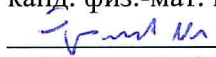


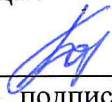
Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Радиофизический факультет

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
канд. физ.-мат. наук, доцент
 М.Л. Громов
« 09 » 06 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА
РАЗРАБОТКА МАКЕТА УСТАНОВКИ УЛЬТРАЗВУКОВОЙ СВАРКИ ПО
ТЕХНОЛОГИИ FLIP-CHIP

по направлению подготовки 03.03.03 Радиофизика
направленность (профиль) «Радиофизика, электроника и информационные системы»

Щербаков Александр

Руководитель ВКР
канд. физ.-мат. наук, зав. каф.
ИТИДиС
 С.Н. Торгаев
подпись
« 09 » 06 2023 г.

Автор работы
студент группы № 071906
 А. Щербаков
подпись
« 09 » 06 2023 г.

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Радиофизический факультет

УТВЕРЖДАЮ

Руководитель ООП

канд. физ.-мат. наук, доцент

Смирнов М.Л. М.Л. Громов

« 05 » 10 20 22 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы бакалавра обучающегося

Щербаков Александр

(Ф.И.О. обучающегося)

по направлению подготовки 03.03.03 Радиофизика, направленность (профиль)
«Радиофизика, электроника и информационные системы»

1 Тема выпускной квалификационной работы бакалавра

Разработка макета установки ультразвуковой сварки по технологии Flip-Chip

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:

а) на кафедре – 06.06.2023

б) в ГЭК – 12.06.2023

3 Исходные данные к работе:

Объект исследования – ультразвуковая сварка полупроводниковых структур по технологии Flip-Chip.

Предмет исследования – ультразвуковая сварка полупроводниковых структур

Цель исследования – разработка макета установки для автоматизации ультразвуковой сварки полупроводниковых структур по технологии Flip-Chip

Задачи:

Обзор технологии монтажа полупроводников

Разработка структуры макета установки УЗ сварки и реализация

Разработка конструкции нагревательного столика

Разработка и реализация схемы сбора температуры

Разработка и реализация системы позиционирования объекта

Разработка программного обеспечения

Проведение экспериментов

Методы исследования

Экспериментальные исследования, компьютерное моделирование

Организация или отрасль, по тематике которой выполняется работа

4. Краткое содержание работы

В процессе работы будет разработана схема макета установки, состоящая из нескольких модулей. Будет создана плата для системы управления с возможностью подключения к главному компьютеру для управления всеми компонентами установки. Будет создано ПО для управления системой позиционирования и сбора данных о температуре. Будет

создан графический интерфейс пользователя для управления установкой и возможность получать изображение с микроскопа в реальном времени.

Руководитель выпускной квалификационной работы

Зав. каф. ИТИДиС, НИ ТГУ

(должность, место работы)


(подпись)

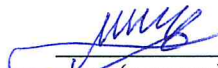
/ С.Н. Торгаев

(И.О. Фамилия)

Задание принял к исполнению

Студент, НИ ТГУ

(должность, место работы)


(подпись)

/ А. Щербаков

(И.О. Фамилия)

АННОТАЦИЯ

Выпускная квалификационная работа бакалавра содержит: 43 страницы, 4 главы, 38 рисунков, 2 таблиц, 54 страницы.

РАЗРАБОТКА МАКЕТА УСТАНОВКИ УЛЬТРАЗВУКОВОЙ СВАРКИ ПО ТЕХНОЛОГИИ FLIP-CHIP

Цель работы – разработка макета комплекса для автоматизации ультразвуковой сварки полупроводниковых структур по технологии Flip-Chip.

В процессе выполнения выпускной квалификационной работы было сделано следующее:

1. Проведен обзор технологии монтажа полупроводников.
2. Разработана структуры макета установки УЗ сварки и создана плата для системы управления.
3. Разработана конструкция нагревательного столика и проведены компьютерные моделирования нагрева.
4. Разработана и реализована схема сбора температуры.
5. Разработана и реализована система позиционирования объекта.
6. Разработано программное обеспечение для управления всеми система установки.
7. Разработан графический интерфейс пользователя с поддержкой многоязычного интерфейса.
8. Проведены эксперименты с получением изображения с цифрового микроскопа.
9. Проведены эксперименты с нахождением объекта на изображении.

Программы написаны на языках C++, Python с привлечением библиотек PyQt5, pyqtgraph, SimpleKeypad.h.

ОГЛАВЛЕНИЕ

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ СИМВОЛОВ, СОКРАЩЕНИЙ, ТЕРМИНОВ	6
ВВЕДЕНИЕ.....	7
1 Обзор методов монтажа полупроводников	8
1.1 Ультразвуковая сварка по технологии перевернутого монтажа.....	8
1.2 Виды межсоединений Flip-Chip.....	10
1.3. Технологии монтажа Flip-Chip	12
1.3.1 Термокомпрессионный монтаж	12
1.3.2 Ультразвуковой-термозвуковой монтаж.....	13
1.4 Обзор технологического оборудования Flip-Chip	13
2 Аппаратная часть разрабатываемой установки.....	15
2.1 Схема разрабатываемой установки	15
2.1.1 Плата системы управления.....	16
2.2 Разработка нагревательного блока	19
2.3 Разработка системы технического зрения	22
2.2.1 Цифровой микроскоп.....	22
2.2.2 Система совмещения и распознавания.....	23
2.4 Система позиционирования	23
3 Разработка алгоритмов работы и программного обеспечения	26
3.1 Система управления (Raspberry Pi)	26
3.1.1 Управление двумя SPI-устройствами.....	26
3.1.2 Взаимодействие Raspberry Pi с Arduino	27
3.2 Алгоритмы Arduino.....	28
3.3 Разработка графического интерфейса пользователя	30
3.3.1 Окно аутентификации пользователя	31
3.3.2 Главная страница и панель навигации программы	33
3.3.3 Вкладка блока нагрева	35
3.3.4 Дополнительные вкладки программы.....	37
3.3.5 Локализация интерфейса: многоязычная поддержка в программе	40
4 Экспериментальная часть.....	44
4.1 Эксперимент с цифровым микроскопом.....	44
4.2 Эксперимент с обработкой изображения и поиск объекта	45
ЗАКЛЮЧЕНИЕ	48
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ.....	50
Приложение А Qr Код на репозиторий программ	51
Приложение Б Чертежи нагревательного стола.....	52
Приложение В Схема подключения системы управления.....	53

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ СИМВОЛОВ, СОКРАЩЕНИЙ, ТЕРМИНОВ

Flip-Chip (перевернутый монтаж) – это процесс, при котором полупроводниковый кристалл присоединяют активной стороной вниз на подложку.

Анизотропные проводящие пленки (АПП) – это известные адгезивы, включающие проводящую составляющую и связующую полимерную смолу в виде пленки

COMSOL Multiphysics® - это универсальная платформа для моделирования реальных систем, устройств и физических процессов в разных отраслях промышленности.

Qt Designer – используется для проектирования и создания графических пользовательских интерфейсов с виджетом Qt.

Raspberry Pi - серия небольших одноплатных компьютеров, разработанных в Соединенном Королевстве на ARM-архитектуре и обладает небольшой ценой и маленькими габаритами

ВВЕДЕНИЕ

Технология ультразвуковой сварки уже давно применяется в промышленном производстве микроэлектроники. В настоящее время проволочное соединение все еще широко используется в качестве метода соединения в производстве микроэлектроники первого уровня. Поскольку процессы ультразвуковой сварки обладают уникальными преимуществами, в последние годы технология перевернутого монтажа (Flip-Chip) все чаще используется в производстве с низким количеством контактов, таких как: смарт-карты, светоизлучающие диоды (LED), микроконтроллеры и операционные усилители. Благодаря данному методу появляется возможность очень малой протяженности межкомпонентных соединений, что сокращает к минимуму величину их индуктивности.

Микроэлектроника, как подраздел электроники, имеет достаточно длительный путь становления, её корни уходят еще в советский период, но, к сожалению, дальнейшего развития с того времени не наблюдается в нашей стране. Соответственно, многие отечественные аппаратные комплексы ультразвуковой сварки полупроводниковых структур на данный период времени не соответствуют современным стандартам. Многие установки закупаются у иностранных компаний, так как отечественные разработки устарели.

С 2014 года наша страна взяла курс на импортозамещение и многие отрасли промышленности начали активно развиваться. В поисках замены иностранного оборудования было решено разработать отечественный комплекс для автоматизации процесса ультразвуковой сварки полупроводниковых структур по технологии Flip-Chip. Данная инновация станет очевидным толчком в развитии микроэлектронной промышленности России.

Актуальность данного исследования заключается в создании простого и точного управления процессом сварки полупроводников, улучшении качества сварки, уменьшении процентов выбраковки при монтаже кристаллов на печатные платы, экономии средств на производство продукции и развитие микроэлектроники в РФ, что обусловлено потребностью государства в совершенствовании данной отрасли.

1 Обзор методов монтажа полупроводников

1.1 Ультразвуковая сварка по технологии перевернутого монтажа

Технология Flip-Chip, также известная как C4 (Controlled Collapse Chip Connection) — контролируемое соединение чипа методом самовыравнивающейся пайки), представляет собой метод межсоединения полупроводниковых устройств, таких как интегральные микросхемы и микроэлектромеханические системы (МЭМС). Технология Flip-Chip позволяет соединять контактные площадки кристаллов и подложек с помощью адгезионных паст, контактных шариков, медных столбиков или столбиков припоя [1]. Этот вид монтажа оптимизирует отвод тепла от изделий силовой электроники с большим количеством контактов, формируя подходящую топологию медных дорожек на подложке [2]. Таким образом, часть тепла будет выведена в окружающую среду через бампы, а их количество — обратно пропорционально эквивалентному термическому сопротивлению соединительного слоя.

Применение Flip-Chip-технологии монтажа кристалла в корпус методом перевернутого кристалла стало настоящим прорывом, позволившим добиться уменьшения габаритов и улучшения характеристик изделий, значительно снизить омическое сопротивление и индуктивность контактных соединений на частотах порядка 100 МГц (таблица 1) [3].

Таблица 1- Электрические параметры Flip-Chip- и проволочного монтажа

Параметр	Проволочный монтаж	Flip-Chip
Диаметры, мкм	Проволока 30	Шарик, 80
R, мОм	133	2,8
L, нГн	1,785	0,0002
C, пФ	18,8	14,8

В качестве альтернативного метода проволочному монтажу кристалла с контактными площадками корпуса или платы применяют столбиковые либо шариковые выводы. Кристалл в данном случае монтируется лицевой стороной к подложке (Flip-Chip) либо обычным способом, но со сквозными выводами (Through Silicon Vias, TSV). Такие способы монтажа позволяют распределить выводы по всей поверхности кристалла

микросхемы, существенно экономя площадь конечного устройства по сравнению с проволочным монтажом (см. Рисунок 1).

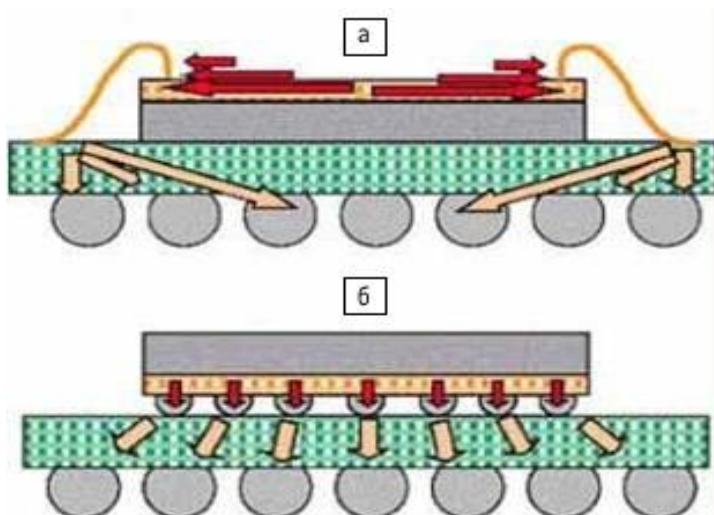


Рисунок 1 - Распределение тепловых потоков в BGA-корпусах: а) при монтаже проволочкой; б) при Flip-Chip-технологии [1]

За счет данного метода появляется возможность очень малой протяженности межкомпонентных соединений и это сокращает к минимуму величину их индуктивности. [5]. Наиболее широко эта технология применяется в модулях смарткарт, микроконтроллерах с количеством выводов до 200 и выше.

Технология Flip-Chip для процессоров и микросхем памяти постоянно растет в компьютерной и телекоммуникационной отраслях благодаря следующим преимуществам:

- повышение плотности выводов, так как шарики могут быть размещены по всей области чипа, а не только по периферии;
- самоцентрирующий шаг процесса в результате поверхностного натяжения припоя;
- возможность уменьшить размер кристалла для ранее ограниченных типов контактной площадки;
- возможность отведения тепла из перевернутого кристалла через внешнюю металлическую крышку с использованием термопасты;
- возможности распределения мощности, в связи с тем, что цепь в центре чипа напрямую доходит к источнику энергии;
- высокая производительность и низкая стоимость монтажа;
- снижение длины проводов и уменьшение количество общих проводов.

1.2 Виды межсоединений Flip-Chip

В настоящее время применяется множество способов для технологии перевёрнутого монтажа (см. Рисунок 2). Как правило, выводы полупроводника могут быть выполнены из шариков оловянно-свинцового припоя (см. Рисунок 2б), смеси высокотемпературного и низкотемпературного припоя (см. Рисунок 2а) или даже из шариков бессвинцового припоя (золотой, медный и т. д.). Применяются также методы, в которых используются специальные адгезии (см. Рисунок 2г–е). Они используют изотропный, анизотропный проводящий и непроводящий клей.

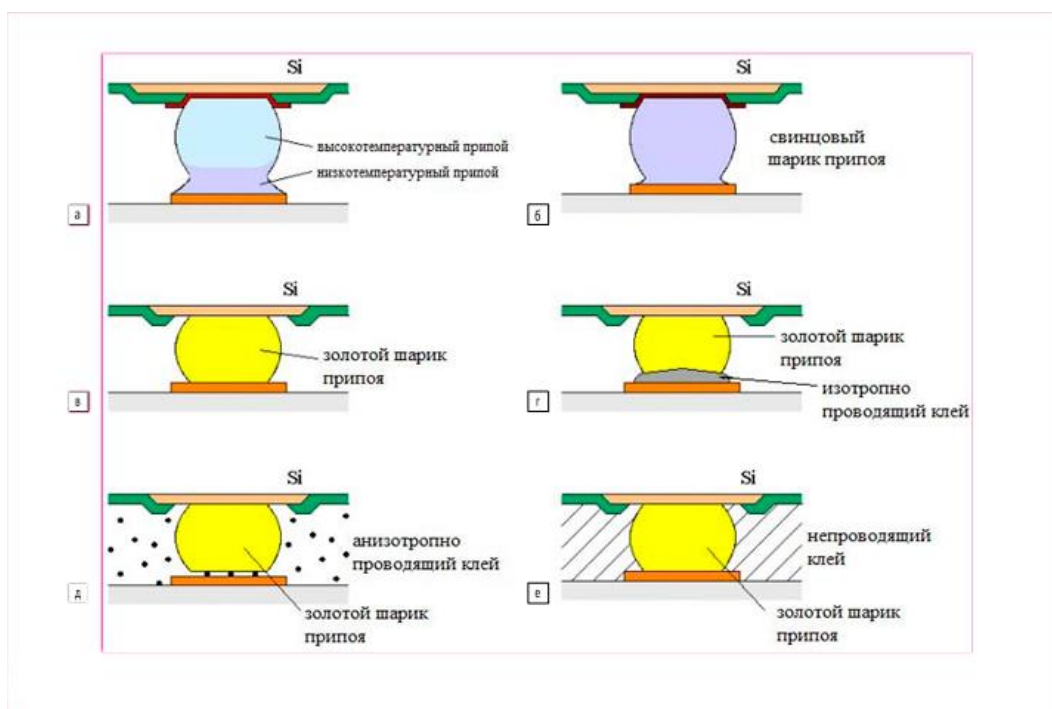


Рисунок 2 - Виды межсоединений Flip-Chip: а) на основе смеси высоко- и низкотемпературного припоев; б, в) шариков припоя; г) изотропного и д) анизотропного проводящего адгезива; е) непроводящего адгезива. [2]

Уникальный проводящий адгезив содержит проводниковые частицы, которые в определенных условиях образуют контакт между площадками кристалла и подложки.

Хотя технология монтажа Flip-Chip применяется уже около 50 лет, ее распространение в производстве электроники довольно ограничено. Основной недостаток метода — это более низкие тепловые характеристики, чем у полупроводника, соединенным обычным способом, а также трудности при герметизации матрицы контактных площадок.

Ограничения частично связаны с неполной совместимостью SMD компонентов с доступным оборудованием для их монтажа. Кроме того, использование свинцового припоя

становится причиной выброса альфа-частиц, что может привести к выходу из строя схемы в чувствительных цепях, таких как компьютерная память. Этот эффект можно минимизировать, ограничивая размещение припоя над чувствительными областями.

