

Санкт-Петербургский политехнический университет Петра Великого  
Институт машиностроения, материала и транспорта  
Федеральное государственное автономное образовательное учреждение  
Высшая школа автоматизации и робототехники

## **Курсовая работа**

**Дисциплина: Объектно-ориентированное программирование**  
**Тема: Разработка ПЛК**

**Выполнил студент гр. 3331506/20401**

**Вчерашний З. Д.**

**Преподаватель**

**Ананьевский М. С.**

Санкт-Петербург  
2025

# Содержание

<b>1</b>	<b>Введение</b>	<b>2</b>
<b>2</b>	<b>Обзор литературы</b>	<b>2</b>
<b>3</b>	<b>Расчет толщины дорожки и дренаж земляного полигона</b>	<b>2</b>
3.1	Толщина дорожки в зависимости от тока . . . . .	2
3.2	Дренаж земляного полигона . . . . .	2
<b>4</b>	<b>Сборка ядра Linux и загрузчика U-Boot</b>	<b>2</b>
4.1	Подготовка окружения . . . . .	2
4.2	Получение исходных кодов . . . . .	3
4.3	Конфигурация ядра Linux . . . . .	3
4.4	Сборка ядра Linux . . . . .	3
4.5	Конфигурация U-Boot . . . . .	3
4.6	Сборка U-Boot . . . . .	3
4.7	Установка ядра и загрузчика . . . . .	4
4.8	Настройка загрузчика . . . . .	4
<b>5</b>	<b>Конфигурация библиотек Libmodbus и libcan для работы с аппаратными портами</b>	<b>4</b>
5.1	Установка библиотек . . . . .	4
5.1.1	Установка Libmodbus . . . . .	4
5.1.2	Установка libcan . . . . .	4
5.2	Конфигурация Libmodbus . . . . .	4
5.2.1	Настройка соединения . . . . .	4
5.2.2	Чтение и запись данных . . . . .	4
5.3	Конфигурация libcan . . . . .	5
5.3.1	Настройка CAN-интерфейса . . . . .	5
5.3.2	Отправка и получение сообщений . . . . .	5
<b>6</b>	<b>Теоретические основы трассировки дифференциальных пар</b>	<b>5</b>
6.1	Основы дифференциальных пар . . . . .	5
6.2	Принципы трассировки . . . . .	5
6.3	Параметры и характеристики . . . . .	6
6.4	Теория . . . . .	6
<b>7</b>	<b>Разработка ПЛК</b>	<b>6</b>
7.1	Разработка аппаратной платформы . . . . .	6
7.2	Реализация промышленных интерфейсов . . . . .	6
<b>8</b>	<b>Результаты</b>	<b>6</b>
<b>9</b>	<b>Заключение</b>	<b>6</b>

# 1 Введение

Современные промышленные системы, объединяющие алгоритмы технического зрения и нейронные сети, часто сталкиваются с проблемой архитектурной фрагментации из-за разделения функциональных модулей. Обычно такие системы состоят из нескольких ключевых компонентов:

1. Специализированный модуль обработки изображений, выполняющий задачи обнаружения и классификации объектов;
2. Контроллер, предназначенный для сбора данных с различных датчиков через интерфейсы, такие как UART, I2C и CAN;
3. Промежуточные узлы, обеспечивающие предварительную обработку сигналов и преобразование интерфейсов.

## 2 Обзор литературы

Интеграция нейросетевых моделей в промышленные системы активно изучается в рамках задач автоматизации, где ведутся разработки нейросетевых регуляторов для существующих программируемых логических контроллеров (ПЛК) [1, 2, 3]. Исследования [4, 5] демонстрируют эффективность моделей YOLO и ResNet для обнаружения объектов на производственных линиях. Другие работы [6, 7] подчеркивают значимость аппаратного ускорения, например, с использованием технологий, таких как TensorRT.

