Содержание

[Введение 4](#_Toc154135083)

[1 Наименование и область применения 4](#_Toc154135084)

[2 Техническая характеристика 4](#_Toc154135085)

[3 Обоснование и описание выбранной конструкции 5](#_Toc154135086)

[4 Структурная схема инженерной модели сверхмалого космического аппарата 6](#_Toc154135087)

[5 Бортовая система энергоснабжения и коммуникации 7](#_Toc154135088)

[5.1 Блок контроля мощности 8](#_Toc154135089)

[5.2 Блок мониторинга и распределения энергии 9](#_Toc154135090)

[5.3 Бортовые коммуникационные шины 10](#_Toc154135091)

[6 Функциональный модуль бортовой системы радиосвязи 11](#_Toc154135092)

[6.1 Блок хранения телеметрических данных 12](#_Toc154135093)

[7 Модуль системы бортового компьютера 12](#_Toc154135094)

[8 Модуль регистрации изображения и измерительных датчиков 14](#_Toc154135095)

[8.1 Электронный модуль регистрации изображения 14](#_Toc154135096)

[8.2 Бесконтактный датчик температуры 16](#_Toc154135097)

[8.3 Датчик ультрафиолетового излучения 18](#_Toc154135098)

[9 Система определения ориентации и стабилизации 18](#_Toc154135099)

[9.1 Модуль инерциальных датчиков ориентации 19](#_Toc154135100)

[9.2 Драйвера электромагнитных катушек 20](#_Toc154135101)

[10 Модуль датчиков направления на Солнце и системы оптического трекинга 21](#_Toc154135102)

[10.1 Датчик направления на Солнце 22](#_Toc154135103)

[10.2 Оптический датчик расстояния 23](#_Toc154135104)

[10.3 ИК-приемник 24](#_Toc154135105)

[10.4 ИК-светодиод 24](#_Toc154135106)

[10.5 Адресные светодиоды 24](#_Toc154135107)

[11 Модуль системы радионавигации, датчиков расстояния и лазерного указателя 25](#_Toc154135108)

[11.1 Модуль радионавигации 26](#_Toc154135109)

[11.2 Лазерный дальномер 27](#_Toc154135110)

[11.3 Лазерный целеуказатель 28](#_Toc154135111)

[12 Технология разработки программного обеспечения для микроконтроллера RP2040 29](#_Toc154135112)

[12.1 Разработка на языках программирования C и C++ при помощи Raspberry Pi Pico SDK 30](#_Toc154135113)

[12.2 Работа в среде MicroPython 31](#_Toc154135114)

[12.3 Работа в среде CircuitPython 33](#_Toc154135115)

[13 Технология разработки ПО для микроконтроллера STM32F030F4P6 34](#_Toc154135116)

[13.1 Установка и запуск среды STM32CubeIDE 34](#_Toc154135117)

[13.2 Конфигуратор STM32CubeMX 36](#_Toc154135118)

[13.3 Настройка контактов программатора 38](#_Toc154135119)

[13.4 Программа изменения состояния светодиода 39](#_Toc154135120)

[13.5 Компиляция программы 40](#_Toc154135121)

[13.6 Загрузка программы 41](#_Toc154135122)

[Список использованных источников 44](#_Toc154135123)

# Введение

В документе представлено техническое описание структурной и электрической схемы опытного образца инженерной модели сверхмалого космического аппарата (ИМ СМКА). Работа по созданию ИМ СМКА выполняется по техническому заданию в рамках договора №395/15 от 03.08.2021 «Разработка аппаратно-программного комплекса инженерных моделей сверхмалых космических аппаратов и наземного комплекса управления для отработки группового полета».

# Наименование и область применения

Инженерная модель сверхмалого космического аппарата разрабатывается для использования совместно с наземным комплексом управления в учебном процессе для подготовки специалистов аэрокосмической отрасли, для отработки группового полета и учебно-методической документации в системе подготовки специалистов по направлению «Группировки СМКА».

# Техническая характеристика

Технические характеристики инженерной модели СМКА представлены в таблице 1.

Таблица 1 ‑ Технические характеристики инженерной модели СМКА

|  |  |
| --- | --- |
| Параметры | Значения |
| Форм-фактор | CubeSat 3U |
| Система питания | Внешний источник 5 В, встроенные аккумуляторные батареи, солнечные панели |
| Диапазон температур | (15 – 35)°C |
| Относительная влажность | 40 - 80% |
| Частота приема по командной радиолинии | 433 - 445 МГц |
| Скорость приема по командной радиолинии | 1200 - 9600 бит/c |
| Частота передачи ответных квитанций по командной радиолинии | 433 - 445 МГц |
| Скорость передачи ответных квитанций по командной радиолинии | 1200 - 9600 бит/c |

Продолжение таблицы 1

|  |  |
| --- | --- |
| Параметры | Значения |
| Максимальная мощность передачи по командной и телеметрической радиолинии | Не менее10 dBm |
| Модуляция сигнала командной радиолинии | GMSK |
| Протокол передачи ответных квитанций командной радиолинии | AX.25 с дополнительной преамбулой |
| Частота передачи информации по телеметрической радиолинии | 433 - 445 МГц |
| Скорость передачи данных по телеметрической радиолинии | 1200 - 9600 бит/c |
| Модуляция сигнала телеметрической радиолинии | GMSK |
| Протокол передачи по телеметрической радиолинии | AX.25 |
| Бортовые интерфейсы передачи данных | SPI, I2C, USB, UART, CAN; |

# Обоснование и описание выбранной конструкции

В разработанной инженерной модели сверхмалого космического аппарата заложена идея «слотовой» компоновки. По наружным габаритам и компоновке несущего каркаса ИМ СМКА соответствует стандарту Cubesat 3U.