Методы присоединения кристаллов по технологии Flip-Chip включают ультразвуковую и термозвуковую сварку, присоединение с помощью анизотропных или изотропных проводящих клеев, пайку оплавлением. Для применений с мелким шагом припой может быть нанесен гальваническим способом, струйной пайкой или в виде твердого припоя. Липкий флюс наносится на области контакта с припоем либо погружением чипа в резервуар для флюса или распылением на подложку. При грубой обработке паяльная паста наносится на подложку методом трафаретной печати. Бампы чипов помещаются в липкую пасту и оплавляются в печи. После оплавления предпочтительна очистка от флюса. Герметизирующий материал наносится дозированием вдоль одной или двух сторон чипа, при этом эпоксидная смола низкой вязкости втягивается капиллярными силами в пространство между чипом и подложкой, а затем отверждается под воздействием тепла.

Существует несколько решений для формирования на кристаллах шариков, обеспечивающих Flip-Chip-монтаж. Для создания контактных шариков размером порядка 100–120 мкм гальваническим методом требуется магнетронное нанесение барьерного слоя Ti или TiW толщиной 0,1–0,2 мкм на всю поверхность пластины с последующим напылением меди толщиной 0,3–0,8 мкм. Слой Ti/TiW-Cu представляет собой наиболее распространенный вариант так называемой подбамповой металлизации (Under Bump Metallurgy, UBM). Далее на пластину наносится 40-мкм слой фоторезиста и выполняется операция фотолитографии для вскрытия окон к UBM. Далее производится гальваническое наращивание приблизительно 5-мкм слоя меди с дальнейшим заполнением гальваническим припоем. Для получения требуемого объема припойного шарика гальваническое осаждение материала продолжается до высоты 15–20 мкм над уровнем слоя фоторезиста с формированием характерной грибовидной структуры, показанной на рисунок 3а. После этого производится плазменное или жидкостное химическое удаление фоторезиста и слоя UBM с последующим оплавлением шариков и формированием припойных шариков на пластине (рисунок 3б) [5].



Рисунок 3 - Типовая структура бампов типа С4 на кремниевом кристалле: а) до удаления фоторезиста; б) после удаления фоторезиста и оплавления [2]

1.3. Технологии монтажа Flip-Chip

1.3.1 Термокомпрессионный монтаж

Для термокомпрессионного Flip-Chip монтажа на контактных площадках полупроводника предварительно сформированы золотые шарики диаметром 70 мкм. Затем проводится совмещение контактных выводов кристалла и контактных дорожек подложки с последующим размещением кристалла на ней. При этом подложка предварительно нагревается до +200 °С. После установки кристалла вся сборка нагревается по заданному температурному профилю до максимального значения +350 °С, и в момент максимального значения температуры осуществляется прижим кристалла к подложке (рисунок 4). Значения требуемого усилия прижима зависят от размера кристалла и количества контактных выводов. Типовые значения усилия прижима находятся в диапазоне 0,1–0,7 Н/бамп [4]. Главный недостаток этой технологии — высокая пиковая температура термического цикла. Стандартная РСВ-подложка из стеклотекстолита FR4 может деформироваться при температурах выше +200 °С [5].

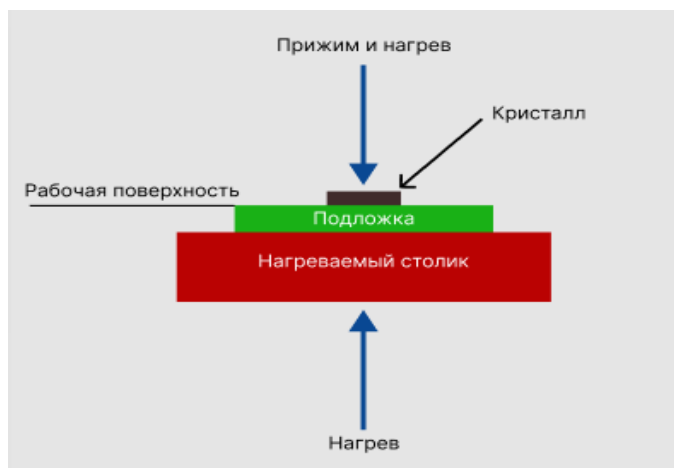


Рисунок 4 - Термокомпрессионный Flip-Chip монтаж

1.3.2 Ультразвуковой-термозвуковой монтаж

Технология Flip-Chip-монтажа с помощью ультразвука схожа с технологией термокомпрессионного монтажа (рисунок 4), но в ходе процесса сборки «кристалл-подложка» не подвергается воздействию столь высоких температур, поскольку в этом нет необходимости. На первом этапе проводится предварительный нагрев подложки до температуры около $+120\text{ }^{\circ}\text{C}$. Затем происходит выравнивание и установка полупроводника на подложку. После чего осуществляется прижим кристалла к подложке с усилием около $0,06\text{ Н/бамп}$, и подается ультразвуковой импульс (порядка 1400 мВт) с одновременным увеличением температуры до $+150\text{ }^{\circ}\text{C}$ [5]. Благодаря ультразвуковым колебаниям удается достичь электрического контакта между шариками кристалла и дорожками подложки при более низких температурах по сравнению с термокомпрессионным монтажом. Однако процесс требует прецизионного контроля параметров монтажа. С одной стороны, слишком большое усилие прижима и высокая мощность ультразвукового импульса могут разрушить соединения между кристаллом и подложкой. С другой — чересчур низких усилий прижима и ультразвуковой мощности может быть просто недостаточно для получения качественного соединения. Кроме того, применение специально разработанного инструмента типа «пирамидка» для захвата кристалла с двух сторон не накладывает ограничений на передачу ультразвуковой мощности в отличие от обычного плоского вакуумного инструмента.

1.4 Обзор технологического оборудования Flip-Chip

Технологическое оборудование для монтажа по технологии Flip-Chip (Рисунок 5) содержит: блок управления, ультразвуковой генератор (УЗГ), блок нагрева инструмента, платформу вертикального перемещения сварочной головки, предметный столик с нагревом, сварочную головку, полупрозрачное зеркало, микроскоп и пневмоцилиндр нагружения [7]. С помощью предметному столику, можно выполнять точные микроперемещения по координатам x , y , z , благодаря винтовых пар качания. Часть стола без подогрева используется как предметный для захвата полупроводников. Кристаллы, находящиеся на предметном столике россыпью, захватываются вакуумом с помощью инструмента, закрепленного в сварочной головке.

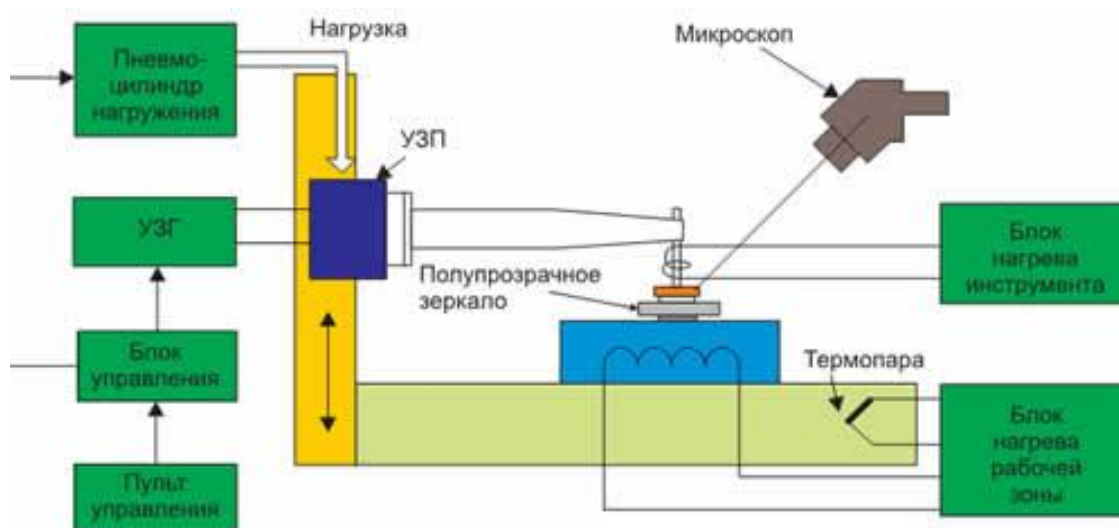


Рисунок 5 – Схема установки для монтажа по технологии Flip-Chip [2]

После подведения, подогреваемого столика под инструмент с захваченным лицевой стороной вниз кристаллом между кристаллом и подложкой вводится полупрозрачное зеркало, предварительно отъюстированное по резкости изображения. Наблюдая в микроскоп одновременное изображение фрагментов топологии кристалла и подложки, полученное через полупрозрачное зеркало, производится совмещение с помощью ручек регулировки координат столика. После совмещения полупрозрачное зеркало возвращается в исходное положение, а сварочная головка, на которой расположен инструмент с кристаллом, опускается по координате z до срабатывания датчика касания, по сигналу которого осуществляется нагружение пневмоцилиндром и подача импульса УЗГ.

В обеспечение качества присоединения предварительно проверяется параллельность торца инструмента относительно поверхности столика, на котором устанавливается подложка. Для этих целей используется тонкая фольга, на которой делается отпечаток инструмента. Отпечаток должен иметь четкий контур торца инструмента. Выставление параллельности относительно подложки осуществляется регулировкой столика.

Основными технологическими параметрами, определяющими процесс группового монтажа кристаллов с объемными выводами, являются: температура подогрева зоны соединения; усилие сжатия в зоне соединения; время присоединения; частота и мощность УЗ-колебаний. При монтаже кристаллов, содержащих 122 объемных микровывода, на кремниевую подложку определены оптимальные режимы процесса: температура нагрева в зоне присоединения $+320...+360\text{ }^{\circ}\text{C}$; время присоединения 30-50 мс; усилие нагружения 60 Н; мощность УЗГ 15 Вт; частота колебаний УЗГ $66\pm 4\text{ кГц}$. Присоединение достигалось равномерным прижимом и небольшой выдержкой приборов после касания для стабилизации температуры в контактной области.

2 Аппаратная часть разрабатываемой установки

2.1 Схема разрабатываемой установки

Разработка установки для ультразвуковой сварки – это сложный процесс, включающий в себя большую цепь электрических узлов (драйверы двигателей, датчики положений, блоки нагрева, система технического зрения и т.д.). Главная задача автоматизировать все процессы монтажа полупроводниковых структур. Для реализации данной установки, разделено на несколько задач:

1. Разработка нагревательного стола;
2. Конструирование источника питания;
3. Прецизионный модуль монтажа;
4. Создание ультразвукового генератора;
5. Создания система технического зрения (СТЗ);
6. Создание системы управления (СУ).

Для построения системы управления выбрана комбинация Raspberry Pi и микроконтроллера Arduino NANO, которые взаимодействуют по интерфейсу (I2CInter-Integrated Circuit). Raspberry Pi - это мощный одноплатный компьютер, который обладает широкими возможностями в области вычислений и связи. Он является основным управляющим устройством системы, предоставляя высокую вычислительную мощность и гибкость для выполнения различных задач. Raspberry Pi обеспечивает удобный интерфейс пользователя и связь с внешними устройствами, включая микроконтроллер Arduino NANO. Arduino NANO был выбран в качестве основного микроконтроллера для платы управления. Он обладает достаточной вычислительной мощностью и обширной поддержкой программного обеспечения, что делает его идеальным выбором для реализации системы автоматизации. Микроконтроллер играет центральную роль в системе, обеспечивая эффективное взаимодействие с другими модулями и обработку полученных данных. Он выполняет несколько важных функций, которые способствуют гармоничной работе компонентов системы.

Во-первых, микроконтроллер обеспечивает связь и управление драйверами шаговых двигателей. Он принимает команды от главного компьютера и передает соответствующие сигналы драйверам, контролируя точное позиционирование и движение шаговых двигателей.

Кроме того, микроконтроллер собирает данные от различных датчиков, таких как термодатчики, датчики давления, положения и касания. Он анализирует эти данные,

осуществляет их обработку и передает важную информацию главному компьютеру для дальнейшего анализа и принятия решений.

Микроконтроллер также активно участвует в управлении нагревательным столом и механизмами нагружения. Он получает соответствующие команды от главного компьютера и контролирует процессы нагрева и механического нагружения, обеспечивая точность и стабильность в работе этих компонентов.

В качестве системы технического зрения выступает цифровой микроскоп с увеличением в 1600 раз. Все вышесказанное можно свести к схеме, показанной на рисунке 6 и отражающей принципиальный состав разрабатываемого макета установки Flip-Chip.

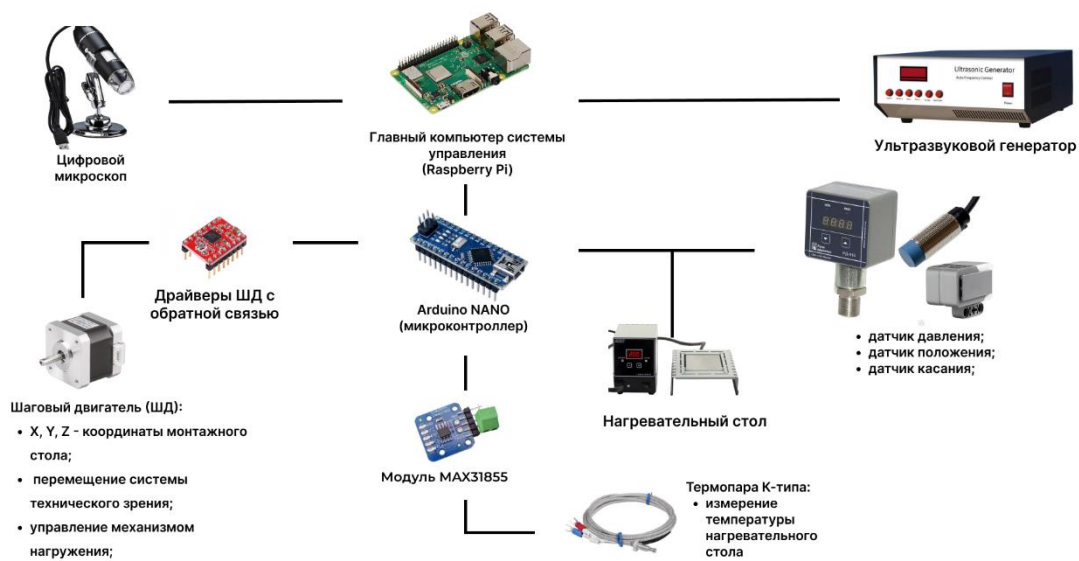


Рисунок 6 – Схема разрабатываемой установки Flip-Chip

2.1.1 Плата системы управления

Была разработана плата блока управления, которая играет ключевую роль в функционировании системы. Плата включает в себя различные компоненты, обеспечивающие эффективное управление и контроль системы автоматизации. Центральным элементом разработанной платы управления является модульное подключение, которое обеспечивает простоту и гибкость при подключении различных компонентов. Благодаря модульной концепции, пользователь может легко заменить датчики и контроллеры в случае их выхода из строя. На рисунке 7 изображено физический макет платы (лицевая часть) с нумерацией элементов платы, а на рисунке 8 можно ознакомиться с задней частью платы.

