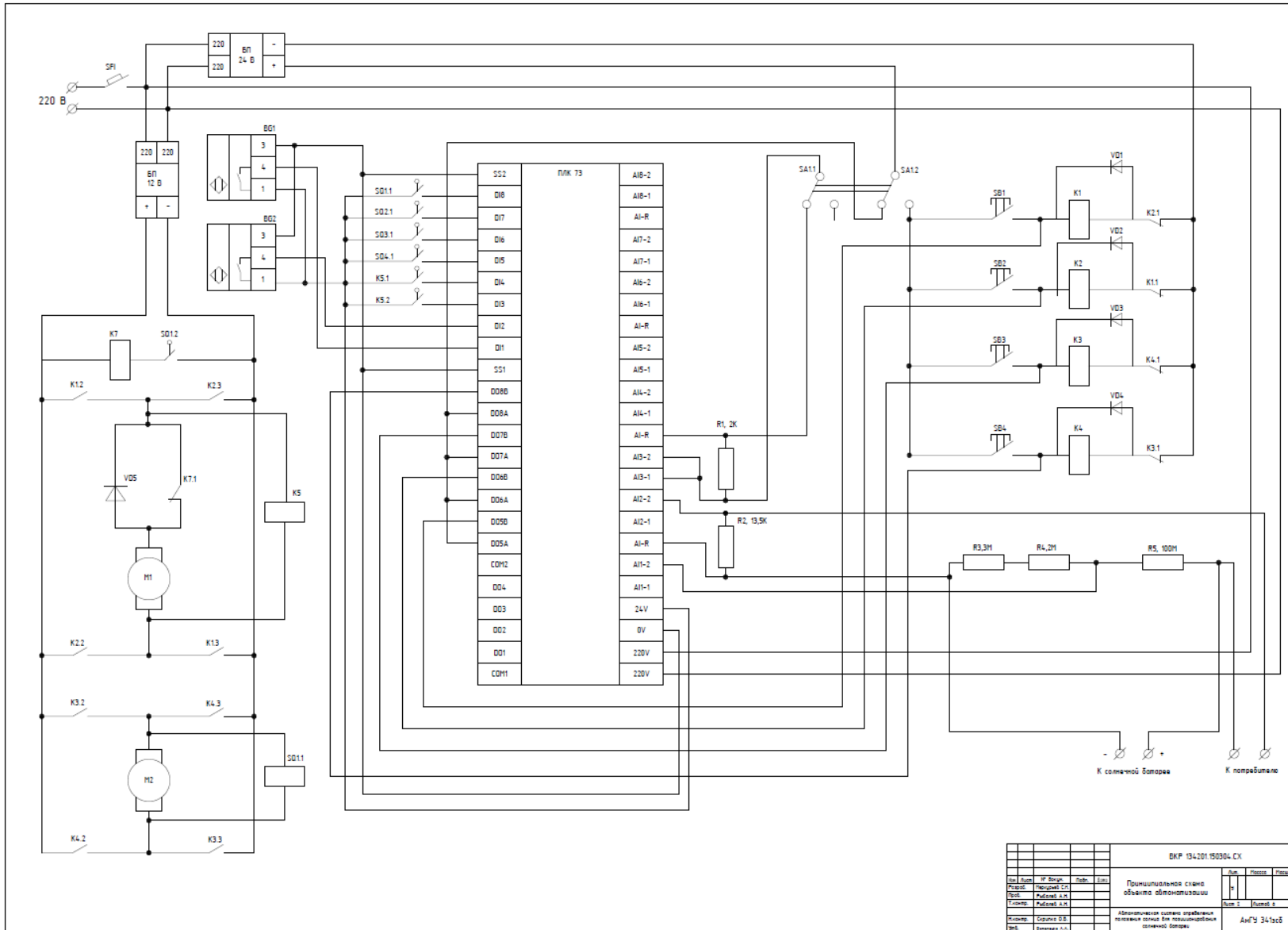
Фотографическое изображение



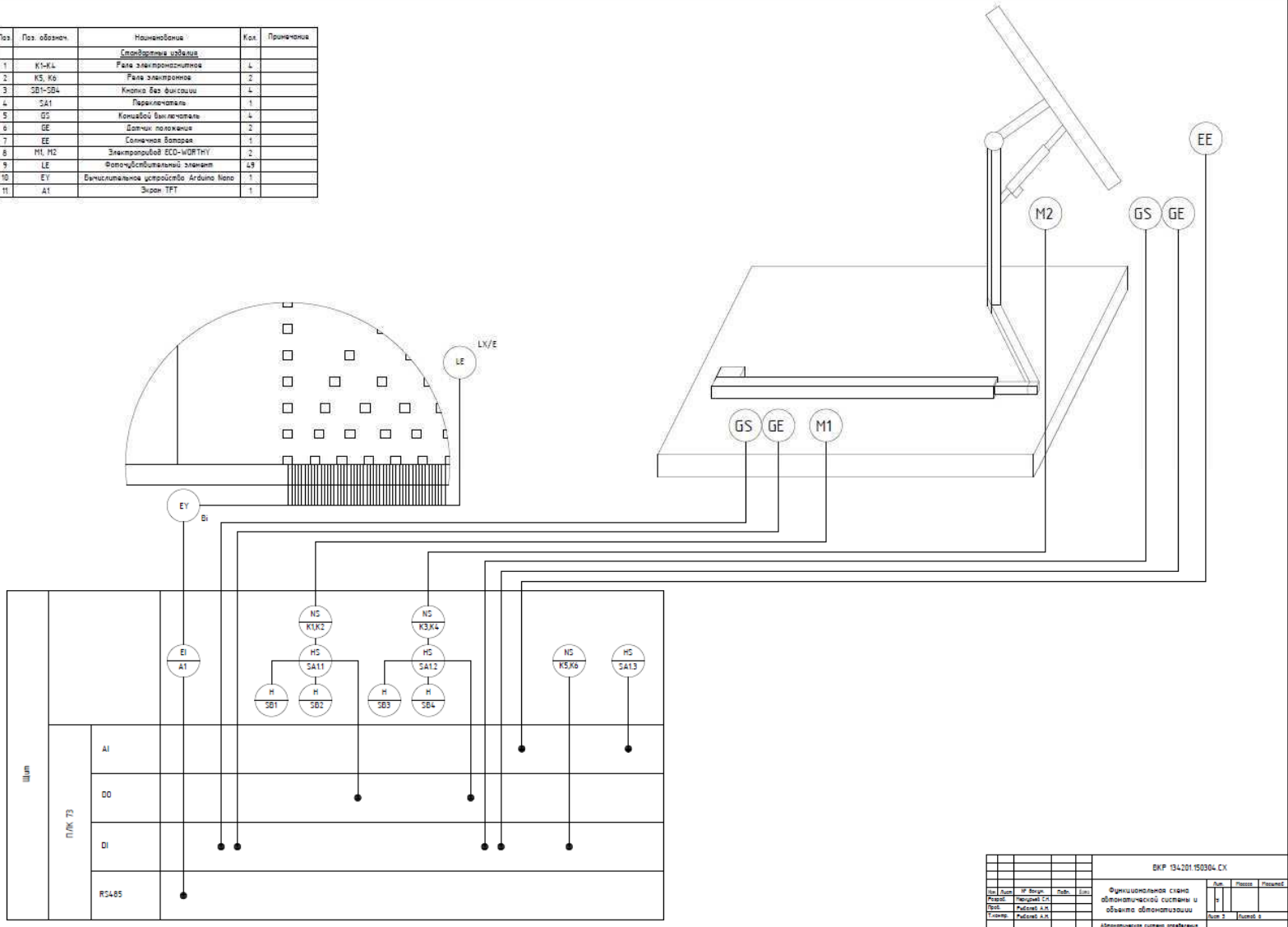
Структурная схема



					ВКР 134.201150304.08			
Имя	Фамилия	ИП	Воскр.	Дата	Лист	Кол-во	Итого	Итого
Исполн.	Иванов И.И.				1			
Проф.	Иванов И.И.							
Статус	Иванов И.И.							
Исполн.	Сидоров С.С.							
Проф.	Сидоров С.С.							
					Объект: Объект автоматизации			
					АвГУ 34.2сб			