Габариты корпуса позволяют использовать ИМ СМКА в условиях учебных лабораторий. Слотовая компоновка предоставляет возможности комбинирования в наборе функциональных модулей для отработки различных исследовательских задач.

Внутри одной из продольных граней каркаса ИМ СМКА, выполненного из широко используемого в авиакосмической отрасли алюминиевого сплава Д16Т, надёжно закреплена «материнская» объединительная электронная печатная плата бортовой системы энергоснабжения и коммуникации, в опорной плоскости которой размещены 10 разъёмов для подключения электронных печатных плат всех функциональных модулей ИМ СМКА, которые вставляются в жёсткие разъемные соединения перпендикулярно «материнской» плате / параллельно торцевым панелям каркаса ИМ СМКА, а также имеют жесткую винтовую фиксацию на продольных панелях каркаса, образуя тем самым виртуальные (без направляющих полозьев) «слоты», приспособленные для оперативной частичной сборки/разборки конструкции с целью индивидуальной замены электронных печатных плат функциональных модулей.

Кроме того, на панелях (гранях) каркаса ИМ СМКА, в местах, свободных от требующих отрытого обзора элементов, располагаются солнечные панели и датчики направления на Солнце.

# Структурная схема инженерной модели сверхмалого космического аппарата

На рисунке 1 представлена структурная схема разработанной ИМ СМКА.



Рисунок 1 – Структурная схема инженерной модели  
сверхмалого космического аппарата

В структуре ИМ СМКА можно выделить следующие функциональные элементы:

– «материнскую» бортовую систему энергоснабжения и коммуникации, включая блоки контроля мощности, мониторинга и распределения энергии, бортовые коммуникационные шины, а также модули солнечных батарей и функциональные модули аккумуляторных батарей (2 шт. для резервирования);

– функциональный модуль системы бортового компьютера со встроенным электронным модулем регистрации изображения и измерительных датчиков;

– функциональные модули бортовой системы радиосвязи (2 модуля для резервирования);

– систему определения ориентации и стабилизации ИМ СМКА, включая функциональный модуль электромагнитной ориентации и стабилизации, модули датчиков направления на Солнце и системы оптического трекинга;

– функциональный модуль системы радионавигации ИМ СМКА со встроенным электронным модулем датчиков расстояния и лазерного указателя.

# Бортовая система энергоснабжения и коммуникации

Бортовая система энергоснабжения и коммуникации (БСЭиК) ИМ СМКА, обеспечивает электропитание других систем и модулей ИМ СМКА, а также коммуникацию между ними. БСЭиК во многом определяет конструкцию, массу и срок активного существования ИМ СМКА. Выход из строя БСЭиК приведёт к отказу всего СМКА, при этом резервирование БСЭиК может быть применено только для аккумуляторных и солнечных батарей.

В состав БСЭиК СМКА входят:

– модули солнечных батарей (первичный источник энергии);

– модули аккумуляторных батарей (вторичный источник электроэнергии);

– блоки контроля заряда, мощности, мониторинга и распределения энергии, включая зарядные и преобразующие устройства вторичных напряжений для электронных плат;

– бортовые коммуникационные шины.

БСЭиК СМКА в автоматическом режиме обеспечивает:

– совместную работу модулей многосекционной солнечной батареи на основе фотоэлектрических преобразователей и бортовых аккумуляторных батарей (АКБ) на общую нагрузку;

– выполняет функции стабилизации напряжений;

– осуществляет управление зарядом и разрядом АКБ, контроль температур, напряжений и токов на всех этапах преобразования энергии;

– поддерживает внутреннюю интерфейсную шину спутника для выдачи всей необходимой телеметрии в бортовую систему управления спутника.

Структурная схема внутренних взаимосвязей БСЭиК, с точки зрения энергетических потоков и узлов управления, представлена на рисунке 2.

Солнечная панель Z+

БМРЭ

Солнечная панель X+

Солнечная панель Y+

МК

АКБ 1

БКМ

АКБ 2

RBF

Слот 1

Слот 2

Слот 9

3.3 В

5 В

КЭ 1

КЭ 2

КЭ 9

Общая шина энергопитания

Общая информационная CAN-шина

CAN<->SPI

I2C#0 Rep

I2C#1 Rep

Общая сервисная I2C-шина

Рисунок 2 – Структурная схема внутренних взаимосвязей  
системы энергоснабжения и коммуникации СМКА

Структуру БСЭиК составляют следующие функциональные узлы:

– блок контроля мощности с модулями солнечной и аккумуляторных батарей;

– блок мониторинга и распределения энергии с коммутаторами энергопитания;

– бортовые коммуникационные шины.

## Блок контроля мощности

Солнечная энергия, преобразованная в электрическую с помощью фотоэлектрических преобразователей – солнечных панелей (на гранях каркаса разработанной ИМ СМКА может быть размещено до 13 панелей), собранных в модуль солнечной батареи, поступает в блок контроля мощности (БКМ), основу которого составляет программно-управляемая платформа многоканального контроля силы постоянного тока (микросхема INA3221), имеющая цифровой интерфейс I2C. Далее энергия поступает на схему зарядки аккумуляторных батарей АКБ 1 и АКБ 2. Оба модуля бортовых АКБ имеют схему контроля силы тока на базе INA3221, а также специализированные электронные модули заряда от солнечной батареи CN3065 и схему защиты батареи от условий использования, выходящих за технический регламент.