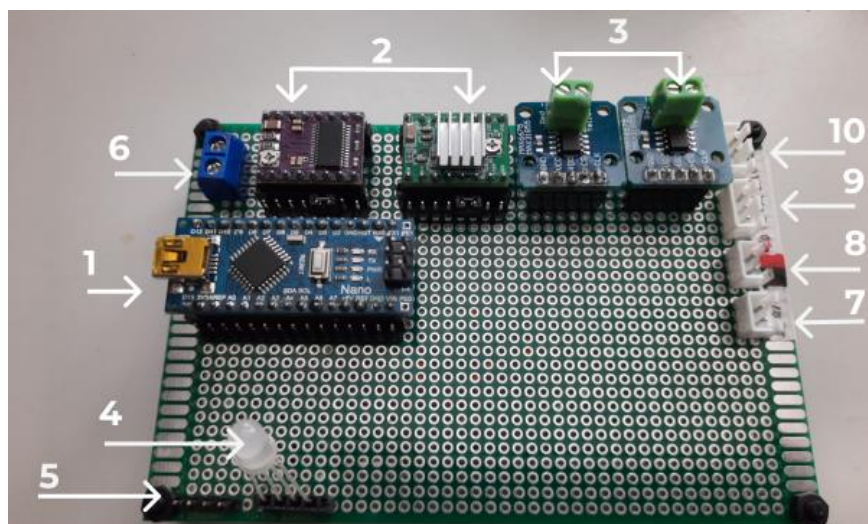


Рисунок 7 - Плата блока управления (лицевая часть): 1 - Микроконтроллер Arduino NANO; 2 - Драйверы для управлений шаговыми двигателями; 3 - Модули MAX31855 для считывания температуры с термодатчиков; 4 - Светодиод для отслеживания двигателей; 5 - Штекеры для подключения матричной клавиатуры; 6 - Питания 12 В для шаговых двигателей; 7 - Гнездо для подключения к Raspberry Pi 4 по интерфейсу I2C; 8 - Питания платы блока управления; 9 - Гнездо для подключения термопары №1; 10 - Гнездо для подключения термопары №2

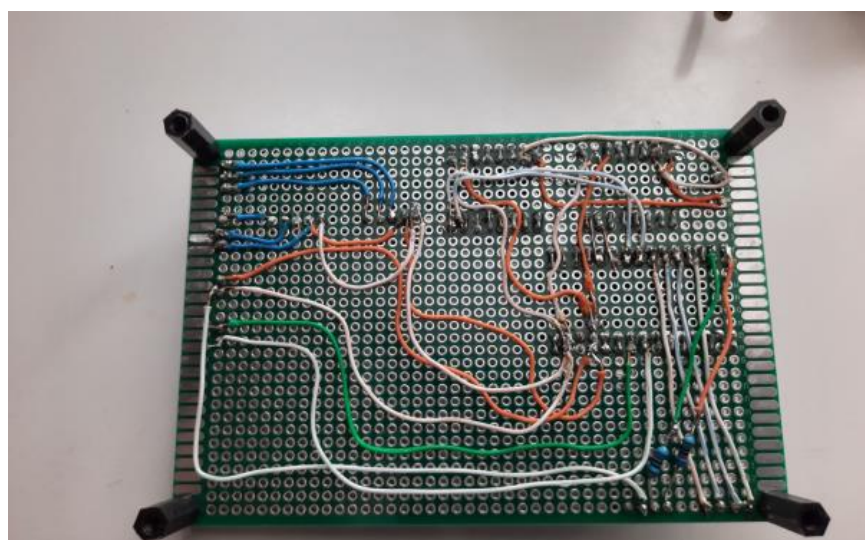


Рисунок 8 - Плата блока управления (задняя часть)

Плата блока управления включает в себя драйверы для управления шаговыми двигателями. Эти драйверы обеспечивают точное позиционирование и управление двигателями, обеспечивая необходимую точность и стабильность в работе системы.

Для считывания температуры с термодатчиков используются модули MAX31855. Они обеспечивают точный и надежный сбор данных о температуре, что позволяет системе эффективно контролировать и регулировать процессы нагрева.

Светодиод на плате предназначен для отслеживания работы двигателей. Он предоставляет визуальную обратную связь о состоянии двигателей, что помогает оператору контролировать процессы и в случае необходимости принимать меры.

Штекеры на плате предоставляют возможность подключения матричной клавиатуры. Это позволяет оператору вводить команды и управлять системой с помощью клавиш, обеспечивая удобство и гибкость взаимодействия с системой.

Для питания шаговых двигателей на плате предусмотрено 12 В. Это обеспечивает необходимую мощность и стабильность работы двигателей, что является важным аспектом в системе управления.

Плата блока управления также имеет гнездо для подключения к Raspberry Pi 4 по интерфейсу I2C. Это позволяет обмениваться данными и координировать работу с Raspberry Pi, расширяя возможности и функциональность системы.

Для обеспечения питания платы блока управления на плате предусмотрено соответствующее питание. Это обеспечивает стабильность работы. Для ознакомления со схемой подключения стоит обратиться к приложению А

2.2 Разработка нагревательного блока

На первом этапе разработки нагревательного блока было проведено моделирование прототипа нагревательного стола с целью сокращения времени и снижения затрат при создании физического прототипа. Результаты модельных исследований позволили определить конструкцию нагревательного стола, материалы для корпуса и метод нагрева. Для компьютерного моделирования был выбран программный продукт COMSOL Multiphysics, который позволяет проводить моделирование физических процессов. Для электротермического анализа тепла использовался мультифизический модуль AC/DC, который включает в себя следующие возможности:

1. Joule Heating (Джоулев нагрев);
2. Heat Transfer in Solids (Теплопередача в твердых телах);
3. Electric Currents (Электрические токи).

С использованием модуля AC/DC проводились исследования нагрева, основанные на токах проводимости и диэлектрических потерях. Для этой цели была создана трехмерная модель нагревательного стола непосредственно в программном обеспечении COMSOL Multiphysics. Благодаря различным геометрическим операциям и инструментам, была разработана итоговая конструкция стола, которая представлена на рисунке 9.

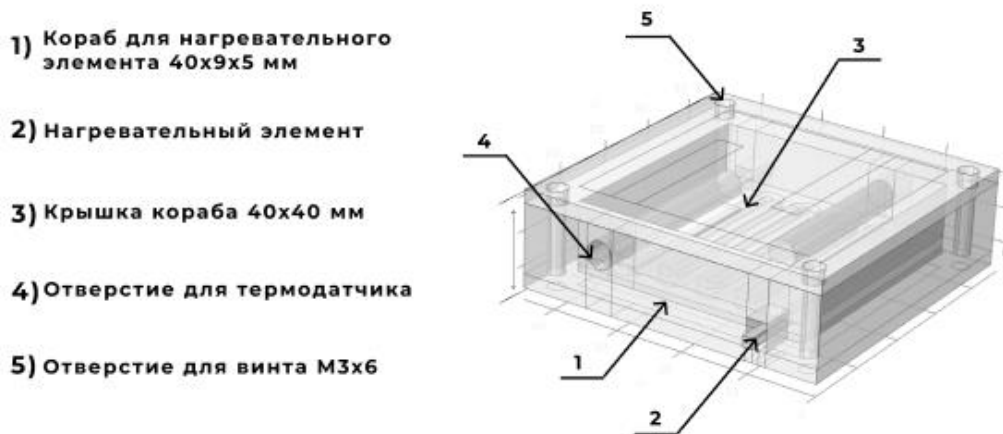


Рисунок 9 – Конструкция нагревательного стола

В качестве нагревательного элемента выбрана нихромовая нить сплава Х20Н80. Данный сплав обладает отличными характеристиками:

1. Жаростойкость;
2. Рабочая температура до 1200 С°;
3. Температура плавления 1400 С°.

Для модельного исследования диаметр проволоки составляла 1,5 мм, а длина 2 м. В короб для нагревательного элемента проволока расположена «змейкой» (Рисунок 10).

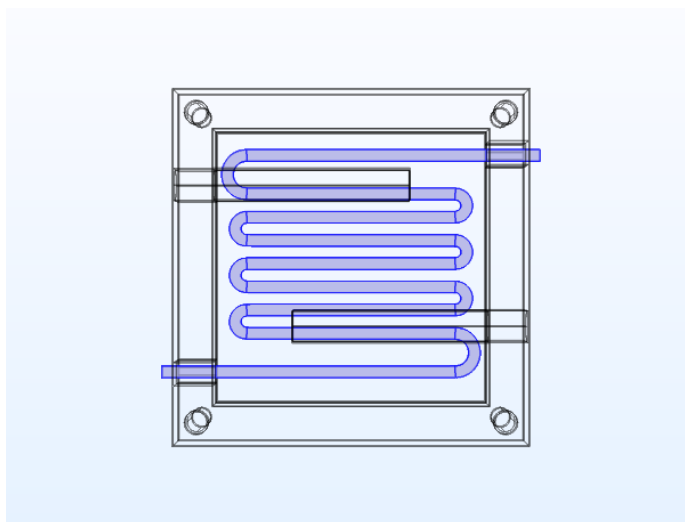


Рисунок 10 - Укладка проволоки «змейкой»

Материал для корпуса нагревательного стола выбрана нержавеющая сталь. Этот материал обладает несколькими преимуществами:

1. Высокая прочность;
2. Продолжительность эксплуатации, без потери первоначальных характеристик;
3. Стойкость к коррозии.

После подачи напряжения 30 В на нихромовую нить и проведения компьютерного моделирования было обнаружено равномерное распределение тепла по всей рабочей поверхности нагревательного стола (Рисунок 11). В результате 20-минутного нагрева, в центральной точке рабочей поверхности была достигнута температура 315°C.

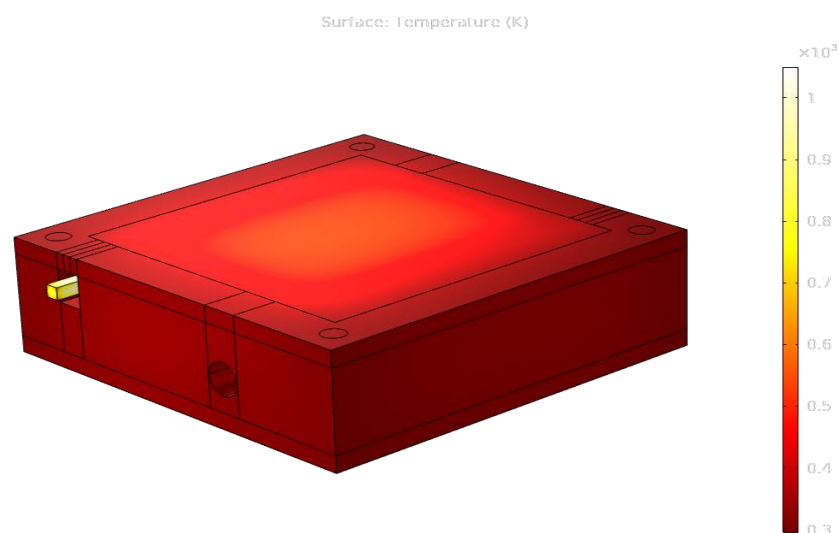


Рисунок 11 - Распространение нагрева

В результате проведения модельных исследований была разработана конструкция термостабилизированного стола, предназначенного для установки ультразвуковой сварки. Были выбраны материалы и определены размеры нагревательного элемента. Компьютерные исследования показали, что для достижения рабочей температуры нагревательной поверхности необходимо применять напряжение, не превышающее 30 В, в течение 20 минут нагрева. Для нагревательного элемента была выбрана нихромовая нить, а для корпуса - нержавеющей сталь. На основе проведенного исследования был составлен чертеж нагревательного стола, который представлен в приложение Б.

Для считывания температуры нагревательного стола и последующей регулировки температуры используется термопара ТС-K-TYPE-5М-30mm. Данная термопара принадлежит к типу К (хромель-алюмель) и предназначена для измерения температур в широком диапазоне от 0 до 400°C. Она обладает высокой точностью и надежностью в измерении температуры.

Для считывания данных с термопары и их передачи контроллеру используется модуль MAX31855, изображенный на рисунке 12. Этот модуль подключается к контроллеру с помощью шины SPI (Serial Peripheral Interface) и обеспечивает высокоточное и стабильное считывание данных о температуре с термопары. Модуль MAX31855 предоставляет информацию о температуре в цифровом формате, что позволяет контроллеру проводить регулировку и управление температурой нагревательного стола на основе полученных данных.

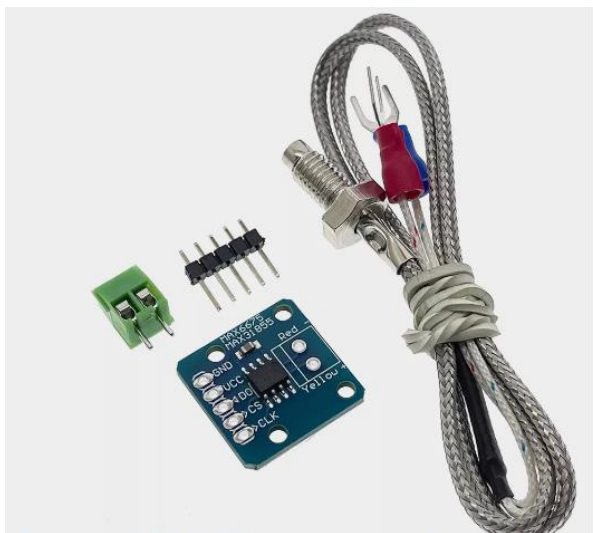


Рисунок 12 – термопара К-типа и модуль MAX31855

2.3 Разработка системы технического зрения

2.2.1 Цифровой микроскоп

На первом этапе разработки системы компьютерного зрения используется цифровой микроскоп с подключением через USB и максимальным увеличением 1600X.

USB микроскоп 1600X – это компактное современное устройство с цифровой камерой с разрешением 0.3 Мп, позволяющей снимать видео или делать фотографии исследуемых предметов с максимальным 1600-кратным оптическим увеличением. Микроскоп подключается к Raspberry Pi через USB порт и не требует дополнительных источников питания.

Порядок работы с микроскопом 1600X:

1. микроскоп устанавливается в штатив,
2. включается и настраивается подсветка,
3. под объектив кладётся исследуемый предмет,
4. наводится фокус.

На начальных этапах эксперимента микроскоп управляется вручную, но в последующих этапах планируется осуществить переход к управлению через главный компьютер Raspberry Pi. Для ознакомления с микроскопом и его компонентами рекомендуется обратиться к рисунку 13, где представлена визуальная демонстрация микроскопа и его функциональных элементов.



Рисунок 13 – Схема деталей USB микроскопа 1600X

2.2.2 Система совмещения и распознавания

К основополагающим относится система совмещения и распознавания. Именно ей необходимо уделять больше внимания, поскольку в случае некачественного совмещения при посадке кристалла с большим числом выводов на основание с малыми контактными площадками результатом будет отсутствие электрических соединений. Стандартом точности позиционирования Flip-Chip машин является точность 5–10 мкм [8]. Исходя из этих требований, спроектированы несколько систем совмещения и распознавания, в основе которых лежит цифровой микроскоп. Наиболее распространенной системой позиционирования Flip-Chip является DIS (система двойного изображения).

Система сложна и построена на основе призм, обычных и полупрозрачных зеркал для разделения луча (рисунок 14). Изображение лицевой стороны кристалла и подложки преломляются и попадают в объектив микроскопа, сигнал от микроскопа обрабатывается ПО и отображается на видеомониторе, при этом два изображения отображаются с разной степенью прозрачности, что позволяет совместить «бампы» и контактные площадки подложки. Резкость и увеличение можно регулировать линзой и управлять шаговым двигателем в режиме микрошага и расположенной между разделителем луча и фокусирующей линзой.

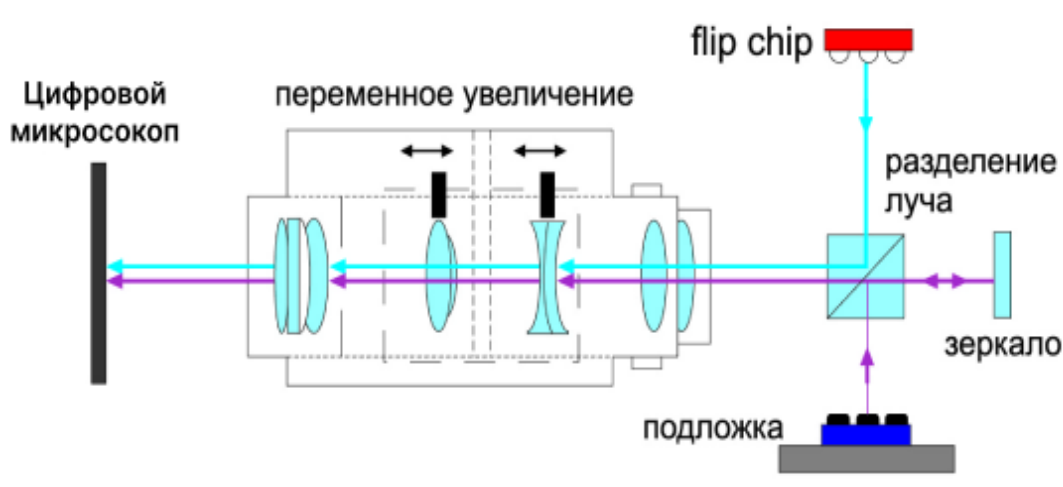


Рисунок 14 - Оптический тракт DIS

2.4 Система позиционирования

Для перемещения цифрового микроскопа по осям X и Y применяются два шаговых двигателя NEMO-17. Эти двигатели относятся к типу постоянного тока и обеспечивают перемещение ротора по малым дискретным шагам. Работа шагового двигателя NEMO-17 основана на том же принципе, что и у других шаговых двигателей. Данный двигатель имеет 6 подводных проводов и работает при напряжении 12 В. Возможно также его

функционирование при более низком напряжении, но при этом уменьшится крутящий момент. При каждом шаге ось двигателя NEMO-17 поворачивается на угол в 1.8 градуса. Расположение подводящих проводов двигателя NEMO-17 представлено на рисунке 15.