## 3 Расчет толщины дорожки и дренаж земляного полигона

### 3.1 Толщина дорожки в зависимости от тока

Для расчета ширины дорожки на печатной плате используется следующая формула:

$$w = \frac{I}{k \cdot \Delta T^b}$$

где  $w$  — ширина дорожки,  $I$  — ток,  $k$  и  $b$  — константы, зависящие от материала платы, а  $\Delta T$  — допустимое повышение температуры.

### 3.2 Дренаж земляного полигона

Дренаж земляного полигона на печатной плате важен для обеспечения стабильного заземления и минимизации помех. Основные принципы включают:

- Минимизация импеданса заземления.
- Разделение аналогового и цифрового заземления.
- Учет тепловых характеристик.

## 4 Сборка ядра Linux и загрузчика U-Boot

### 4.1 Подготовка окружения

Перед началом сборки необходимо подготовить окружение, установив все необходимые инструменты и зависимости. Обычно это включает компилятор, утилиты для сборки и библиотеки.

- Установка необходимых пакетов:

```
sudo apt-get update
sudo apt-get install build-essential bc libncurses-dev flex bison libssl-dev
```

## 4.2 Получение исходных кодов

Необходимо получить исходные коды ядра Linux и загрузчика U-Boot. Это можно сделать, клонируя официальные репозитории с помощью Git.

- Клонирование ядра Linux:

```
git clone https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
```

- Клонирование U-Boot:

```
git clone https://git.denx.de/u-boot.git
```

## 4.3 Конфигурация ядра Linux

После получения исходных кодов необходимо сконфигурировать ядро Linux в соответствии с требованиями вашей аппаратной платформы.

- Переход в директорию с исходниками ядра:

```
cd linux
```

- Запуск конфигурации:

```
make menuconfig
```

## 4.4 Сборка ядра Linux

После конфигурации можно приступить к сборке ядра.

- Запуск процесса сборки:

```
make -j$(nproc)
```

## 4.5 Конфигурация U-Boot

Аналогично ядру, необходимо сконфигурировать U-Boot для вашей аппаратной платформы.

- Переход в директорию с исходниками U-Boot:

```
cd u-boot
```

- Запуск конфигурации:

```
make menuconfig
```

## 4.6 Сборка U-Boot

После конфигурации можно собрать U-Boot.

- Запуск процесса сборки:

```
make -j$(nproc)
```

## 4.7 Установка ядра и загрузчика

После успешной сборки необходимо установить ядро и загрузчик на целевую систему.

- Копирование ядра и загрузчика на целевую систему:

```
scp arch/arm/boot/zImage user@target:/boot/  
scp u-boot.bin user@target:/boot/
```

## 4.8 Настройка загрузчика

Последний этап включает настройку загрузчика для загрузки нового ядра.

- Настройка U-Boot для загрузки ядра:

```
setenv bootcmd 'fatload mmc 0:1 0x80007fc0 zImage; bootz 0x80007fc0'  
saveenv
```

# 5 Конфигурация библиотек Libmodbus и libcan для работы с аппаратными портами

## 5.1 Установка библиотек

Для работы с аппаратными портами через протоколы Modbus и CAN необходимо установить соответствующие библиотеки. Ниже приведены инструкции по установке и базовой настройке.

### 5.1.1 Установка Libmodbus

Libmodbus — это библиотека, которая позволяет взаимодействовать с устройствами через протокол Modbus. Установка библиотеки может быть выполнена с помощью следующих команд:

```
sudo apt-get update  
sudo apt-get install libmodbus-dev
```

### 5.1.2 Установка libcan

Для работы с CAN-интерфейсами может потребоваться библиотека, такая как SocketCAN. Установите необходимые пакеты с помощью следующих команд:

```
sudo apt-get update  
sudo apt-get install can-utils libsocketcan-dev
```