Поз.	Поз. обознач.	Наименование	Кол.	Примечание
		Специальные изделия		
1	K1-K4	Реле электромеханические	4	
2	K5, K6	Реле электронные	2	
3	SB1-SB4	Кнопки без фиксации	4	
4	SA1	Переключатель	1	
5	GS	Контакты выключателя	4	
6	GE	Датчики положения	2	
7	EE	Солнечные батареи	1	
8	M1, M2	Электромоторы ЕСД-и/ДПТН	2	
9	LE	Фотоэлектрический элемент	1	ЛЭ
10	EY	Вычислительное устройство Arduino Nano	1	
11	A1	Экран TFT	1	

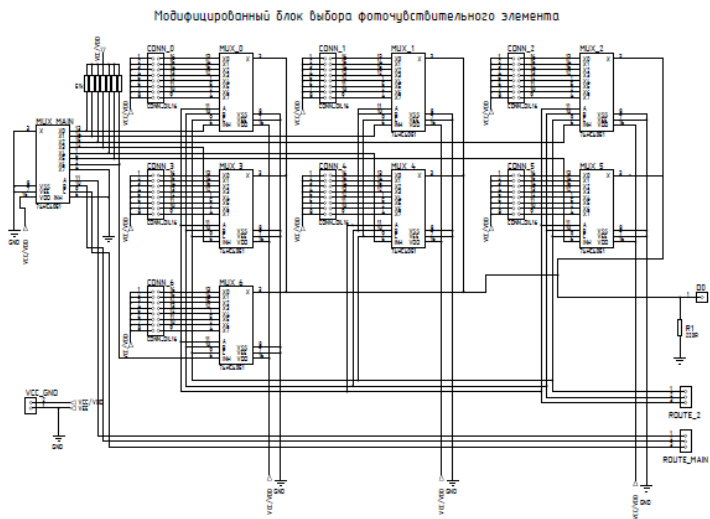
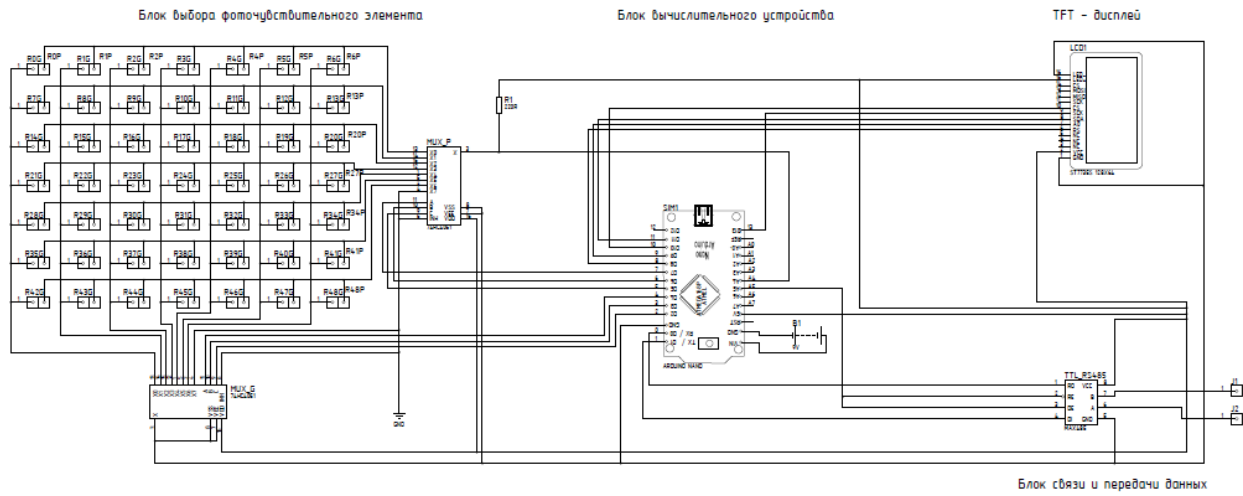