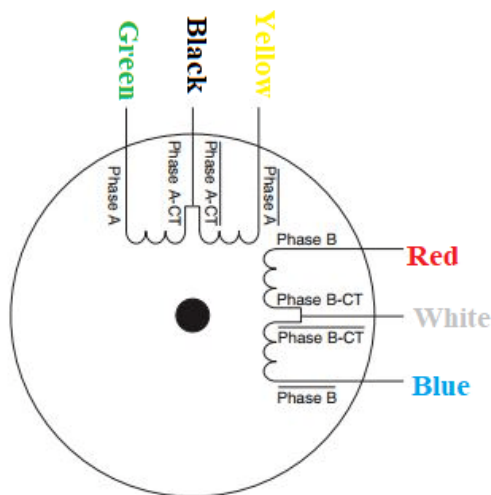


Рисунок 15 - Расположения подводящих проводов двигателя NEMO-17

По представленному рисунку видно, что шаговый двигатель NEMO-17 имеет 6 проводов с униполярным расположением. Эти провода подключены к двум отдельным обмоткам. Черный, желтый и зеленый провода соединены с первой обмоткой, а красный, белый и синий провода - с другой обмоткой. В обычном режиме центральные провода обмоток (черный и белый) остаются неподключенными. Число шагов на полный оборот (Steps per Revolution) для каждого конкретного шагового двигателя рассчитывается на основе угла, на который двигатель поворачивается за один шаг (step angle). Для двигателя NEMO-17 этот угол составляет 1.8 градуса.

Полный оборот = $360 / \text{шаг угла}$

$360 / 1.8 = 200$ Полный оборот.

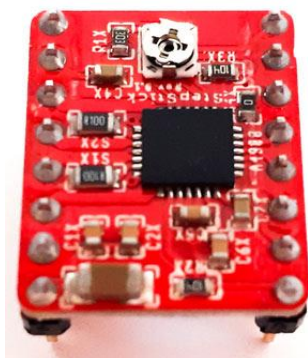


Рисунок 16 – Модуль драйвера шагового двигателя A4988

Модуль драйвера шагового двигателя A4988 (рисунок 16) предназначен для управления шаговым двигателем NEMO-17 и другими аналогичными биполярными шаговыми двигателями. Он оснащен встроенным транслятором, который позволяет контролировать и скорость, и направление вращения шаговых двигателей, используя небольшое количество контактов.

Для управления движением шаговых двигателей используется мембранная матричная клавиатура 1x3. На клавиатуре имеются три кнопки, каждая из которых выполняет определенную функцию. Первая кнопка отвечает за переключение между двигателями №1 и №2, позволяя выбирать, какой из них будет управляться. Вторая и третья кнопки отвечают за управление направлением движения. Нажатие на вторую кнопку осуществляет движение вверх, а нажатие на третью кнопку - вниз. Таким образом, пользователь может выбрать нужный двигатель и указать желаемое направление его движения с помощью указанных кнопок на мембранной клавиатуре.

Физический прототип, изображенный на рисунке 17, демонстрирует подключение шагового двигателя NEMO-17 к драйверу A4988 и матричной клавиатуре. Для управления этой системой в эксперименте использовалась отладочная плата Arduino NANO. Отладочная плата Arduino NANO выполняет функцию контроллера, обеспечивая взаимодействие между драйвером A4988, матричной клавиатурой и шаговым двигателем NEMO-17. Это позволяет контролировать движение шагового двигателя с помощью клавиатуры, предоставляя удобный и простой способ управления системой.

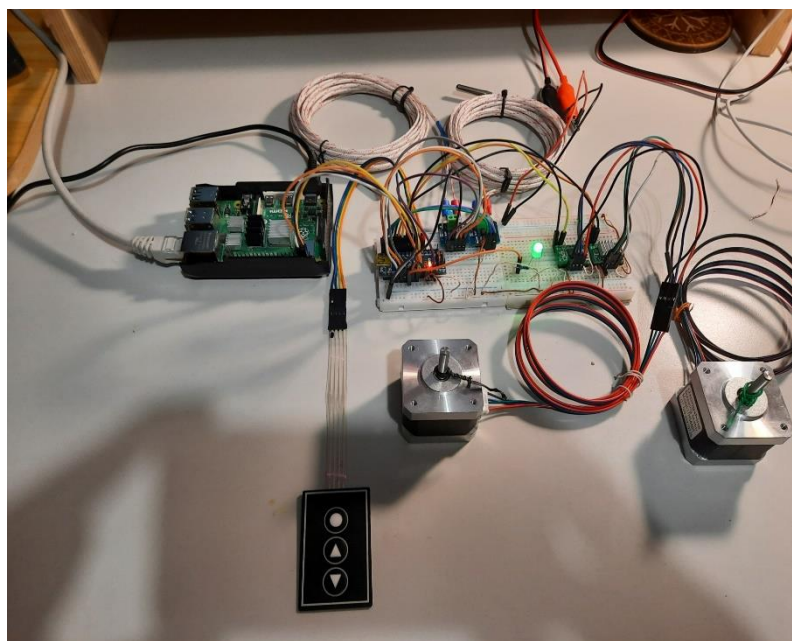


Рисунок 17 – Физический прототип системы позиционирования

3 Разработка алгоритмов работы и программного обеспечения

3.1 Система управления (Raspberry Pi)

В настройке, над которой мы работаем, Raspberry Pi служит главным компьютером для системы управления. Этот одноплатный компьютер отвечает за управление всеми системами и обработку команд оператора. Подобно обычному компьютеру, Raspberry Pi оснащен различными периферийными устройствами и интерфейсами. Он может выполнять множество функций, сравнимых с мощными настольными системами, и даже больше. Raspberry Pi обладает значительными графическими возможностями и может быть использован для разработки пользовательского программного обеспечения и автоматизации управления всеми системами комплекса.

На начальных этапах нашего проекта мы установили ОС Raspbian, которая основана на Debian и специально разработана для Raspberry Pi. Эта операционная система поставляется с предустановленным набором программного обеспечения, которое облегчает разработку программного обеспечения для установки. Все программы, написанные в данной работе представлены в приложении А в виде Qr-Code для перехода в репозиторий.

3.1.1 Управление двумя SPI-устройствами

Для управления двумя микроконтроллерами по интерфейсу SPI разработана программа на языке программирования Python. С помощью библиотеки `spidev` мы можем управлять двумя устройствами MAX31855 по интерфейсу SPI. Данные контроллеры предназначены для считывания данных с термопар. В начале программы мы определяем настройки для двух выводов, предназначенных для взаимодействия по интерфейсу SPI - это SPI0 и SPI1. Оба вывода настраиваются на режим работы 0 (CPOL = 0, CPHA = 0) и максимальную скорость передачи данных 5 МГц. Затем создаются объекты `spi0` и `spi1` относящихся к классу `Spidev`, у нас открываются соответствующие порты (SPI0 CE0 и SPI1 CE1) и устанавливаются заданные настройки, которые нам нужны. Команда чтения температуры из MAX31855 задается из 4 байтов ([0b00000001, 0b00000000, 0b00000000, 0b00000000]). Эта команда помогает передавать для получения данных о температуре.

В программе функция `read_temp()` ответственна за чтение температуры с первого датчика MAX31855, используя объект `spi0`. С помощью метода `xfer2(cmd)` команда чтения отправляется по SPI0, и полученные данные сохраняются в переменной `resp_1`. Затем происходит преобразование полученных данных в температуру: первые 14 битов объединяются в значение `temp1`, и если старший бит равен 1, то значение `temp1`

корректируется. Возвращается значение температуры temp1, умноженное на 0.25 для получения конечного результата.

Аналогично функция read_temp1(spi1) осуществляет чтение температуры со второго датчика MAX31855, используя объект spi1. Данные получаются из spi1.xfer2(cmd), и после преобразования в температуру возвращается результат temp2. Таким образом, программа позволяет получать данные о температуре с двух датчиков MAX31855, подключенных к различным SPI-портам (SPI0 и SPI1), и выполняет преобразование этих данных в удобный формат. Это позволяет использовать программу для мониторинга и контроля температуры в различных приложениях и системах.

3.1.2 Взаимодействие Raspberry Pi с Arduino

Одной из главной задачей разработки макета установки ультразвуковой сварки по технологии Flip-Chip, является создания взаимодействия Raspberry Pi с Arduino, чтобы создать возможность подключение большего количество датчиков и устройств для наших задач. Для задачи реализации взаимодействия главного микроконтроллера с главным компьютером. Для этой цели выбран интерфейс взаимодействия устройств между собой под названием Inter-Integrated Circuit (I2C). I2C является последовательным двухпроводным интерфейсом коммуникации между микроконтроллерами, микросхемами и другими электронными устройствами. Он был разработан компанией Philips (сейчас NXP Semiconductors) и стал широко распространенным стандартом для обмена данными между устройствами. Он использует две линии для связи.

Первая линия связи называется Serial Data Line (SDA). SDA - это двунаправленная линия связи, по которой передаются данные между двумя и более устройствами. Каждое устройство, подключенное к шине, имеет свой уникальный адрес и данные передаются в формате пакетов.

Вторая линия связи это – синхронизационная. Имеющая название Serial Clock Line (SCL). Это линия синхронизации ответственна за передачу тактового сигнала, контролирующая передачу данных между устройствами. Сигналы на линии SCL определяет время передачи данных и синхронизируют работу подключенных устройств.

Протокол I2C имеет возможность поддерживать подключения многих устройств, подключая к одной шине. При этом каждый микроконтроллер имеет уникальный семи битный или десяти битный адрес, которое помогает идентифицировать его на шине. Передача данных осуществляется в виде пакетов данных, состоящих из стартового бита (Start), бита адреса, бита чтения или записи, байтов данных и бита подтверждения (ACK/NACK).

Главные особенности интерфейса I2C это - возможность подключать несколько «мастер» устройств, которые могут отдавать команды омы устройствам. При появлении конфликтов между главными устройствами применяется правила арбитража для определения приоритета доступа к шине. Так же мы можем подключать много и других устройств. Интерфейс поддерживает подключение нескольких устройств, что позволяет создавать нам сложные системы с различными периферийными устройствами для определённых задач.

Данный интерфейс активно используется разработчиками для подключения микроконтроллеров, таких как датчиков температуры, датчиков касания, датчиков давления и ЖК-дисплеев. Интерфейс обеспечивает простую и эффективную взаимосвязь между различными устройствами в системах, что делает его отличным вариантом для реализации взаимодействия Raspberry Pi и Arduino, которые поддерживают данный интерфейс.

Для программной реализации взаимодействия между контроллером и главным компьютером выполнено благодаря библиотеки `smbus`. В начале программы в переменной `SLAVE_ADDRESS` мы устанавливаем адрес Arduino, который имеет значение `0x04`. Благодаря команде `self.bus.write_byte (self.SLAVE_ADDRESS, ord('u'))`, мы отправляем команду микроконтроллеру для движения шагового двигателя. В программе реализованы четыре функции с данными командами, которые отвечают за управлениями двигателями.

1. Функция `def moveUp` – отдает команду двигателю делать поворот вперед;
2. Функция `def moveDown` – поворачивает первый двигатель назад;
3. Функция `def moveLeft` – поворачивает второй двигателей влево;
4. Функция `def moveRight` – поворачивает второй двигатель вправо.

В итоге данный алгоритм программы позволяет в будущем расширить функционал программы, для того чтобы подключать больше устройств по интерфейсу I2C, для управления ими и получения определённых данных. Программа для приёма команд и их обработка для Arduino, описаны в главе [3.2](#).

3.2 Алгоритмы Arduino

Программа для управления устройствами подключенных к Arduino написан на языке программирования Arduino, он предназначен для управления движением шаговых двигателей с возможностью управления через мембранную матричную клавиатуру и запросов от Raspberry Pi.

В начале программы подключаются необходимые библиотеки и объявляются глобальные переменные. Затем определяются порты для управления двумя шаговыми

двигателями, порты для подключения термопары и порты для светодиодов. Создается экземпляр клавиатуры и инициализируются порты.

В программе `motor1` – это функция, отвечающая за управления первым двигателем. В данной функции расположена переменная `motor1_active` в которой устанавливается значение `true`, показывая активность первого двигателя. Функция `set_motor1_direction()` используется для установки направления вращения вала первого двигателя. С помощью функции `digitalWrite()` устанавливаем состояние порта.. Функция `step_motor1()` выполняет один шаг первого двигателя. С помощью функции `digitalWrite()` устанавливается состояние порта `STEP1_PIN` в `HIGH` для выполнения шага в определенном направлении, затем с помощью задержки `delayMicroseconds()` устанавливается пауза, а затем порт `STEP1_PIN` устанавливается в `LOW` для завершения шага. В зависимости от параметра «forward» вызывается функция `set_motor1_direction ()` для установки направления движения, а затем в цикле выполняется заданное количество шагов.

Принцип работы второго двигателя аналогичен алгоритму работе первого двигателя, отличие только в использование соответствующих функций и портов для движения.

Так же в программе есть функция `toggleLED()`. Она отвечает за переключение состояния светодиодов. С помощью `digitalWrite()` устанавливаются состояния портов `Led1` и `Led2` в соответствии с значениями переменных `Led_1_On` и `Led_2_On`. Переключение светодиодов позволяет отслеживать активность определенных двигателей.

Функции `processMessage()` и `sendAnalogReading()` ответственны за взаимосвязь Arduino и Raspberry Pi. Они обрабатывают команды, полученные от главного компьютера через шину I2C и так же отправляют данные с датчиков главному компьютеру при соответствующем запросе. В зависимости от принятой команды, вызываются функции, которые ответственны за движением двигателей.

Управления матричной клавиатуры реализуется благодаря библиотеки `SimpleKeypad`. Создаются переменные `byte colPins[KP_COLS]` и `byte rowPins[KP_ROWS]` в которых происходит инициализация штекеров по порядку для матричной клавиатуры 1x3. Данные переменные определяют количество строк и столбцов клавиатуры. Функция `getKey()` находится в бесконечном цикле и считывает нажатия кнопки. Затем в зависимости от нажатой кнопки выполняются соответствующие действия: переключение между двигателями, движение вперед или назад. Переключение между первым и вторым двигателем происходит благодаря кнопки №1, после нажатия на данную кнопку происходит инвертирования значений в переменных `motor1_active` и `motor2_active`. Параллельно происходит инверсия переменных ответственных за переключения состояния светодиодов.

В итоге программа, написанная на языке Arduino, позволяет управлять двигателями нажатием соответствующей кнопкой, так же обрабатывать команды главного компьютера для управления системой позиционирования и возможность отправлять данные с датчиков, которые в будущем будут подключены.

3.3 Разработка графического интерфейса пользователя

Не отомленной частью разработки макета установки является создание пользовательского интерфейса для управления и контроля процессов работы установки. Для реализации данной задачи используется язык программирования Python и библиотеки PyQt5. Данная библиотека имеют привязку к фреймворку Qt. PyQt5 позволяет разработчикам создавать приложения для рабочего стола с богатым функционалом графических интерфейсов, используя язык программирования Python. Она даёт обширный набор инструментов и модулей для создания окон, диалоговых окон, кнопок, меню и других графических компонентов. Библиотека также поддерживает обработку событий, механизм сигналов и слотов для взаимодействия между объектами, а также интеграцию с различными другими возможностями и библиотеками Qt. Она предоставляет разработчикам удобный и мощный инструментарий для проектирования, разработки, отладки и развертывания приложений. На рисунке 18 представлено окно данной среды разработки.

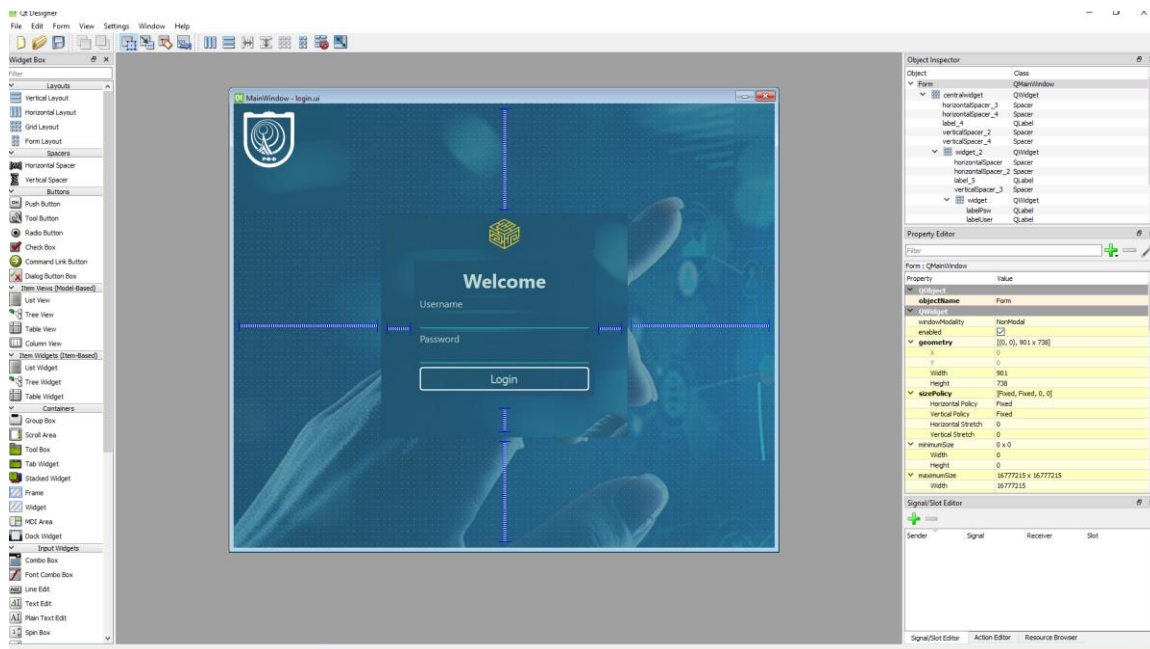


Рисунок 18 - Окно среды разработки Qt Designer

Создание макета интерфейса происходит перетаскиванием элементов и инструментов в окно будущего приложения. После создания макета интерфейса устанавливаются обработчики событий и связи между элементами интерфейса, при

необходимости. В последующем сохраняется созданные интерфейсы в файлы формата ui, которые затем могут быть загружены и использованы в будущей разработки приложений. В последующем так же можно данный файл преобразовать в формат .py, получив тем самым программный код на языке Python. Данная операция выполняется через консоль командой “pyuic5 mydesign.ui -o mydesign.py”. Главное – это производить конвертацию файлов в той же директории, где находятся файлы.

