

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ГК МФМК»
(ООО «ГК МФМК»)

УТВЕРЖДАЮ
Генеральный директор
ООО «ГК МФМК»

_____ Лудиков А.В.
« ____ » _____ 2026 г.

ПРОШИВКА МОДУЛЯ ДИСКРЕТНОГО ВВОДА
«TITAN_N1DIXKK_FACTORY»
Описание архитектуры

РАЗРАБОТАНО
ООО «ГК МФМК»

Дата введения: _____
Без ограничения срока действия.

г. Москва
2026

Инв. № подл.	Подпись и дата
Взам. инв. №	Инв. № дубл.
Подпись и дата	Подпись и дата

Содержание

1 Общие положения	4
2 Структура	5
3 Компоненты	6
4 Взаимодействие	8
5 Технологии	10

Введение

Настоящий документ содержит описание технической архитектуры прошивки модуля дискретного ввода «Titan_N1DIXKK_factory».

Прошивка «Titan_N1DIXKK_factory» представляет собой встраиваемую операционную систему реального времени для модулей дискретного ввода, разработанную на основе ядра FreeRTOS и стека CANopen.

В данном документе приведено описание структуры, состава компонентов, взаимодействия между элементами системы и технологического стека, используемого в составе прошивки «Titan_N1DIXKK_factory»

1 Общие положения

1.1. Настоящий документ содержит описание архитектуры прошивки «Titan_N1DIXKK_factory» (далее – Программа).

1.2. Исключительные права на Программу принадлежат ООО «ГК МФМК», г. Москва (далее – Компания).

1.3. Настоящий документ подлежит размещению на сайте Компании в сети Интернет по адресу: <https://1.titanplc.ru/> (далее – официальный сайт).

2 Структура

2.1. Прошивка модуля дискретного ввода «Titan_N1DIXKK_factory» представляет собой встраиваемую операционную систему реального времени, построенную на основе архитектуры, включающей несколько ключевых слоёв. Система разработана на базе ядра FreeRTOS и стека CANopen и предназначена для функционирования на модулях дискретного ввода ПТК «ТИТАН КОНТРОЛ 1000».

2.2. Архитектура системы включает следующие основные уровни:

- Ядро FreeRTOS обеспечивает планирование и управление задачами в режиме реального времени, управление очередями сообщений, семафорами, таймерами и потоками данных;
- Стек CANopen реализует обмен данными по протоколу CANopen, включая профили CiA 301, LSS, Safety, PDO, SDO, NMT, Heartbeat и другие;
- Драйверы периферийных устройств обеспечивают работу с АЦП, ЦАП, SPI, UART, GPIO, флэш-памятью, EEPROM и другими интерфейсами;
- Обработка аналоговых сигналов включает фильтрацию входных сигналов (медианный фильтр, фильтр Калмана, скользящее среднее), масштабирование данных и программную настройку диапазонов;
- Управление светодиодной индикацией обеспечивает индикацию состояния модуля и входных/выходных сигналов;
- Сбор и передача диагностической информации включает самодиагностику, трассировку и логирование событий системы.

3 Компоненты

3.1. Основные компоненты Программы перечислены в таблице 1.

Таблица 1 – Основные компоненты прошивки «Titan_N1DIXKK_factory»

Компонент	Назначение
Ядро ОС (FreeRTOS)	Планировщик, управление задачами, очередями, семафорами, таймерами и потоками данных
API-заголовки	Интерфейсы для прикладного кода (API-функции ОС)
Управление памятью	Реализация аллокатора памяти (heap_4.c)
CANopen	Базовые файлы стека CANopen; профиль CiA 301 (Emergency, Heartbeat, NMT, Node Guarding, PDO, SDO, SYNC, TIME); LSS-протокол (мастер и слейв); Safety-профиль (GFC, SRDO); ASCII-шлюз для доступа к объектному словарю; трассировка и диагностика; хранение параметров в EEPROM и флэш-памяти; реализация модулей аналогового/дискретного ввода-вывода; драйверы для STM32 (CAN, EEPROM, storage); объектный словарь (OD.c, OD.h)
HDD (XDD-файл)	Описание функций и параметров CANopen-модуля (индивидуальный XDD-файл для каждого типа модуля)
CMSIS	Заголовочные файлы CMSIS для ARM-ядер
STM32F0	Заголовочные файлы CMSIS для STM32F0, скрипты линкера, SVD-файл, системная инициализация
Драйверы	Драйверы АЦП (SGM5860X, HX711); драйверы ЦАП (CBM128S085, DAC8775); драйверы датчиков (MAX31865 –