Альтернативным источником питания, а также зарядки бортовых АКБ в условиях подготовки и тестирования оборудования СМКА, является сервисный порт USB, расположенный на плате БСЭиК.

На борту ИМ СМКА в штатном режиме размещаются как основной, так и резервный модули бортовых АКБ. Электрическая энергия, запасённая в этих АКБ, поступает в блок мониторинга и распределения энергии (БМРЭ).

## Блок мониторинга и распределения энергии

БМРЭ базируется на управляющем микроконтроллере (МК) RP2040, размещённом на платформе разработки Raspberry Pi Pico. В схеме подключения МК имеется встроенная схема контроля напряжения питания на микросхеме ADM706, обрабатывающая сигнал контрольного таймера от МК и генерирующая сигнал сброса. Основным источником питания МК является цепь АКБ, в качестве дополнительного источника питания может быть использована цепь питания порта USB внешнего компьютера при его подключении к разъему microUSB на плате БСЭиК. МК также имеет измерительные каналы напряжения в цепи батарейного питания, цепи солнечных панелей, а также в цепи питания от сервисного разъема microUSB.

В составе БМРЭ имеются десять коммутаторов энергопитания (КЭ) для каждого слота электронных печатных ИМ СМКА. КЭ реализованы на базе микросхемы переключателя нагрузки с ограничением по току AP22814. Номинальное значение величины предела по потреблению тока – 3 А. Контроль тока с цепи каждого слота выполняется с помощью монитора тока и напряжения на микросхеме INA219. В случае превышения установленного лимита по каналу потребления, коммутатор закрывается и выдает адресное сообщение на шину данных, где фиксируются все события о превышении лимита по энергопотреблению. Управление КЭ и сбор информации о превышении энергетических затрат выполняется на базе двух 8-битных расширителей цифровых входов/выходов для шины I2C (микросхемы PCA9554), программно-управляемых от МК по шине I2C.

В схеме БМРЭ также установлены вторичные преобразователи напряжения TPS 61030 и NCP 1529, которые нестабилизированное напряжение от солнечных панелей и АКБ преобразуют в необходимые источники вторичных напряжений (соответственно 5 В и 3,3 В) используемых для питания модулей солнечных датчиков.

В схеме модуля МК имеется преобразователь системного питания от АКБ в напряжение 3,3 В. Питание используется потребителями в составе функциональных модулей ИМ СМКА.

БСЭиК ИМ СМКА оборудована технологическим ключом RBF (от англ. Remove Before Flight – «извлечь перед полётом»), который позволяет прерывать энергетический поток от АКБ в единую шину энергоснабжения ИМ СМКА для его длительного хранения в обесточенном состоянии. Подзарядка бортовых аккумуляторных батарей при этом возможна от внешнего источника напряжения или порта USB компьютера.

Источники вторичных напряжений подключаются к индивидуальным потребителям (слотам) с помощью КЭ БМРЭ. В качестве таких потребителей энергии выступают:

– контроллер БМРЭ;

– модуль системы бортового компьютера;

– модули радиосвязи 1 и 2;

– система определения ориентации и стабилизации;

– модуль радионавигации;

– другие модули целевой нагрузки.

После подключения напряжения к МК БМРЭ, а это произойдет лишь при извлеченном ключе RBF, стартует программное обеспечение МК, которое запускает алгоритм мониторинга бортового напряжения и токов потребления, и осуществляет заданную коммутацию энергетических потоков для бортовых модулей и подсистем, согласно логике этапов полетного алгоритма.

Модуль МК осуществляет временную привязку команд и параметров к бортовому времени ИМ СМКА, сбор и обработку телеметрической информации о параметрах работы бортовых систем ИМ СМКА.

## Бортовые коммуникационные шины

Командами от наземного центра управления полётами можно управлять этапами полетного алгоритма и осуществлять удалённую коммутацию энергии для бортовых модулей целевой нагрузки, так как все служебные модули ИМ СМКА объединены в единую информационную CAN-шину. Таким образом, модули бортовой системы радиосвязи осуществлять дублирование функций БМРЭ.

В БМРЭ имеется преобразователь двунаправленной информационной шины SPI МК в шину CAN, являющуюся общим информационной шиной ИМ СМКА. Преобразователь выполнен на элементах MCP2515Т (уровень протокола) и SN65HVD234D (физический уровень). Подобная схема преобразования применяется на всех бортовых модулях ИМ СМКА, имеющих микропроцессорный электронный модуль на базе RP2040.

В дополнение к шине CAN, ИМ СМКА имеет также две общие для бортовых модулей коммуникационные шины I2C. В составе БМРЭ имеется два ретранслятора шины I2C, разделенных физически. Это реализовано на элементе PCA9512ADP. Шина I2C#0 используется для коммутации слотов и контроля параметров энергопотребления. Шина I2C#1 выполняет функции резервной информационно-сервисной шины. Схема ретрансляции повторяется на всех электронных платах функциональных модулей ИМ СМКА.

В центральной секции группы слотов (номера слотов 4–7) размещается также модуль энергонезависимого памяти типа EEPROM емкостью 512 Кибит на базе микросхемы 24LC512T. Эта память может быть использована для хранения конфигурационной информации для бортовой системы энергоснабжения и коммуникации. Интерфейс подключения – I2C#0.

# Функциональный модуль бортовой системы радиосвязи

Структурная схема функционального модуля бортовой системы радиосвязи представлена на рисунке 3.