3.3.1 Окно аутентификации пользователя

Создание страницы входа с логином и паролем для доступа к программному обеспечению управляющей установкой для монтажа кристаллов по технологии Flip-Chip. Данная страница направлена на обеспечение безопасности управлением установкой и процессом работы с ней. Если пользователь не имеет логина и пароля для входа, то он не сможет запустить установку и работать с ней. Это позволяет избежать вмешательства в процесс работы монтажа, так как несанкционированный пользователь не мог навредить установке через ПО. На рисунке 19 представлен начальная страница входа, где располагается поля для ввода логина и пароля.

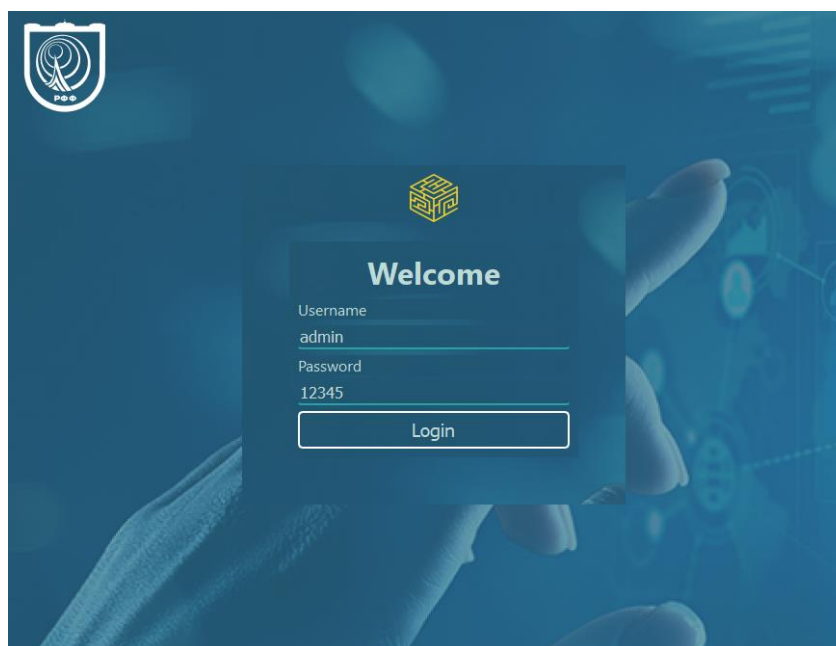


Рисунок 19 - Страница аутентификации пользователя

На рисунке видно, что пользователь вводит логин «admin» и пароль «1234» и после нажимает кнопку Login. После нажатия на кнопку событие обрабатывается в функции def loginBtn_clicked, в которой также происходит проверка логина и пароля. При случае введенного неверного логина или пароля, QMessageBox.warning создает окно с

предупреждением о не верных введенных данных. На рисунке 20 продемонстрирована работа уведомления о не верных данных.

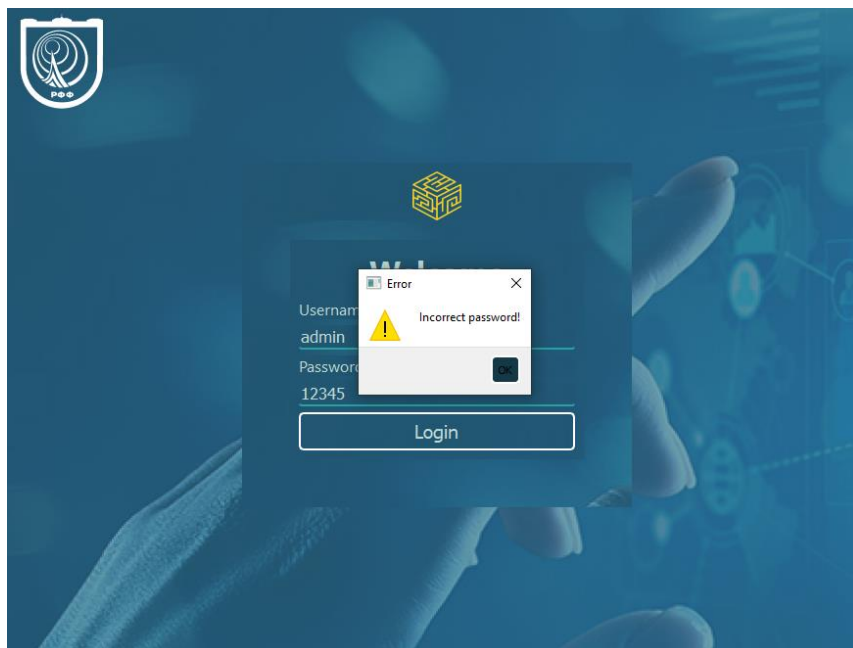


Рисунок 20 – Окно предупреждения о неверных введенных данных пользователем

В случае правильного введенного логина и пароля, вызывается `self.close()` ответственная за закрытие окна входа и функция `self.main_window.show()` вызывает уже главное окно графического интерфейса, в котором располагаются компоненты для управления системой (рисунок 21)

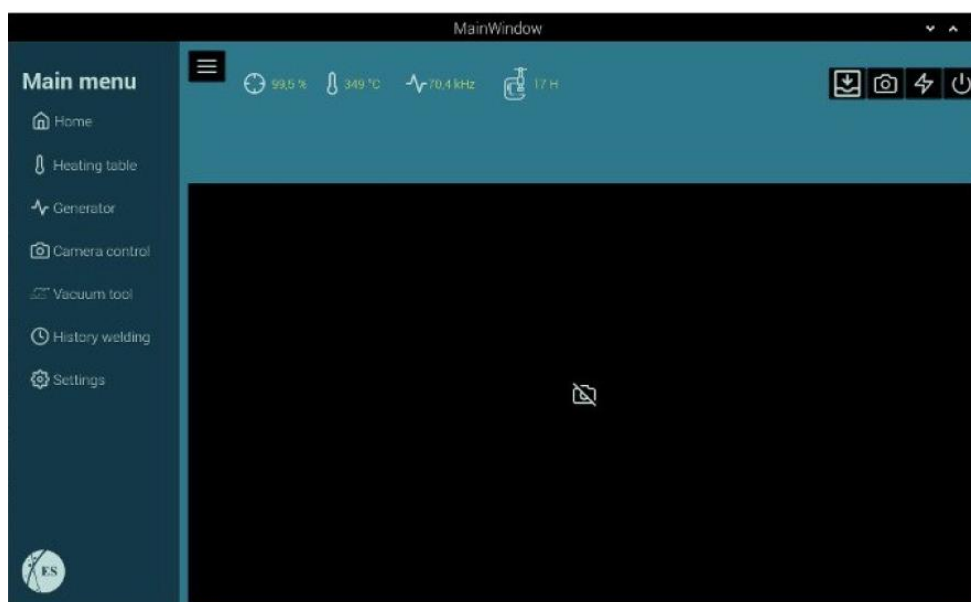


Рисунок 21 – Главное окно управления установки

3.3.2 Главная страница и панель навигации программы

Программа графического интерфейса пользователя представляет собой комплексное приложение с графическим интерфейсом, разработанное для управления микроскопом и выполнения различных операций. Рисунок из главы 3.3.1 демонстрирует окно, встречающее пользователя после его аутентификации. В данном окне слева расположена панель навигации, благодаря которой появляется возможность удобного переключения между вкладками приложения. Так же эту панель навигацию можно свернуть, сделав её меньше нажав на кнопку изображенной в виде трех горизонтальных полосок. Свертывания панели навигации происходит благодаря данной строчки программного кода `self.buttonMenu.toggled['bool'].connect(self.leftLiteMenu.setVisible)`, она обрабатывает событие нажатия на кнопку и в последующем происходит свертывания окна. На рисунке 22 демонстрируется результат после обработки события нажатия на кнопку `buttonMenu`.

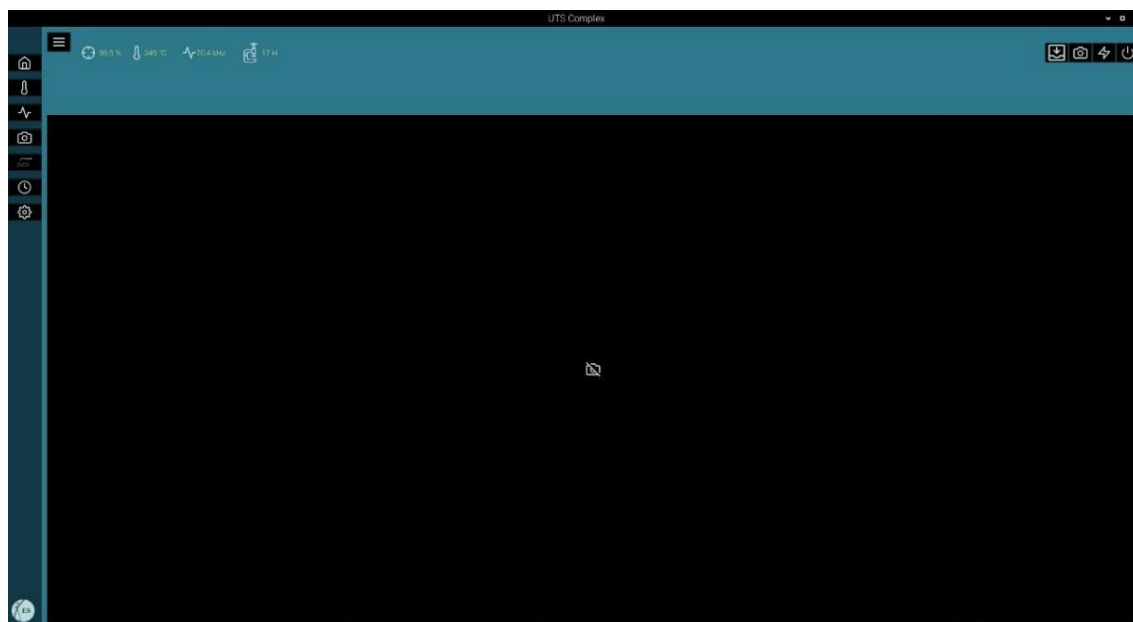


Рисунок 22 – демонстрация результата свертывания панели навигации

Панель навигации имеет кнопки, которые позволяют производить переход между вкладками благодаря функции `self.stackedWidget.setCurrentIndex(0)`, где в скобках указывается номер вкладки. При нажатии соответствующей кнопки, происходит переход. Каждая вкладка ответственна за определенную часть управления установкой. В таблице 1 можно ознакомиться с функциями и индексами вкладок.

Таблица № 1- Функции и индексы вкладок

Название вкладки	Функция	Индекс вкладки
Домашняя страница (Home)	on_buttonHome_toggled	0
Блок нагрева (Heating table)	on_buttonTable_toggled	1
Генератор (Generator)	on_buttonGenerator_toggled	2
Управление камерой (Camera control)	on_buttonCameraControl_toggled	3
Вакуумный инструмент (Vacuum tool)	on_buttonVacuum_toggled	4
История процесса (History welding)	on_buttonHistory_toggled	5
Настройки (Settings)	on_buttonSettings_toggled	6

На рисунке 22 так же можно увидеть черное окно, в котором в последующем будет транслироваться изображение с цифрового микроскопа в реальном времени. В файле под названием camera_control.py описывается программным кодом работа с захватом изображения с цифрового микроскопа и его транслирования в графическом интерфейсе пользователя. Вызов функции захвата изображения происходит после обработки события нажатия на кнопку. За обработку нажатия на кнопку отвечает функция `self.cameraOn.clicked.connect(self.start_camera)`. На рисунке 23 демонстрируется полученное изображение с микроскопа.

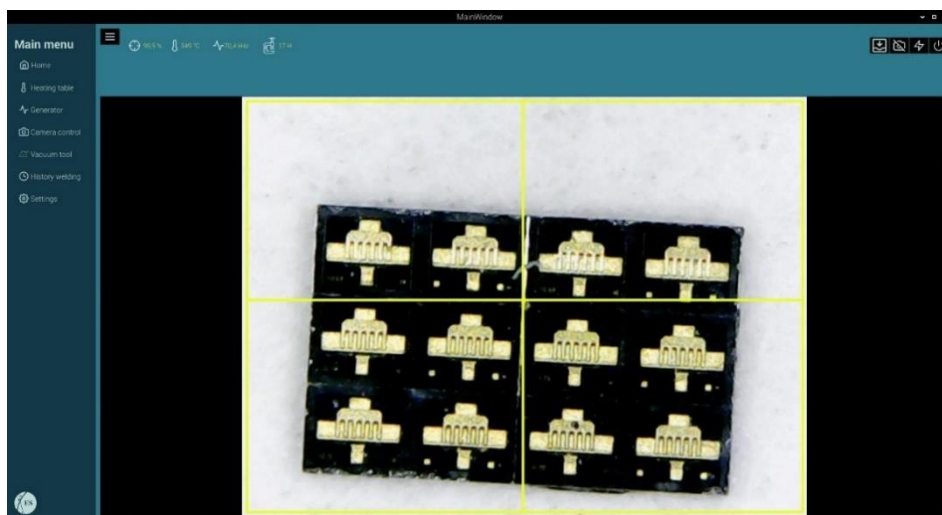


Рисунок 23 – Результат захвата изображения с цифрового микроскопа

3.3.3 Вкладка блока нагрева

Программа имеет вкладку ответственна за управлением блоком нагрева, в который входит нагревательный стол и датчики температуры. На рисунке 24 изображена данная вкладка. В ней расположено окно, в котором после нажатия кнопки Start Heating вызываются функции `def read_temp` и `def read_temp1`. Данные функции ответственны ха сбор данных с термопар, алгоритм сбора описано в главе 3.1.1.

После начало получения данных с термопар начинается построение графика температуры в реальном времени благодаря библиотеки `pyqtgraph`. Алгоритм построения графика описан в функции `def graph_temp`. Внутри функции происходит следующие операции. Сначала получается текущий момент времени в формате `timestamp` с помощью функции `QDateTime.currentDateTime().toSecsSinceEpoch()`. Далее значения времени добавляются в список `self.data['x']`, который содержит значения времени на оси x графика. Далее, температурные значения, хранящиеся в переменной `temp3` добавляется в список `self.data['y']`, который содержит значения температуры на графике по оси y.

Для обновления графика в реальном времени используется функция `setData`, которая принимает два аргумента: список значений времени на оси x и на оси y. Дополнительно, функция `setXRange` вызывается для настройки оси x графика так, чтобы она оставалась видимой в течение последних 60 секунд. Она принимает два аргумента: начальное значение времени и текущие значение времени по оси x.

Таким образом, функция `graph_temp` обновляет данные графика температуры, а также помогает настраивать ось x для отображения последних данных. Это и позволяет отслеживать изменение температуры в реальном времени на графике. Рисунок 24 демонстрирует работу функции.



Рисунок 24 – Построение графика температуры

Помимо построения графика, у пользователя есть возможность сохранять полученные графики в формате png (рисунок 25). Для этого используется функция `def saveGraph`, в ней настраивается размеры сохранения изображения и путь к папке в которой сохраняется картинка.

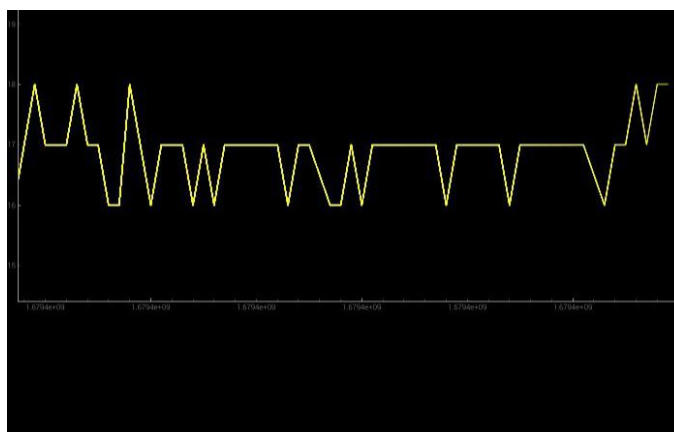


Рисунок 25- График температуры от времени

Вкладка блока нагрева дополнительно даёт возможность пользователю отслеживать температуру каждого датчика температуры. Отображение данных о температуре происходит с помощью функций `label_12` и `label_14`. Обновление отображаемых значений происходит благодаря декоратору `@QtCore.pyqtSlot()`, он ответственен за перемещение функцию `def request_reading` в слот, которая отображает данные в окне программы в реальном времени.