Продолжение таблицы 1

	термопара, MS5614 – давление, SD2421); драйверы расширителей (MCP23S08); HAL- обёртки для SPI, GPIO, UART; интерфейс работы с флэш-памятью; эмуляция EEPROM во флэш-памяти
Фильтры	Алгоритм медианной фильтрации, Фильтр Калмана
Скользящее среднее	Алгоритм скользящего среднего
Масштабирование	Масштабирование данных
Антидребезг	Антидребезг кнопок
Циклический буфер	Реализация циклического буфера
CRC16	Подсчёт контрольной суммы CRC16
TraceRecorder	Запись трассировки: конфигурационные, заголовочные файлы и реализация модулей
Самодиагностика	Самодиагностика модуля
Управление светодиодами	Управление светодиодной индикацией состояния модуля и сигналов
Типы данных	Определение типов данных (boolean.h, float.h)
Сенсоры	Обёртка для работы с сенсорами

4 Взаимодействие

4.1. Загрузка системы

Программа загружается во внутреннюю флэш-память модуля дискретного ввода через интерфейс загрузки SWD/JTAG в виде исполняемого файла Titan_N1DIXKK_factory.elf (формат ELF). После подачи питания на модуль ядро FreeRTOS выполняет инициализацию аппаратного обеспечения, настройку периферийных устройств и запуск системных задач реального времени.

4.2. Взаимодействие с аппаратным обеспечением

Взаимодействие с аппаратными компонентами модуля осуществляется через драйверы устройств, реализованные в составе Программы:

- Аналоговые входы – чтение данных с АЦП (SGM5860X, HX711) для преобразования аналоговых сигналов в цифровой код;
- Цифровые интерфейсы – работа с SPI, UART, GPIO через HAL-обёртки;
- Флэш-память и EEPROM – хранение параметров конфигурации и объектного словаря через интерфейс флэш-памяти и эмуляцию EEPROM;
- Светодиоды – управление светодиодной индикацией состояния модуля и сигналов.

4.3. Взаимодействие по протоколу CANopen

Взаимодействие с внешними устройствами (контроллерами, другими модулями) осуществляется через стек CANopen:

- PDO (Process Data Object) – обмен данными аналоговых входов в реальном времени;
- SDO (Service Data Object) – доступ к объектному словарю для настройки параметров модуля;
- NMT (Network Management) – управление состоянием модуля (Operational, Pre-operational, Stopped);
- Heartbeat / Node Guarding – мониторинг состояния модуля;

- LSS (Layer Setting Services) – настройка идентификаторов и параметров сети;
- Emergency – передача аварийных сообщений.

4.4. Обработка аналоговых сигналов

Взаимодействие с подсистемой обработки сигналов включает:

- Применение медианного фильтра, фильтра Калмана и алгоритмов скользящего среднего к входным аналоговым данным;
- Преобразование цифрового кода АЦП в физические значения (напряжение, ток, температура и т.д.) с программной настройкой диапазонов;
- Сбор и передачу диагностической информации о состоянии каналов.

4.5. Сбор и передача диагностической информации

Сбор и передача диагностической информации осуществляется через:

- Проверку работоспособности модуля и каналов;
- Запись событий системы для последующего анализа;
- Передачу аварийных сообщений по CANopen.

4.6. Взаимодействие с пользователем

Взаимодействие с пользователем осуществляется через:

- Визуальное отображение состояния модуля и входных/выходных сигналов;
- Доступ к объектному словарию через ASCII-протокол для настройки и диагностики;
- Удалённую настройку параметров модуля через протокол CANopen.

5 Технологии

5.1. Базовая платформа

- FreeRTOS;
- CANopen;
- CMSIS.

5.2. Поддерживаемые микроконтроллеры

- ARM Cortex-M.

5.3. Промышленные протоколы

- CANopen (CiA 301);
- CANopen LSS;
- CANopen Safety;
- PDO, SDO, NMT, Heartbeat, Emergency.

5.4. Периферийные интерфейсы

- SPI;
- UART;
- GPIO;
- SWD/JTAG.

5.5. Аналоговые и цифровые компоненты

- АЦП: SGM5860X, HX711;
- ЦАП: CBM128S085, DAC8775;
- Датчики: MAX31865 (термопара), MS5614 (давление), SD2421;
- Расширитель: MCP23S08.

5.6. Алгоритмы обработки сигналов

- Медианный фильтр;
- Фильтр Калмана;
- Скользящее среднее;
- Масштабирование;
- Антидребезг.

5.7. Хранение данных

- Флэш-память;
- EEPROM-эмуляция;
- XDD-файл.

5.8. Диагностика и отладка

- TraceRecorder;
- Самодиагностика;
- Светодиодная индикация.

5.9. Инструменты разработки

- CMSIS;
- STM32F0;
- SWD/JTAG.