Разъем слота

СAN-контроллер

E10-433MD-SMA

Микроконтроллер

RP2040

EEPROM

SPI

SPI

CAN

Энергопитание из бортовой шины

I2C

Рисунок 3 – Структурная схема бортовой системы радиосвязи СМКА

Для реализации приемопередатчиков бортовой системы радиосвязи ИМ СМКА используется специализированная микросхема Si4463 фирмы Silicon Labs. Ее основные характеристики:

– диапазон частот – от 119 до 1050 MГц;

– чувствительность приема – минус 126 дБм;

– модуляция – (G)FSK, 4(G)FSK, (G)MSK, OOK;

– максимальная выходная мощность – 20 дБм;

– скорость передачи – от 100 бит/с до 1 Мибит/с;

– напряжение питания – от 1,8 до 3,6 В;

– буфер – 64 байта FIFO на прием и передачу;

– автоматическая подстройка частоты (AFC) и регулировка усиления (AGC);

– детектор низкого уровня питания, датчик температуры.

Микросхема Si4463 используется в составе электронного радиомодуля приёмо-передатчика E10-433MD-SMA. Он содержит кварцевый резонатор и пассивные компоненты согласования, необходимые для корректного функционирования Si4463, а также предоставляет доступ к выводам GPIO0-3 и шине SPI. Управление электронным радиомодулем осуществляется программно, с помощью микроконтроллера RP2040. Для хранения конфигурируемых параметров электронного радиомодуля используется микросхема EEPROM 24LC512.

Контроллер СAN-шины, реализованный по схеме, аналогичной описанной выше в разделе описания бортовой системы энергоснабжения и коммуникации, обеспечивает сопряжение радиомодуля бортовой системы радиосвязи с единой бортовой информационной шиной ИМ СМКА.

## Блок хранения телеметрических данных

Для хранения данных телеметрии в ИМ СМКА используется Flash память W25Q32JVSSIM с последовательным интерфейсом SPI, имеющая объем 32 Мибит.

В качестве EEPROM для хранения заводских настроек бортовых электронных модулей (например, номинальной частоты каналов радиосвязи для бортового радио-модуля, позывного спутника и т. п.) используется микросхема 24LC512 (фирма Atmel) объемом 512 Kибит, с защитой от (пере)записи и длительностью срока хранения информации более 200 лет.

# Модуль системы бортового компьютера

Система бортового компьютера является центральным элементом ИМ СМКА. Основные функции бортового компьютера – управление основной полетной программой и полезной научной нагрузкой, хранение телеметрии, управление ориентацией и стабилизацией ИМ СМКА.

На рисунке 4 представлена структурная схема модуля системы бортового компьютера с блоком регистрации изображения и измерительных датчиков.

В составе модуля бортового компьютера имеется два встроенных компьютерных модуля – основной и комплементарный. Основной выполняет функции бортового компьютера. Комплементарный – управляет оборудованием полезной нагрузки, включающей блок регистрации изображения, а также датчики ультрафиолетового излучения и температуры.

На рисунке 4 также отражены все коммуникационные связи между составляющими блоками. Модуль основного компьютера через разъём слота подключается к общим для ИМ СМКА коммуникационным интерфейсам CAN и I2C, а также к общей силовой сети питания.

Модуль комплементарного компьютера связан с основным по шине UART, а также сигналом разрешения на включения линии питания для блоков полезной нагрузки.

Подключение комплементарного компьютера к датчикам производится по шине I2C, к модулю камеры – по шине данных и шине управления специфичных для применяемого в модуле оборудования.