ДНП 34-201-150304.СХ					Изм.	Исполн.	Проверен.
№	Датум	ИП	Вариант	Лист	1	И	И
Исполн.	Проверен.	Дата	Листов	Кол-во			
Исполн.	Проверен.	Дата	Листов	Кол-во	Итого: 1		
Исполн.	Проверен.	Дата	Листов	Кол-во	Итого: 1		
Исполн.	Проверен.	Дата	Листов	Кол-во	Итого: 1		

Функциональная схема автоматической системы и объекта автоматизации

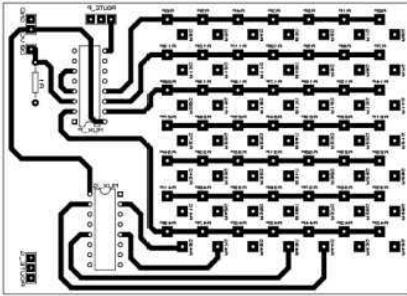
Автоматическая система управления технологическим процессом для производства полимерных пленочных батарей

ДНП 34-201-150304.СХ

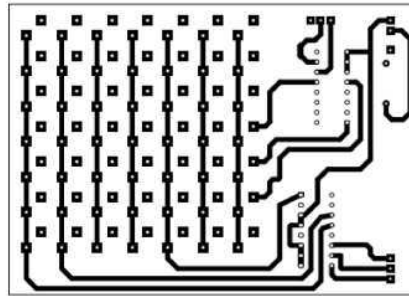


БКР 134201.150304.СХ					
№	Конт.	ИП	Возв.	Лист	Этаж
Рисунг	Модуль СХ				Страна сборки: Беларусь
Титло	Модуль СХ				Лист 4
Исполн.	Сидорин С.В.				Контракт №
Дат.	2014.04.04				Акт № 34/1сБ

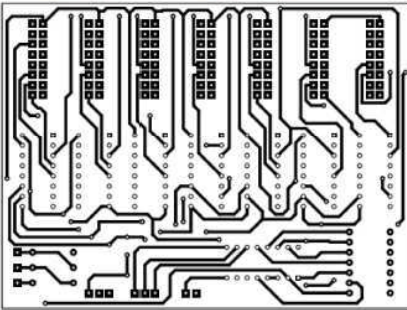
Верх изготовленной печатной платы. Зеркальное отображение



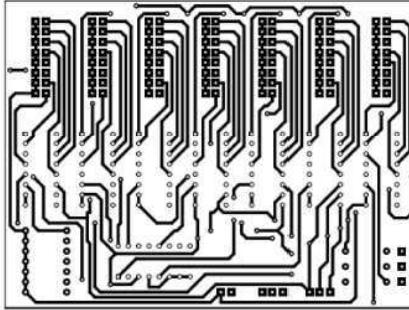
Низ изготовленной печатной платы. Обычный вид