Название датчика и значение температуры выделено зеленым цветом, что говорит пользователю о их корректной работе. В случае выхода из строя одно из датчика или

отсутствие данных срабатывает условие и вызывается функция QMessageBox(). После вызова функции, пользователю выплывает окно предупреждающие его о проблемах с датчиком (рисунок 26).

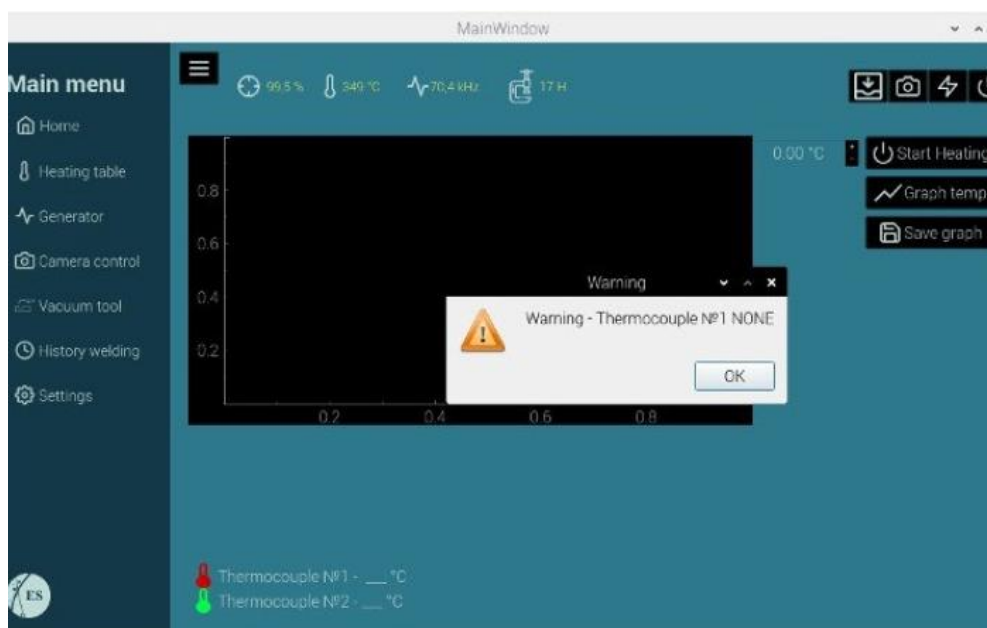


Рисунок 26 - Окно с сообщением о неполадках с датчиком температуры

3.3.4 Дополнительные вкладки программы

Вкладка дает возможность пользователю управлять системой позиционирования, которая включает в себя шаговые двигатели NEMO-17. На рисунке 27 изображена данная вкладка, в ней так же находится окно, которое позволяет получать изображение с цифрового микроскопа и кнопки, которые помогают управлять направлениями двигателей.

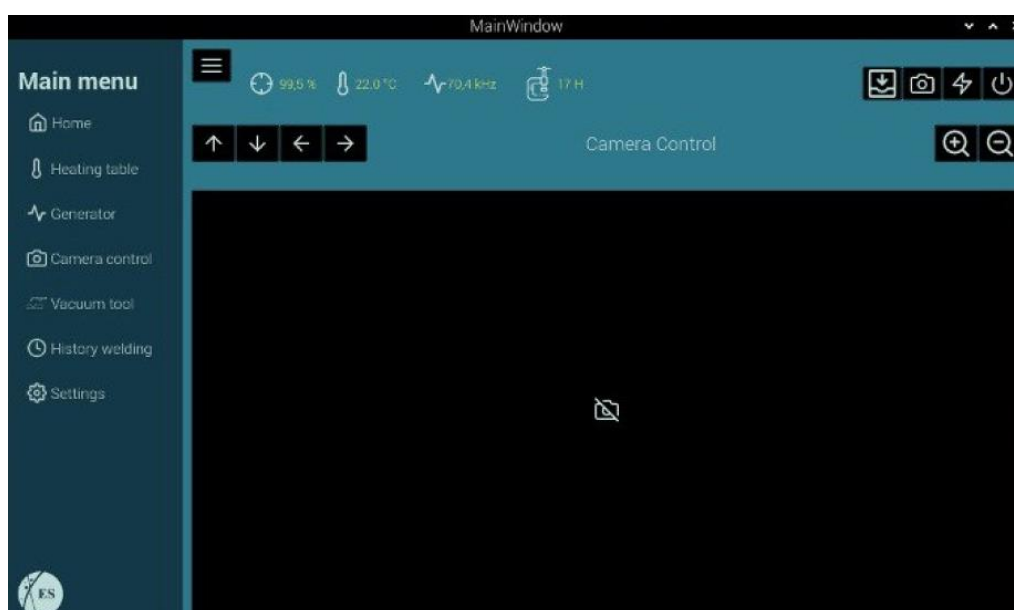


Рисунок 27 – Вкладка управления камеры

Для управления движением двигателей ответственны определенные кнопки, расположенные на вкладке, которые после нажатия вызывают определённые функции (таблица 2). Вызванная функция отправляет команду, в которой указано направление движения двигателя. Подробно описание взаимодействие Arduino и Raspberry Pi описано в главе 3.1.2 и 3.2.

Таблица № 2- Функции и кнопки управления двигателя

Направления движения двигателя	Название кнопки	Название функции	Команда отправки на Arduino
Движение вверх	self.buttonUp	def moveUp	self.SLAVE_ADDRESS, ord('u')
Движение вниз	self.buttonDown	def moveDown	self.SLAVE_ADDRESS, ord('d')
Движение влево	self.buttonLeft	def moveLeft	self.SLAVE_ADDRESS, ord('l')
Движение вправо	self.buttonRight	def moveRight	self.SLAVE_ADDRESS, ord('r')

Программное обеспечение графического интерфейса пользователя позволяет ввести историю всех процессов работы в формате журнала, который располагается на вкладке история процесса. Это позволит отслеживать работу установки и контролировать процесс работы и на основе сохранённых данных улучшать работу установки. Журнал выполнен в виде таблицы, хранящие в себе такую информацию:

1. Имя оператора;
2. Время работы (дата);
3. Материал полупроводников;
4. Общие количество сваренных плат;
5. Количество исправных плат (платы, прошедшие проверку на дефекты);
6. Количество не исправных плат (платы, не прошедшие проверку на качество).

Запись в журнал будет происходить автоматически, пользователь может только просмотреть его и сохранить его в формате CSV или Excel для оформления отчета о работе или дальнейших исследований. На рисунке 28 представлен примерный вариант журнала.

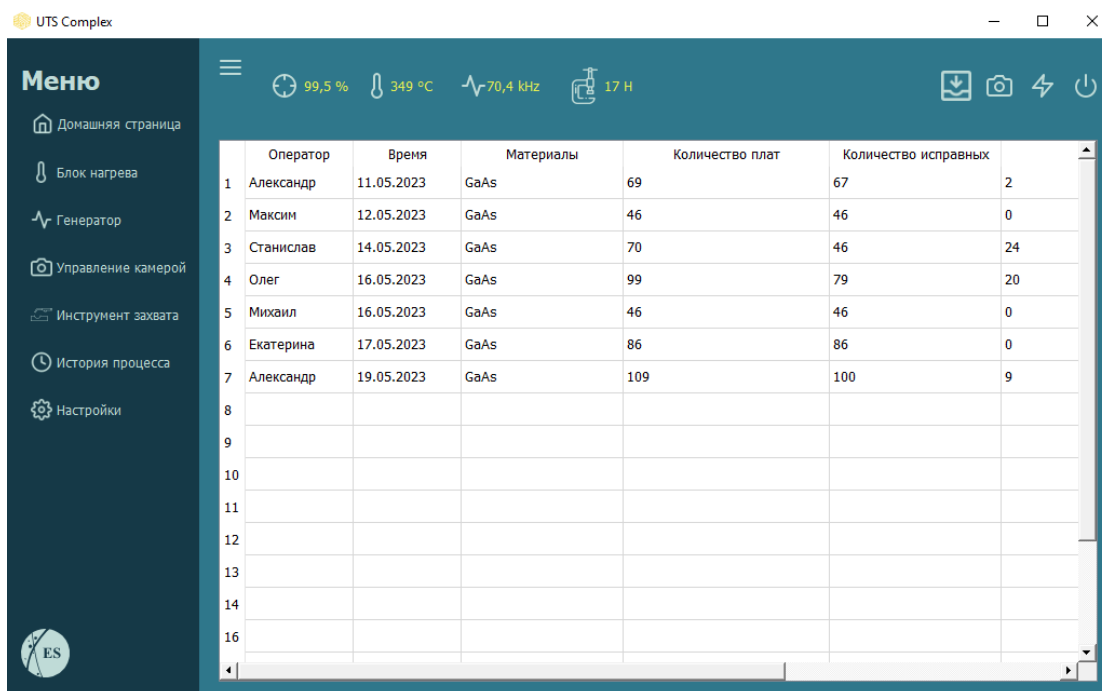


Рисунок 28 – Вкладка история процесса

Вкладка настройки отвечает за настройку программного обеспечения и установки. На данный момент на данной вкладке, на данный момент есть возможность изменить язык графического интерфейса с английского на русский и наоборот. Данная возможность дает людям со знанием русского или английского языка пользоваться установкой и ПО. Подробное описание работы перевода графического интерфейса на другой язык описано в главе 3.3.5. Рисунок 29 демонстрирует вкладку настройки, а рисунки 30 и 31 показывает результат работы перевода интерфейса на русский язык.

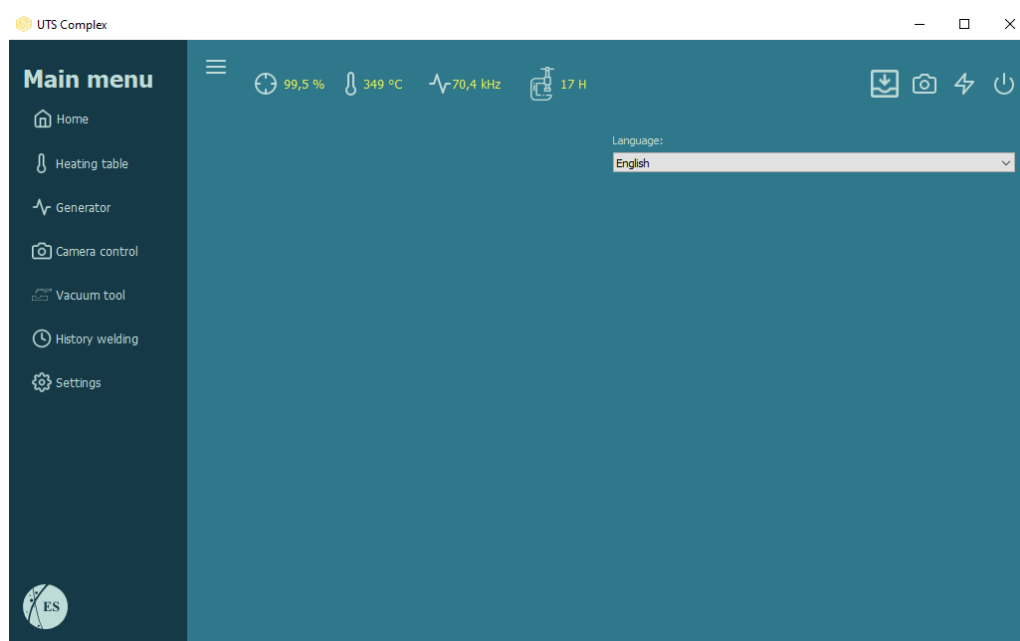


Рисунок 29 – Вкладка Настройки

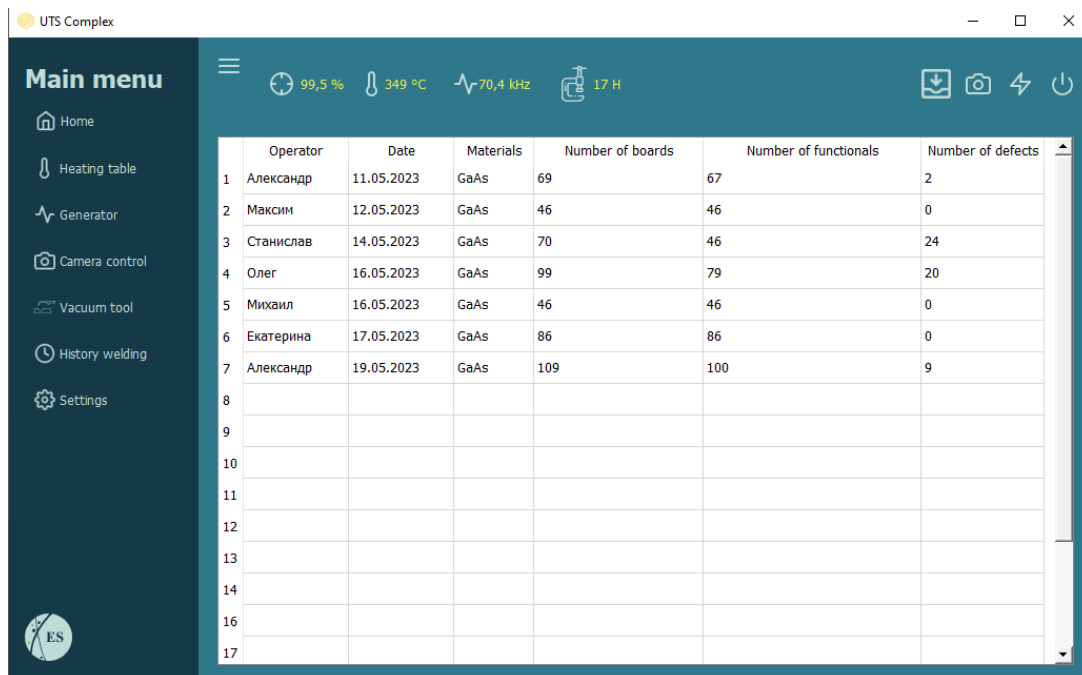


Рисунок 30 – Вкладка история процесса, переведенная на английский язык

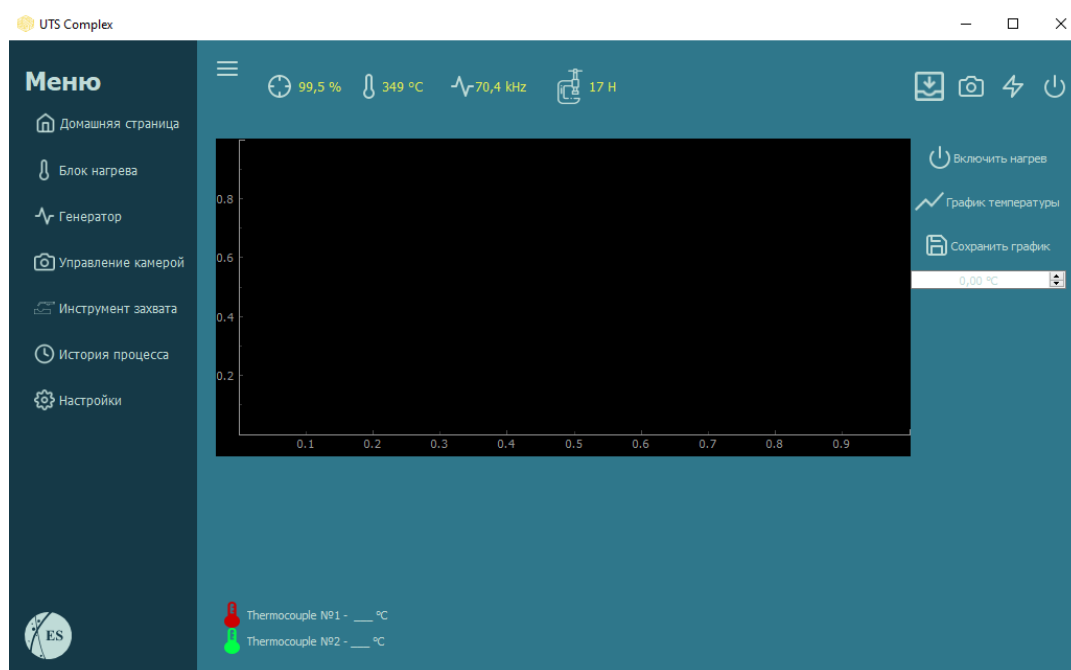


Рисунок 31 – Вкладка Heating table переведенная на русский язык

3.3.5 Локализация интерфейса: многоязычная поддержка в программе

Создание графического интерфейса с поддержкой множества языков является важным этапом разработки программного обеспечения. Локализация программы делает её более удобной в использовании, более доступной и удобной для пользователей, чьи родные языки отличаются от языка той страны, где была разработана программа. Поддержка

разных языков дает возможность пользователю легче освоить программу, быстрее найти нужные функции и уменьшить возможные ошибки из-за непонимания.