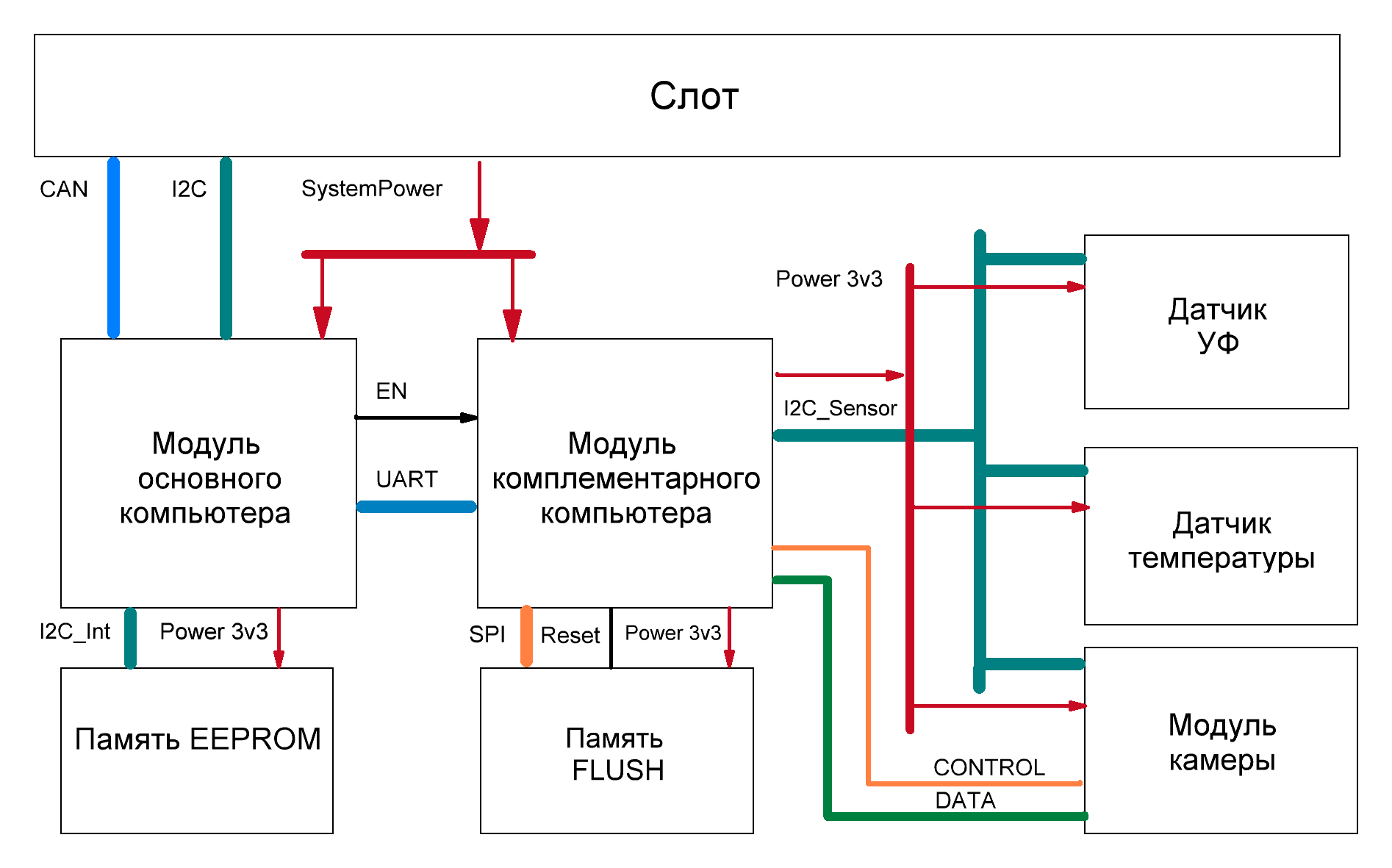


Рисунок 4 – Структурная схема модуля системы бортового компьютера  
с блоком регистрации изображения и измерительных датчиков

Для унификации электрических схем и процесса разработки программного обеспечения для всех бортовых модулей ИМ СМКА в качестве основного микропроцессорного блока был выбран микроконтроллер RP2040 на отладочной плате Raspberry Pi Pico, внешний вид показан на рисунке 5.

Это достаточно производительный двухядерный процессор с низким энергопотреблением и гибкой периферией.

Основные технических характеристик RP204:

– два ядра ARM Cortex-M0+, тактовая частота – 133 МГц;

– объём оперативной памяти – 264 КиБ;

– объём FLASH-памяти – 2 МиБ;

– наличие DMA-контроллера;

– наличие 12-разрядных аналоговых входов – 3;

– наличие портов UART – 2;

– наличие два порта SPI;

– наличие портов I2C –2;

– наличие PWM (ШИМ) каналов – 16;

– наличие встроенного датчика температуры;

– наличие GPIO контактов – 30;

– наличие MicroUSB-B порт с USB 1.1 контроллером и поддержкой хоста;

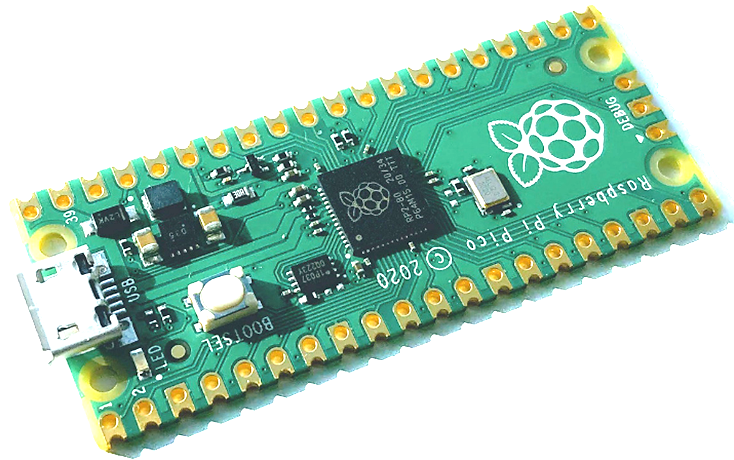
– поддержка режимов пониженной частоты для снижения потребления.

Рисунок 5 – Отладочная плата Raspberry Pi Pico

На отладочной плате Raspberry Pi Pico имеется преобразователь питания на 3,3 В понижающе-/повышающего типа (buck-boost) на микросхеме Richtek RT6150B. Благодаря этому отладочная плата имеет широкий диапазон входного питания в пределах (1,8–5,5) В.

Разработчикам кода для RP2040 доступны следующие варианты программного обеспечения: C/C++ с использованием предлагаемого Pico SDK; CircuitPython для Pico; MicroPython для Pico.

# Модуль регистрации изображения и измерительных датчиков

В состав модуля регистрации изображения и измерительных датчиков входят следующие компоненты:

– электронный модуль регистрации изображения;

– бесконтактный датчик температуры;

– датчик ультрафиолетового излучения.

## Электронный модуль регистрации изображения

В качестве целевой нагрузки ИМ СМКА, в составе модуля регистрации изображения и измерительных датчиков, была выбрана видеокамера, выполненная в виде электронного модуля (отладочной платы) VGA камеры OV7670, внешний вид которого приведен на рисунке 6.

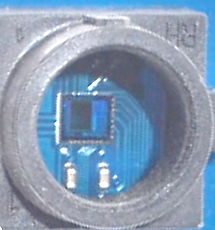




Рисунок 6 – Отладочная плата VGA-модуля OV7670

Основные характеристики модуля:

– разрешение сенсора – 640х480 точек;

– фокусное расстояние оптической системы – 1/6 дюйма;

– напряжение питания – 3,3 В;

– ток потребления – 20 мА;

– отношение сигнал/шум – 46 дБ;

– динамический диапазон – 52 дБ;

– форматы передачи цвета – RAW RGB;

– методы автокоррекции – AEC, AGC, AWB, ABF, ABLC;

– интерфейсы – SCCB (совместим с I2C) и параллельный 8 бит.