## 5.2 Конфигурация Libmodbus

### 5.2.1 Настройка соединения

Для настройки соединения с устройством через Modbus необходимо указать параметры соединения, такие как адрес устройства, скорость передачи данных и другие параметры. Пример настройки соединения:

```
#include <modbus.h>  
modbus_t *ctx;  
  
ctx = modbus_new_rtu("/dev/ttyUSB0", 9600, 'N', 8, 1);  
modbus_connect(ctx);
```

### 5.2.2 Чтение и запись данных

Пример чтения и записи данных с использованием Libmodbus:

```
uint16_t tab_reg[64];  
  
modbus_read_registers(ctx, 0, 10, tab_reg);  
modbus_write_register(ctx, 0, value);
```

## 5.3 Конфигурация libcan

### 5.3.1 Настройка CAN-интерфейса

Для работы с CAN-интерфейсом необходимо настроить параметры интерфейса, такие как битрейт. Пример настройки CAN-интерфейса:

```
#include <net/if.h>
#include <sys/ioctl.h>
#include <linux/can.h>
#include <linux/can/raw.h>

int s;
struct sockaddr_can addr;
struct ifreq ifr;

s = socket(PF_CAN, SOCK_RAW, CAN_RAW);
strcpy(ifr.ifr_name, "can0");
ioctl(s, SIOCGIFINDEX, &ifr);
addr.can_family = AF_CAN;
addr.can_ifindex = ifr.ifr_ifindex;
bind(s, (struct sockaddr *)&addr, sizeof(addr));
```

### 5.3.2 Отправка и получение сообщений

Пример отправки и получения сообщений через CAN-интерфейс:

```
struct can_frame frame;

frame.can_id = 0x123;
frame.can_dlc = 2;
frame.data[0] = 0x11;
frame.data[1] = 0x22;

write(s, &frame, sizeof(struct can_frame));

read(s, &frame, sizeof(struct can_frame));
```

## 6 Теоретические основы трассировки дифференциальных пар

### 6.1 Основы дифференциальных пар

Дифференциальные пары представляют собой пару проводников, по которым передаются сигналы с противоположными фазами. Основное преимущество использования дифференциальных пар заключается в их устойчивости к внешним помехам и способности передавать данные на высоких скоростях с минимальными искажениями.

### 6.2 Принципы трассировки

При трассировке дифференциальных пар на печатных платах необходимо соблюдать ряд правил для обеспечения оптимальной передачи сигналов:

- **Симметрия:** Обеспечение симметрии в длине и геометрии проводников для минимизации задержек и искажений.
- **Расстояние между проводниками:** Поддержание постоянного расстояния между проводниками пары для контроля импеданса.
- **Избегание пересечений:** Минимизация пересечений с другими сигнальными линиями для снижения перекрестных помех.

## 6.3 Параметры и характеристики

Контроль импеданса является критически важным аспектом при проектировании дифференциальных пар. Импеданс должен быть согласован с импедансом источника и приемника для минимизации отражений сигнала. Длина трасс должна быть одинаковой для обоих проводников пары, чтобы избежать временных задержек и фазовых искажений.

## 6.4 Теория

Импеданс дифференциальной пары можно рассчитать с использованием следующей формулы:

$$Z_{diff} = \frac{V_{diff}}{I_{diff}}$$

где  $V_{diff}$  — это разность напряжений между двумя проводниками, а  $I_{diff}$  — ток, протекающий через пару.

Задержка распространения сигнала в дифференциальной паре может быть выражена как:

$$t_{pd} = \frac{l}{v}$$

где  $l$  — длина трассы, а  $v$  — скорость распространения сигнала в материале печатной платы.

Потери в дифференциальных парах могут быть описаны уравнением:

$$\alpha = \alpha_c \sqrt{f} + \alpha_d f$$

где  $\alpha_c$  — коэффициент потерь, связанный с проводимостью,  $\alpha_d$  — коэффициент диэлектрических потерь, а  $f$  — частота сигнала.