Верх модифицированной печатной платы. Зеркальное отображение



Низ модифицированной печатной платы. Обычный вид



Процесс сборки системы

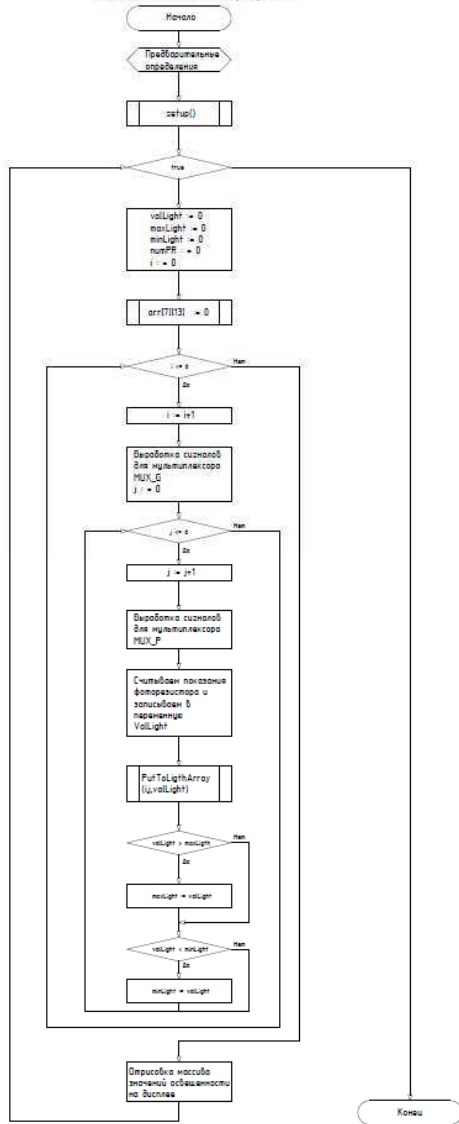


Общий вид изделия в штатном режиме работы

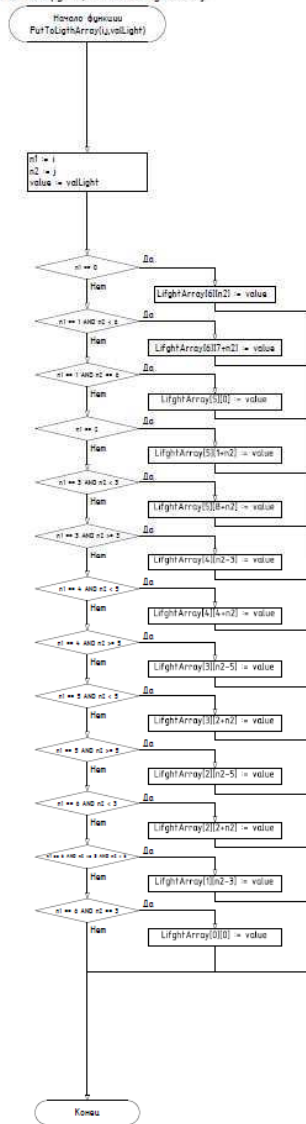


ВКР 134.20115030-08									
№ п/п	№ докум.	Исполнитель	Имя	Дата	Статус	Исполнитель	Имя	Дата	Статус
1	134.20115030-08	Михайлов С.И.				Михайлов С.И.			
2		Михайлов С.И.				Михайлов С.И.			
3		Михайлов С.И.				Михайлов С.И.			
Абсолютная система управления						Акт 34-15сб			

Блок-схема основной программы



Блок-схема функции PutToLightArray



Глобальные переменные программы

Наименование	Пояснение
RED_LIGHT	Преобразованный цвет
RED_MIDDLE	Преобразованный цвет
RED_DARK	Преобразованный цвет
BLUE_LIGHT	Преобразованный цвет
BLUE_MIDDLE	Преобразованный цвет
BLUE_DARK	Преобразованный цвет
GREEN_LIGHT	Преобразованный цвет
GREEN_MIDDLE	Преобразованный цвет
GREEN_DARK	Преобразованный цвет
BLACK	Преобразованный цвет
WHITE	Преобразованный цвет
rst	Выход дисплея - сброс
ds	Выход дисплея - данные
cs	Выход дисплея - clock sat
maxi	Выход дисплея для SPI
zclk	Выход дисплея для SPI
LightArray[7][13]	Массив значений освещенности
nt	Объем - дисплей

Локальные переменные программы

Наименование	Пояснение
valLight	Значения освещенности на выбранном датчике
maxLight	Максимальное значение освещенности после опроса всех датчиков
minLight	Минимальное значение освещенности после опроса всех датчиков
numPF	Номер резистора с максимальной освещенностью
colorEffect	Цвета для отображения текущей области на дисплее

Блок-схема алгоритма									
№	Дата	ИП	Взнос	Лист	Лист	Всего	Листов	Листов	Листов
Исполн:							1		
Провер:									
Утверд:									
Исполн:									
Провер:									
Утверд:									

Алгоритмическая схема обработки показаний датчика для микроконтроллера PIC16C62

Алг № 34.1a08