VGA-модуль OV7670 имеет цифровой выход, совместимый с различными микроконтроллерами. Высокая чувствительность позволяет работать в условиях низкой освещенности. По интерфейсной шине управления можно настроить качество изображения, формат данных и режим передачи информации по интерфейсной шине данных. Особенности модуля позволяют автоматически поддерживать высокое качество изображения путем уменьшения или устранения зашумленности, выравнивания баланса цвета, повышения четкости изображения, установки оптимальной насыщенности, контрастности, гаммы и оттенка изображения.

Обработка изображения настраивается записью данных в специальные регистры микросхемы OV7670 с помощью интерфейса Serial Camera Control Bus (SCCB) – аналога шины I2C. Частота дискретизации 30 кадров/с соответствует стандарту VGA. Скорость передачи кадров устанавливается программно. Также доступны форматы: QVGA 320х240, CIF 352х240, QCIF 176x144. Разрешающая способность может принудительно снижаться до 40x30 точек. Данные о цвете передаются с помощью установленной программистом кодировки. Используются два основных типа кодирования – YCbCr и RGB, который, в свою очередь, имеет три варианта: RGB565, RGB555, RGB444 (цифры означают количество бит на один цвет, например, RGB565 это 5 бит на красный, 6 бит на зеленый и 5 бит на синий цвета). В случае кодировки RGB для передачи данных о цвете пикселя потребуется 2 байта. Передача кодировки цвета YCbCr сложнее. По умолчанию установлен формат VGA 640х480, 30 кадров. На выходе HREF – сигнал с частотой 14,4 кГц (частота обусловлена параметрами изображения 30 кадров х 480 строк = 14400 Гц).

Изображение объектов, находящихся перед объективом камеры, фокусируется на поверхности микросхемы OV7670, расположенной на плате под объективом. Микросхема преобразует фотоинформацию в цифровой вид и обеспечивает передачу данных по интерфейсу. Для питания микросхемы на плате смонтированы два интегральных стабилизатора напряжений – на 2,8 В и 1,8 В. В схему устройства входит несколько пассивных компонентов, обеспечивающих работу стабилизаторов и светочувствительной микросхемы.

Модуль регистрации изображений на основе оптического сенсора OV7670 с помощью комплементарного процессорного узла модуля системы бортового компьютера и позволяет реализовать поддержку режима имитации целевых задач дистанционного зондирования Земли, отработку алгоритмов бортовой системы ориентации и стабилизации, отработку алгоритмов взаимной ориентации нескольких СМКА при имитации группового полета.

Модуль измерительных датчиков представлен измерительной платформой, которая содержит средства измерения температуры и детектирования УФ излучения. Ознакомится с ресурсами по программному управлению модулем можно по ссылке [1].

## Бесконтактный датчик температуры

В состав модуля регистрации изображения и измерительных датчиков ИМ СМКА включён измерительный модуль датчика с бесконтактным считыванием температуры объекта посредством приема и преобразования инфракрасного излучения на основе компонента MLX90614, внешний вид которого представлен на рисунке 7.

Основные характеристики датчика:

– напряжение питания – от 3,3 до 5 В;

– ток потребления – не более 23 мA;

– диапазон применения – от минус 40 °C до 85 °C;

– диапазон измерения – от минус 40 °C до 300 °C;



Рисунок 7 – Бесконтактный датчик температуры

– абсолютная погрешность в условиях (0 °C ~ 50 ℃) – до ±0.5 °C;

– частота обновления – от 0,5 до 64 Гц;

– поле зрения (FOV) – 55°х35°;

– интерфейс – I2C (адрес 0х33 по умолчанию);

– размер – 28x16 мм.

Датчик может работать в трех режимах: «термостат»; «ШИМ»; «SMBus».

В режиме «термостат» датчику не требуется контроллер, он регулирует температуру в заданных пределах, управляя драйвером нагрузки открытым стоком. В режиме «ШИМ» на выходе датчика появляется широтно-импульсно модулированный сигнал (ШИМ), скважность которого зависит от температуры. Для подключения к контроллеру используется режим «SMBus». Так как этот протокол электрически и сигнально совместим с I2C, то с датчиком работают, используя аппаратный I2C. Все режимы датчика настраиваются записью в определенные ячейки EEPROM. По умолчанию датчик находится в режиме «SMBus».

Датчик в своем составе содержит два устройства: инфракрасный термоэлектрический детектор (обнаруживающий элемент) и вычислительное устройство, построенное на принципах цифровой обработки сигналов. Принцип работы датчика основан на законе Стефана–Больцмана, который говорит о том, что каждое нагретое тело излучает инфракрасную энергию, интенсивность которой прямо пропорциональна температуре этого тела. Обнаруживающий элемент датчика измеряет какое количество энергии инфракрасной энергии излучается выбранным объектом, а вычислительный блок конвертирует это значение энергии в значение температуры используя встроенный 17-битный аналого-цифровой преобразователь (АЦП).

Информация о температуре передается по [интерфейсу I2C](https://microkontroller.ru/arduino-projects/ispolzovanie-interfejsa-i2c-v-arduino-polnoe-rukovodstvo/), физически связанному с шиной I2C комплементарного компьютера. Конфигурация датчика и управление режимом измерения температуры производится программно. Ознакомится с ресурсами по программному управлению датчиком можно по ссылке [2].