Многоязычность интерфейса помогает привлечь пользователей из разных стран. Это особенно важно для компаний, которые стремятся проникнуть на новые рынки и расширить свою географическую аудиторию. В некоторых странах и регионах существуют законодательные требования, которые требуют предоставления программного обеспечения на местном языке. Создание локализации приложения позволяет соответствовать этим требованиям и избежать правовых проблем.

В целом, создание многоязычного интерфейса является важным шагом в разработке программного обеспечения, который позволяет программе достичь более широкой аудитории, повысить удобство использования и улучшить пользовательский опыт. Это способствует росту популярности и успешности программы на глобальном уровне.

В создании многоязычности программы помогают инструменты Qt. Qt Linguist - это инструмент, предоставляемый фреймворком Qt, который позволяет разработчикам локализовать интерфейс своих программ на различные языки. На рисунке 32 представлен окно приложения.

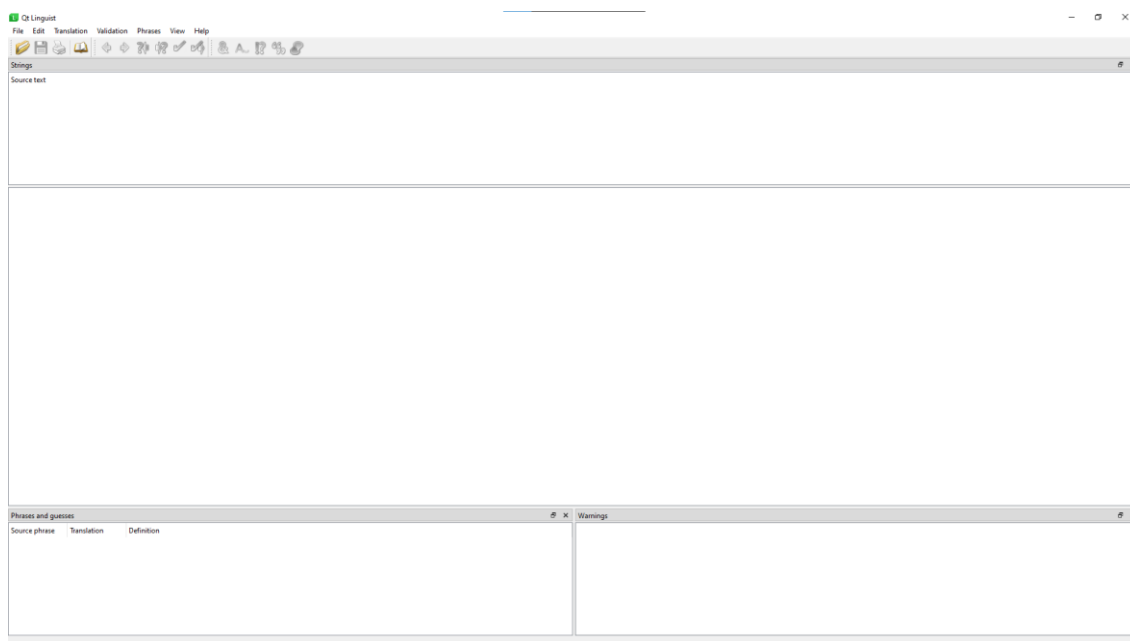


Рисунок 32 – Окно приложения Linguist

Для начала работы мы создаём файлы переводов для каждого языка, на который хотим локализовать интерфейс программы. Эти файлы имеют расширение ".ts". Для создания файла с переводом на каждый язык, используем меню Linguist "File -> New Translation". Для нашей задачи данная функция не подходит, так как мы используем конвертацию программного кода в файлы переводов с помощью утилиты pylupdate5. В

консоли на компьютере ввели команду “pylupdate5 complex_ui.py –ts app_en.ts”. После преобразования файла, переводимый текст собран в специальных файлах, после переводим на целевой язык. В утилите Qt Linguist открываем наши ts-файлы. При первой загрузке выбрали исходный язык и язык перевода. У утилиты есть одна полезная функция – это возможность открыть сразу несколько словарей, чтобы сразу выполнить перевод на несколько языков. На рисунке 33 представлен процесс перевода с английского на русский.

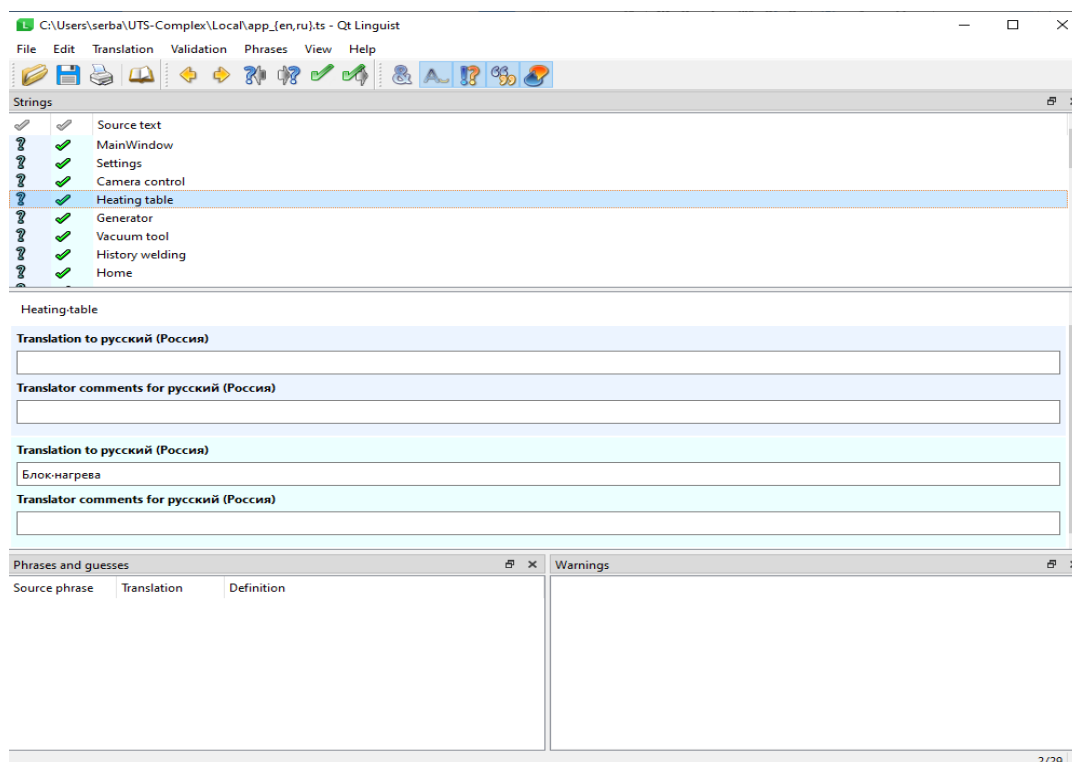


Рисунок 33 – Создания перевода в Qt Linguist

После создания файлов с переводом, созданные словари мы инициализируем их в программном коде. В начале в классе MainWindow, в методе `__init__()` написали данную строчку кода `self.trans = QTranslator()`. Эта строчка создает новый объект `QTranslator`, который используем для перевода текстовых строк на другие языки в нашей программе. После создания объекта, его используем для загрузки файлов перевода и настройки текущего языка соответствующим образом.

Для изменения языка создали выпадающий список с помощью виджета `QComboBox`. Обработка событий выбранного элемента из списка происходит за счет изменения индекса. В нашем случае у каждого языка имеется свой индекс, у русского языка он равняется 1, а у английского 0. Обработка выбора языка происходит в функции `def changeLanguage`, которая располагается в файле `localization.py`. За перевод на английский язык ответственна функция `def trigger_english`. Функция сначала загружает файл с нашим переводом `app_en`. Затем она получает экземпляр приложения Qt и устанавливает загруженный перевод с помощью

функции `installTranslator`. После чего эта функция переводит интерфейс нашей программы на английский язык, используя метод `retranslateUi()`. Аналогичные действия происходят и в функции `def trigger_russian` только имеется одно отличие. Оно заключается в загрузке файла с переводом, функция загружает файл с именем `app_ru`.

4 Экспериментальная часть

4.1 Эксперимент с цифровым микроскопом

Для получения изображения используется цифровой микроскоп, у которого интерфейс подключения происходит по проводу USB. Для трансляции изображения в реальном времени написана программа, которая располагается в файле под названием camera_control.py. Программа написана на языке программирования Python, для работы с изображением использовалась библиотека OpenCV. Для начала работы с камерой был создан класс Microscore, который содержит следующие функции:

1. def start_camera() – происходит открытие камеры и запуск захвата изображения;
2. def _queryFrame() – ответственна за чтение текущего кадра с микроскопа;
3. def take_snapshot() – захватывает кадр изображения и сохраняет его в папке.

В функции start_camera() имеется переменная is_camera1_opened, она хранит в себе состояние камеры. Если значение переменной равен False значит камера не открыта, а если она равна True, значит камера уже открыта. После проверки состояния камеры запускается таймер _timer, который вызывает функцию _queryFrame() каждые секунду. Это позволяет получать изображение с камеры в реальном времени (рисунок 34). Если камера закрыта, то таймер останавливается и прекращается получения изображения с микроскопа.

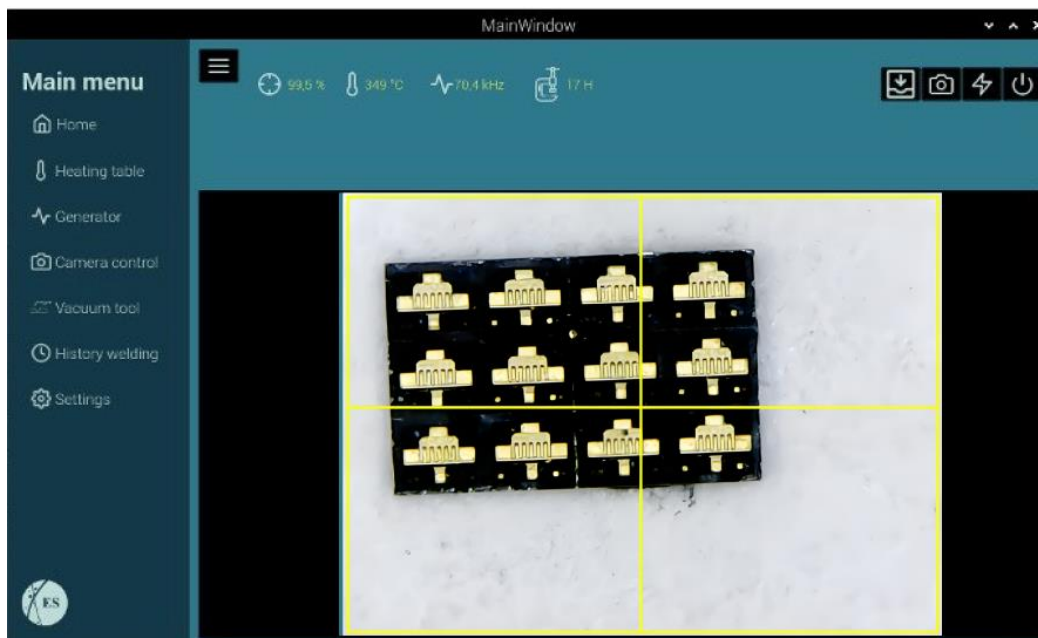


Рисунок 34 – Результат захвата изображения с микроскопа

Таким образом, функция "start_camera()" позволяет контролировать открытие и закрытие камеры. При открытии камеры запускается таймер, который периодически

вызывает функцию `_queryFrame()` для получения изображения с камеры в реальном времени. При закрытии камеры таймер останавливается, и получение изображений прекращается.

4.2 Эксперимент с обработкой изображения и поиск объекта

Как видно на рисунке 35, изображение платы, полученное с микроскопа, не имеет четкой картинки и границ. Для решения данной проблемы используем встроенные функции для обработки изображения входящие в библиотеку OpenCV в целях повышения качества изображения с камеры.

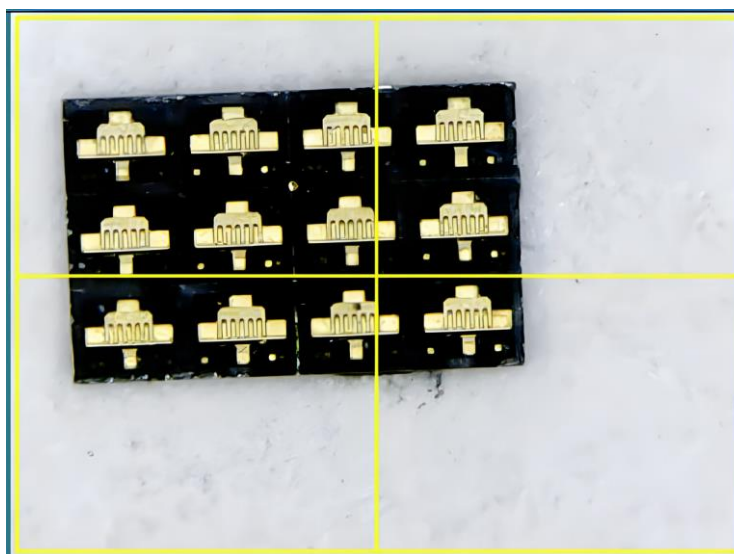


Рисунок 35 – Изображение платы без обработки

Сначала исходно изображение мы переводим из цветного формата в оттенки серого с помощью функции `cv.cvtColor()`. Это позволяет нам упростить обработку изображения и сделать больше акцента на контурах и деталях изображения. Далее добавляем медианное размытие используя функцию `cv.medianBlur()`. Данная функция сглаживает шумы и мелкие детали на кадре с микроскопа. Затем мы используем фильтр увеличения резкости с помощью функции `cv.filter2D()`. Этот фильтр усиливает контраст и границы объектов на изображении, на выходе давая нам более чёткую картинку. После применения фильтра мы добавляем бинаризацию используя функцию `cv.threshold()`. Она переводит изображение в бинарное с использованием порогового значения. Данный алгоритм позволяет убрать фоновый шум. На рисунке 36 демонстрируется результат обработки изображения после применения фильтров.

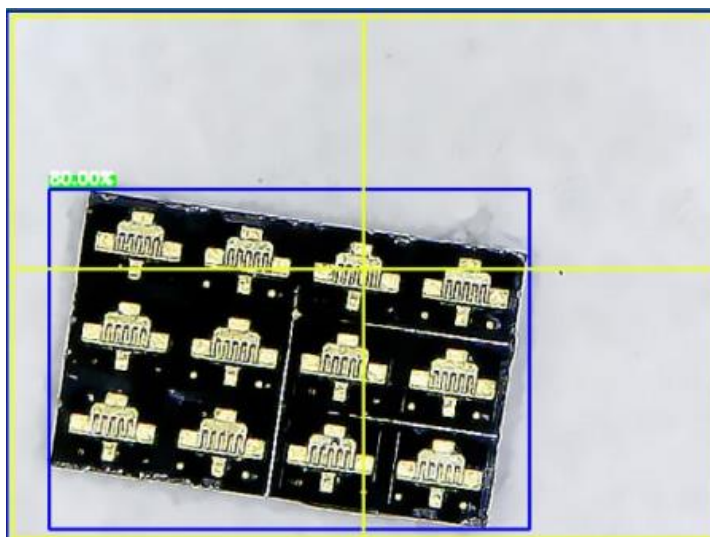


Рисунок 36 – Изображение после применения фильтров

После применения фильтров осталась одна проблема, которая заключается в возникновении бликов на плате. Для избежания бликов был добавлен дополнительный шум, который повышал резкость изображения и при этом не портил качество изображения, это отображено на рисунке 37.