# 7 Разработка ПЛК

## 7.1 Разработка аппаратной платформы

Первостепенной задачей проектирования стало формирование технического задания, учитывающего требования к производительности, надежности и совместимости с промышленными стандартами.

## 7.2 Реализация промышленных интерфейсов

Среди промышленных интерфейсов реализованы:

- CAN: 2 изолированных канала;
- RS485: 2 канала с поддержкой Modbus и скоростью 8 Мбит/с;
- Ethernet: 4 порта со скоростью 1000 Мбит/с.

# 8 Результаты

В результате разработаны печатные платы для интерфейсов (рис. 2а) и управляющего модуля ПЛК (рис. 3б).

# 9 Заключение

Разработанная аппаратная платформа демонстрирует эффективность интеграции нейросетевых моделей в промышленные системы. Исключение промежуточных узлов позволило уменьшить задержки, повысить надежность и облегчить процесс масштабирования.

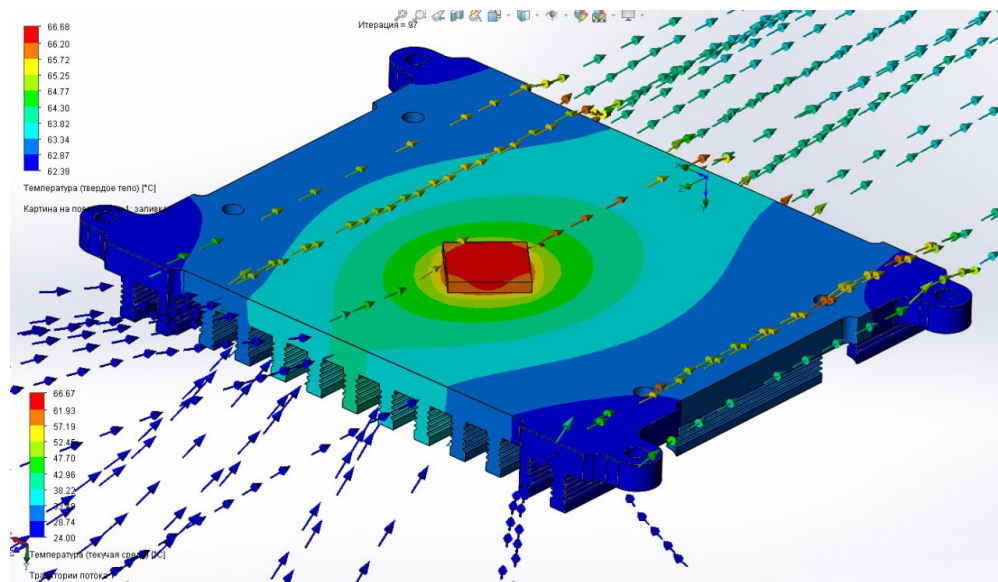
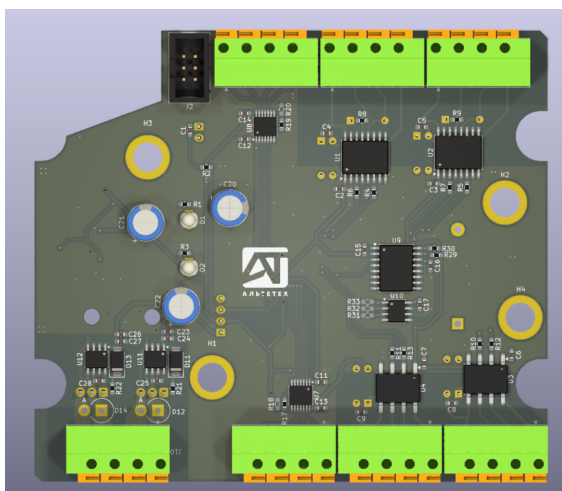
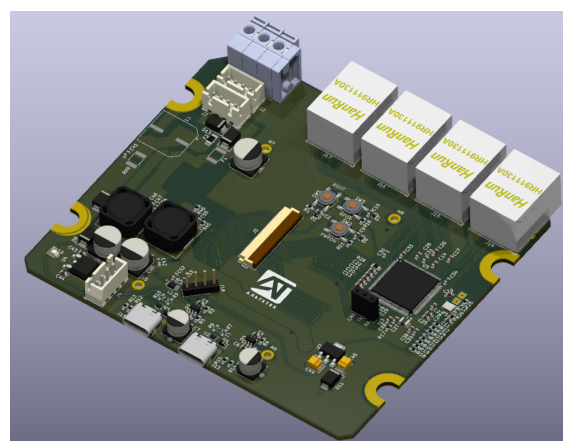