## Датчик ультрафиолетового излучения

На рисунке 8 представлен внешний вид модуля датчика ультрафиолетового излучения. Модуль базируется на компоненте Si1145 со встроенным АЦП. Он определяет не только ультрафиолет, но и интенсивность окружающего света. Имеет непосредственный вывод значения УФ-индекса через интерфейс I2C. Поддерживает выход прерывания, программируемый верхний/нижний порог. Имеет встроенный преобразователь напряжения, совместимый с рабочим напряжением 3,3 В / 5 В.

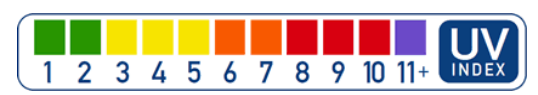
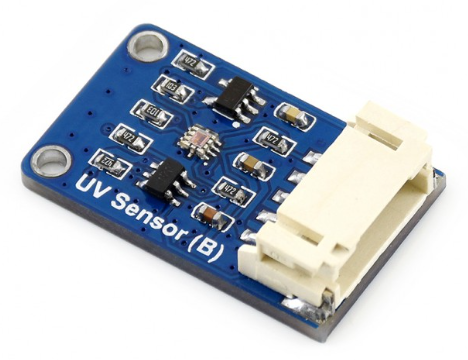


Рисунок 8 – Модуль датчика УФ

Информация об индексе УФ передается по [интерфейсу I2C](https://microkontroller.ru/arduino-projects/ispolzovanie-interfejsa-i2c-v-arduino-polnoe-rukovodstvo/), физически связанному с шиной I2C комплементарного компьютера. Программно производится конфигурация датчика и управление режимом измерения.

Ресурсы по программному управлению датчиком находятся по ссылке [3].

# Система определения ориентации и стабилизации

Функция системы определения ориентации и стабилизации ИМ СМКА – реализация алгоритмов определения ориентации с помощью систем различного типа, а также стабилизации и управления угловым положением с помощью имеющихся аппаратных средств ИМ СМКА (магнитных катушек).

На рисунке 9 представлена структурная схема функционального модуля системы электромагнитной ориентации и стабилизации.

Как и остальные функциональные модули, он устанавливается в один из слотов ИМ СМКА, тем самым осуществляя подключение к бортовой системе питания и бортовым коммуникационным шинам CAN и I2C.

Как и в остальных функциональных модулях, в качестве управляющего микропроцессора модуля системы электромагнитной ориентации и стабилизации используется встроенный электронный микропроцессорный модуль на основе Raspberry Pi RP2040.