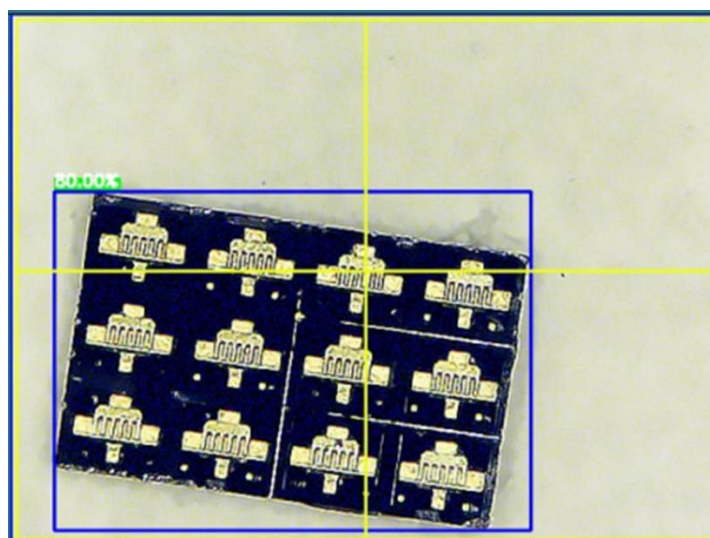


Рисунок 37 - Изображение с добавлением шума

Так же на рисунке 37 демонстрируется результат нахождения объекта. Для этого применяем метод адаптивного порога для выделения контуров объекта. Данный метод используется для преобразования изображения в черно-белое, при котором каждый пиксель на выходе будет иметь значение 0 или 255 на основании порогового значения, вычисленного по пикселям в некоторой окрестности текущего пикселя. Это позволяет учесть изменчивость освещения на изображении и делает метод более гибким и точным, чем простой пороговый метод. В библиотеки OpenCV метод реализован с помощью функции `adaptiveThreshold()`, которую мы используем. Далее применяем

ограничивающий прямоугольник с помощью метода `boundingRect`. Если размеры ограничивающего прямоугольника больше 100x100 пикселей, то на изображении рисуется прямоугольник с границей синего цвета, но заметим, что границы обрисовываются не четко вокруг платы, а только вокруг области. Для этого была добавлена функция `cv.minAreaRect`, она позволяет рисовать границы вокруг объекта независимо от его положения и угла поворота (рисунок38).

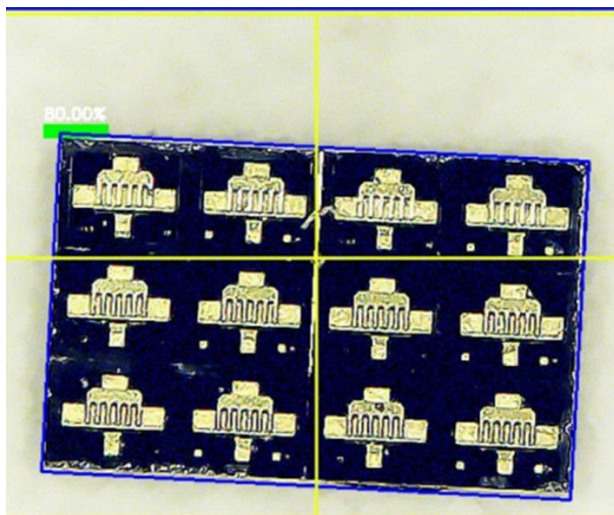


Рисунок 38 – Изображение с четкой обрисовкой контура платы

ЗАКЛЮЧЕНИЕ

В работе была предложена структура схемы разрабатываемого макета установки для ультразвуковой сварки по технологии Flip-Chip. Схема включает в себя в стандартные блоки для данных установок, главное преимущество разрабатываемого макета – это возможность управления всеми блоками через графический интерфейс пользователя и будущие автоматизация процессов сварки полупроводниковых структур на различные подложки.

Реализована плата систему управления включающая в себя микроконтроллер Arduino, цифровой преобразователь для термопар и драйверы для управления шагом движения двигателей. Плата имеет разъем для подключения платы по интерфейсу I2C с главным компьютером, через который и происходит управления всеми система комплекса.

Написано ПО для управления системой позиционирования, сбора данных с датчиков и взаимодействия микроконтроллера и главного компьютера. Пользователю дается возможность управлять системой позиционирования через матричную клавиатуру или графический интерфейс пользователя.

Графический интерфейс спроектирован на базе PyQt, он служит для управления всеми системами установки, в том числе позволяющий получать изображение с цифрового микроскопа с возможностью распознавания платы. Программа состоит из множества вкладок, которые отвечают за определённый модуль установки. Кроме того, программа поддерживает два языка – это русский и английский.

Во вкладке блока нагрева пользователь может отслеживать температуру датчиков, отображаемых в окне приложения и на основе полученных с них значений может наблюдать изменения температуры на графике, при желании который он может сохранить в формате png. В программу добавлено предупреждения пользователя о выходе из строя датчиков температуры.

На главной странице и вкладке управления камерой, можно наблюдать за изображением в реальном времени получаемый с цифрового микроскопа и управления системой позиционирования. На вкладке история процесса пользователь в будущем может отслеживать работу установки вести отсчет или проводить дальнейшие исследования при монтаже полупроводников, а на вкладке настройки может изменять язык программы с у

Проведены эксперименты с захватом изображения с цифрового микроскопа, полученные изображения не удовлетворяли качеством для наших задач и после были применены функции для увеличения резкости изображения, для медианного размытия и для бинаризации кадра, что поспособствовало улучшению визуального восприятия.

После повышения качества картинки, было реализован алгоритм поиска объекта на изображении с использованием метода адаптивного порога для выделения контуров, на которых применяется ограничивающий прямоугольник. После нахождения границ обрисовывает вокруг него.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Davis E., Harding W., Schwartz R., Corning J. Solid Logic Technology: Versatile, High Performance Microelectronics // IBM Journal of Research and Development. 1964. Vol. 8. Is. 2. P. 102–114.
2. Борисова Ю. Технологии флип-чип монтажа кристаллов CMOS на органические подложки // Технологии в электронной промышленности. 2017. № 4.
3. Arshad M.K., Hashim U., Muzamir I. Under Bump Metallurgy (UBM) Technology Review for Flip Chip Packaging // International Journal of Mechanical and Materials Engineering. 2007. Vol. 2. No. 1.
4. Tian Y., Wang C. Characteristics of Laser Reflow Bumping of Sn3.5Ag and Sn3.5Ag0.5Cu Lead-Free Solder Balls // J. Mater. Sci. Technol, 2008. No. 2.
5. Totta P., Sopher R. SLT Device Metallurgy and Its Monolithic Extension // IBM Journal of Research and Development. 1969. Vol. 13. Is. 3. P. 226–238.
6. Comparing Flip-Chip and Wire-Bond Interconnection Technologies [Электронный ресурс] – URL: processsolutionsconsulting.com (дата обращения 15.05.2022)
7. Ли Н.-Ч. Технология пайки оплавлением, поиск и устранение дефектов: поверхностный монтаж, BGA, CSP и Flip-Chip технологии. М.: ИД «Технологии». 2006.
8. Алина Томасян Какой тип исследования использовать для проведения численного электротермического анализа? – URL: <https://www.comsol.ru/blogs/which-study-type-should-i-use-for-my-electrothermal-analysis/> (дата обращения: 04.01.2023)- Режим доступа: Открытый.
9. ГОСТ 5632-2014 Межгосударственный стандарт нержавеющие стали и сплавы коррозионно-стойкие, жаростойкие и жаропрочные: дата введения 2015-01-01 – URL: <https://docs.cntd.ru/document/1200113778> - Режим доступа: Открытый.
10. 12. Владимир Ланин Технологические особенности монтажа Flip-Chip – URL: <https://power-e.ru/tech/tehnologicheskie-osobennosti-montazha-flip-chip/> (дата обращения: 22.09.2022) - Режим доступа: Открытый
11. Summerfield, Mark. Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming. Prentice Hall, 2015. 648 p.
12. Kaehler, Adrian, and Gary Bradski. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. O'Reilly Media, Inc., 2016. 1024 p.
13. Официальная документация Pyqtgraph. URL: <https://www.pyqtgraph.org/documentation/> Дата обращения: 29.03.2023.

ПРИЛОЖЕНИЕ А
QR КОД НА РЕПОЗИТОРИЙ ПРОГРАММ



ПРИЛОЖЕНИЕ Б **ЧЕРТЕЖИ НАГРЕВАТЕЛЬНОГО СТОЛА**

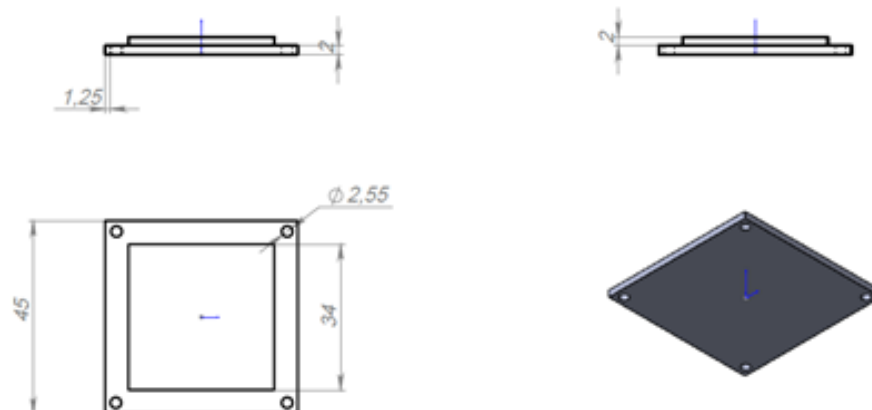
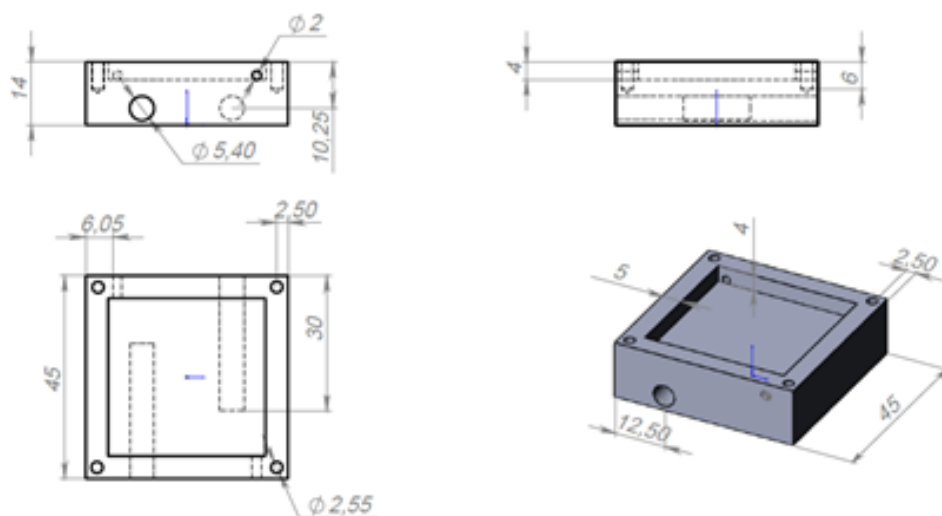
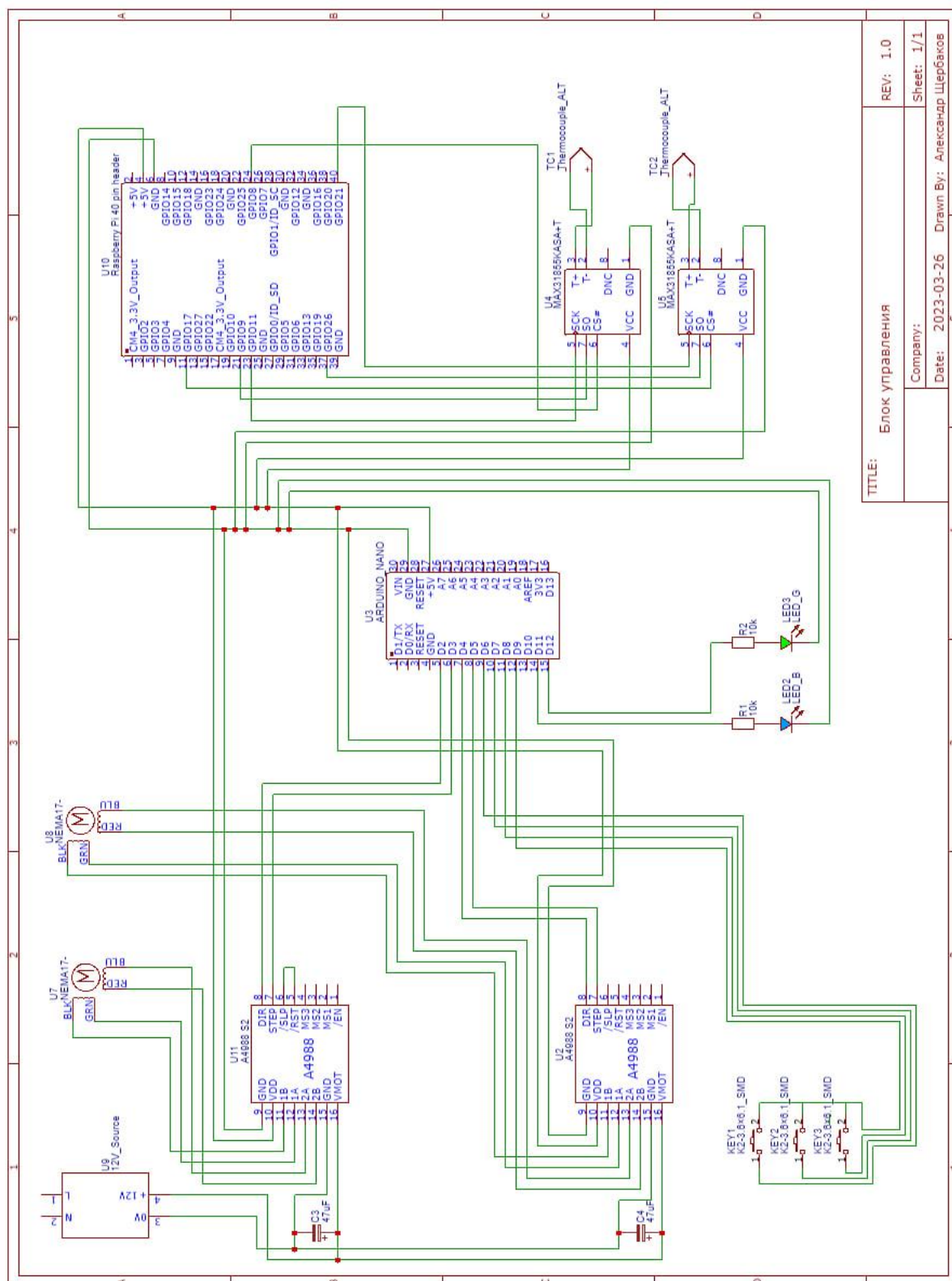


СХЕМА ПОДКЛЮЧЕНИЯ СИСТЕМЫ УПРАВЛЕНИЯ





АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

СПРАВКА

Томский Государственный Университет

о результатах проверки текстового документа
на наличие заимствований

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Щербаков Александр
Самоцитирование
рассчитано для: Щербаков Александр
Название работы: ВКР Щербаков А.
Тип работы: Выпускная квалификационная работа
Подразделение:

РЕЗУЛЬТАТЫ

■ ОТЧЕТ О ПРОВЕРКЕ КОРРЕКТИРОВАЛСЯ: НИЖЕ ПРЕДСТАВЛЕНЫ РЕЗУЛЬТАТЫ ПРОВЕРКИ ДО КОРРЕКТИРОВКИ

СОВПАДЕНИЯ	22.37%	СОВПАДЕНИЯ	22.37%
ОРИГИНАЛЬНОСТЬ	77.63%	ОРИГИНАЛЬНОСТЬ	77.63%
ЦИТИРОВАНИЯ	0%	ЦИТИРОВАНИЯ	0%
САМОЦИТИРОВАНИЯ	0%	САМОЦИТИРОВАНИЯ	0%

ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 09.06.2023

ДАТА И ВРЕМЯ КОРРЕКТИРОВКИ: 09.06.2023 06:41

Структура документа: Проверенные разделы: основная часть с.2-4, 6-52, приложение с.52-54
Модули поиска: ИПС Адилет; Библиография; Сводная коллекция ЭБС; Интернет Плюс*; Сводная коллекция РГБ; Цитирование; Переводные заимствования (RuEn); Переводные заимствования по eLIBRARY.RU (EnRu); Переводные заимствования по коллекции Гарант: аналитика; Переводные заимствования по коллекции Интернет в английском сегменте; Переводные заимствования по Интернету (EnRu); Переводные заимствования по коллекции Интернет в русском сегменте; Переводные заимствования издательства Wiley ; eLIBRARY.RU; СПС ГАРАНТ: аналитика; СПС ГАРАНТ: нормативно-правовая документация; Медицина; Диссертации НББ; Коллекция НБУ; Перефразирования по eLIBRARY.RU; Перефразирования по СПС ГАРАНТ: аналитика; Перефразирования по Интернету; Перефразирования по Интернету (EN); Перефразированные заимствования по коллекции Интернет в английском сегменте; Перефразированные заимствования по коллекции Интернет в русском сегменте; Перефразирования по коллекции

Работу проверил: Лапутенко Андрей Владимирович

ФИО проверяющего

Дата подписи: 09-06-2023

Подпись проверяющего



Чтобы убедиться
в подлинности справки, используйте QR-код,
который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование
корректным, система оставляет на усмотрение проверяющего.
Предоставленная информация не подлежит использованию
в коммерческих целях.

Щербаков А.С.