Рис. 1: Расчет теплораспределения при 15 Вт потребления



(а) Плата-расширения интерфейсов

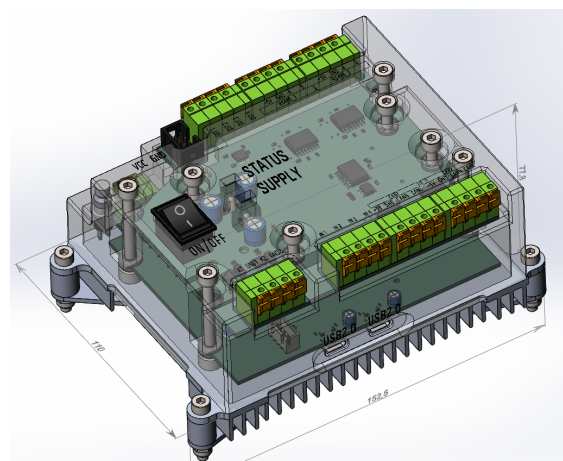


(б) Плата управляющего модуля

Рис. 2: Разработанные печатные платы



(а) Изделие в сборе



(б) 3D-модель

Рис. 3



## Список литературы

- [1] Толстель О. В., Ширкин А. Е., Калабин А. Л. Построение системы технического зрения для выравнивания содержимого упаковок дельта-манипулятором на пищевом производстве // Программные продукты и системы. – 2023. – Т. 36. – №. 2. – С. 334-341.
- [2] Volkov A. K., Mironova L. V., Potapova S. E. The use of pretrained neural networks for solving the problem of reverse searching of X-ray images of prohibited items and substances // МГТУ ГА. – С. 9.
- [3] Мокрецов Н. С., Татарникова Т. М. Оптимизация процесса обучения при ограниченном объеме вычислительных ресурсов.
- [4] Степаненко С. О., Якимов П. Ю. Использование высокопроизводительной платформы глубокого обучения для ускорения обнаружения объектов // СБОРНИК ТРУДОВ ИТНТ-2019. – 2019. – С. 624-630.
- [5] Kortli Y. et al. Deep embedded hybrid CNN–LSTM network for lane detection on NVIDIA Jetson Xavier NX // Knowledge-based systems. – 2022. – Т. 240. – С. 107941.
- [6] Ismagilov R. Performance Evaluation of the Rockchip Systems-on-Chip Through YOLOv4 Object Detection Model // 2023 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). – IEEE, 2023. – С. 241-243.
- [7] Савиных А. А., Астахов А. М. Прочностной и термодинамический расчет теплообменника // ББК 1 Н 34. – С. 5247.
- [8] Тестоедов Н. А. и др. Контроль зазоров в конструкциях технических изделий в процессе вибрационных испытаний // Обработка металлов: технология, оборудование, инструменты. – 2021. – Т. 23. – №. 2. – С. 40-53.
- [9] Ibrahim M. Q. et al. Optimizing Convolutional Neural Networks: A Comprehensive Review of Hyperparameter Tuning Through Metaheuristic Algorithms // Archives of Computational Methods in Engineering. – 2025. – С. 1-38.