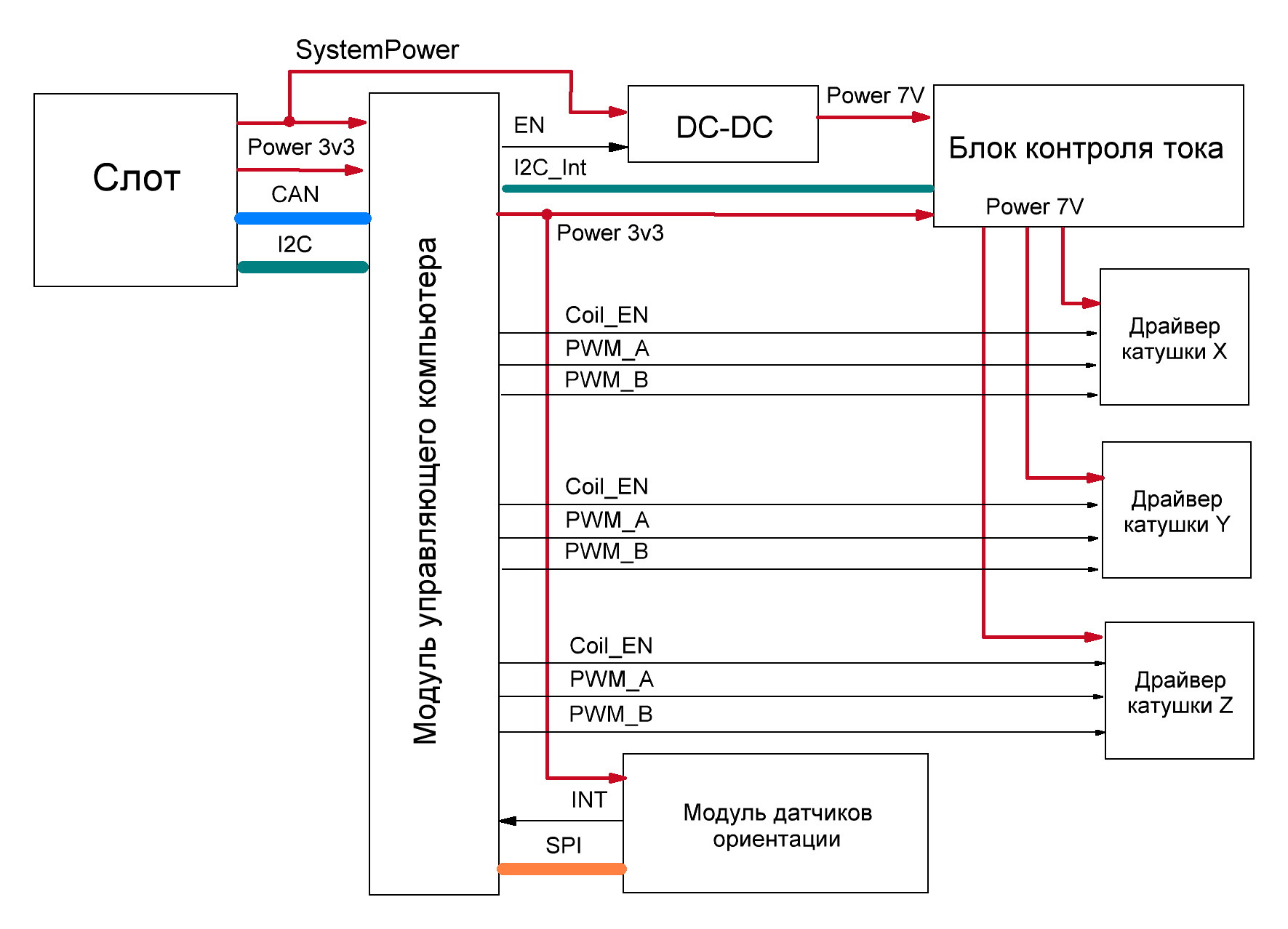


Рисунок 9 – Структурная схема модуля системы электромагнитной ориентации и стабилизации

## Модуль инерциальных датчиков ориентации

Один из эффективных из способов определения ориентации космического аппарата – использование сенсоров инерциального типа. Этот способ был реализован в ИМ СМКА применением девяти-осевого MEMS-сенсора MPU9250. В миниатюрном корпусе микросхемы MPU-9250 объединены два кристалла: на одном размещён трёх-осевой гироскоп, реагирующий на изменение углов ориентации в пространстве и трёх-осевой акселерометр, который сравнивает проекцию ускорения объекта с гравитационным ускорением и способен измерять линейную скорость объекта, а на втором кристалле – трёх-осевой магнитометр AK8963 компании Asahi Kaisei Microdevices, предназначенный для измерения интенсивности магнитного поля, действующего на объект. Данные с сенсоров оцифровываются 16-битными АЦП, обрабатываются цифровым сигнальным процессором перемещения DMP (Digital Motion Processor), с применением алгоритмов Motion Fusion, обрабатывающих данные от всех внутренних сенсоров, собирают полный набор данных, которые передаются управляющему микроконтроллеру по шинам SPI или I2C. MPU925 характеризуется низким потреблением, низкой стоимостью и высокой производительностью.

Основные технические характеристики MEMS-сенсора MPU9250:

– напряжение питания – (3–5) В;

– интерфейс – SPI (1 МГц) / I2C (400 кГц);

– диапазон измерения акселерометра – ±2 g; ±4 g ; ±8 g; ±16 g;

– диапазон измерения гироскопа – ±250; ±500; ±1000; ±2000°/с;

– чувствительность гироскопа – 131; 65,5; 32,8; 16,4 LSB/°/c;

– диапазон измерения компаса магнитометра – ±4800 мкТл;

– буфер – тип FIFO, объём 512 байт;

– ток потребления – 3,2 мА (гироскоп); 450 мкА (акселерометр); 280 мкА (магнитометр);

– ток в режиме «сна» – 16 мкА (гироскоп и акселерометр).

MPU-9250 имеет программируемую систему прерываний, которая может генерировать сигнал прерывания на выводе INT. Флаги состояния указывают источник прерывания. Источники прерываний можно включать и отключать индивидуально.

MPU-9250 связан с управляющим компьютером модуля по цифровой шине SPI. Также используется вывод INT для программного управления со стороны блока микропроцессора.

Данные основных датчиков определения ориентации передаются по основной CAN-шине к бортовой системе радиосвязи ИМ СМКА для дальнейшей обработки, формирования пакетов телеметрии основных датчиков ориентации и их передачи на командный центр.

## Драйвера электромагнитных катушек

На основе данных, полученных от системы определения ориентации и стабилизации ИМ СМКА, наземный командный центр определяет величины вектора коррекции положения ИМ СМКА в пространстве для выполнения определенной полетной миссии. Далее, с помощью программно-управляемых элементов трёх ортогонально расположенных электромагнитных катушек Х, Y, Z, наводится магнитное поле в направлении вектора коррекции положения.

Управление электромагнитными катушками (их «накачка», то есть, формирование в них электрического тока заданной величины) производится с помощью специализированных элементов – A3918SESTR-T (далее A3918), представляющих собой драйвер двигателя постоянного тока низкого напряжения.

Основное назначение этого элемента – управление низким напряжением двигателя постоянного тока с помощью широтно-импульсной модуляции (ШИМ). A3918 рассчитан на выходной ток до 1,5 А и рабочее напряжение от 2,5 до 9 В. Для питания A3918 в схеме используется повышающий преобразователь питания на элементе МТ3608. A3918 имеет внутренний таймер ШИМ с фиксированным временем выключения, который устанавливает пиковый ток на основе выбора номинала резистора на клемме датчика тока. Предусмотрен выходной флаг перегрузки по току, который уведомляет пользователя, когда ток в обмотке двигателя достигает пикового значения. Элемент A3918 управляется программно от управляющего компьютера. Сигнал Coil\_EN используется для включения/отключения выходов, к которым подключена нагрузка. Состояние на сигнальных линиях PWM\_А, PWM\_В определяет направление тока в цепи катушек, а в результате – полярность индуцируемого поля.

Контроль силы тока в катушках выполняется на базе элемента INA3221, управляемого по шине I2C от модульного компьютера. Эта опция заменяет функцию выходного флага перегрузки по току в A3918.

# Модуль датчиков направления на Солнце и системы оптического трекинга

Структурная схема модуля датчиков направления на Солнце и системы оптического трекинга представлена на рисунке 10.

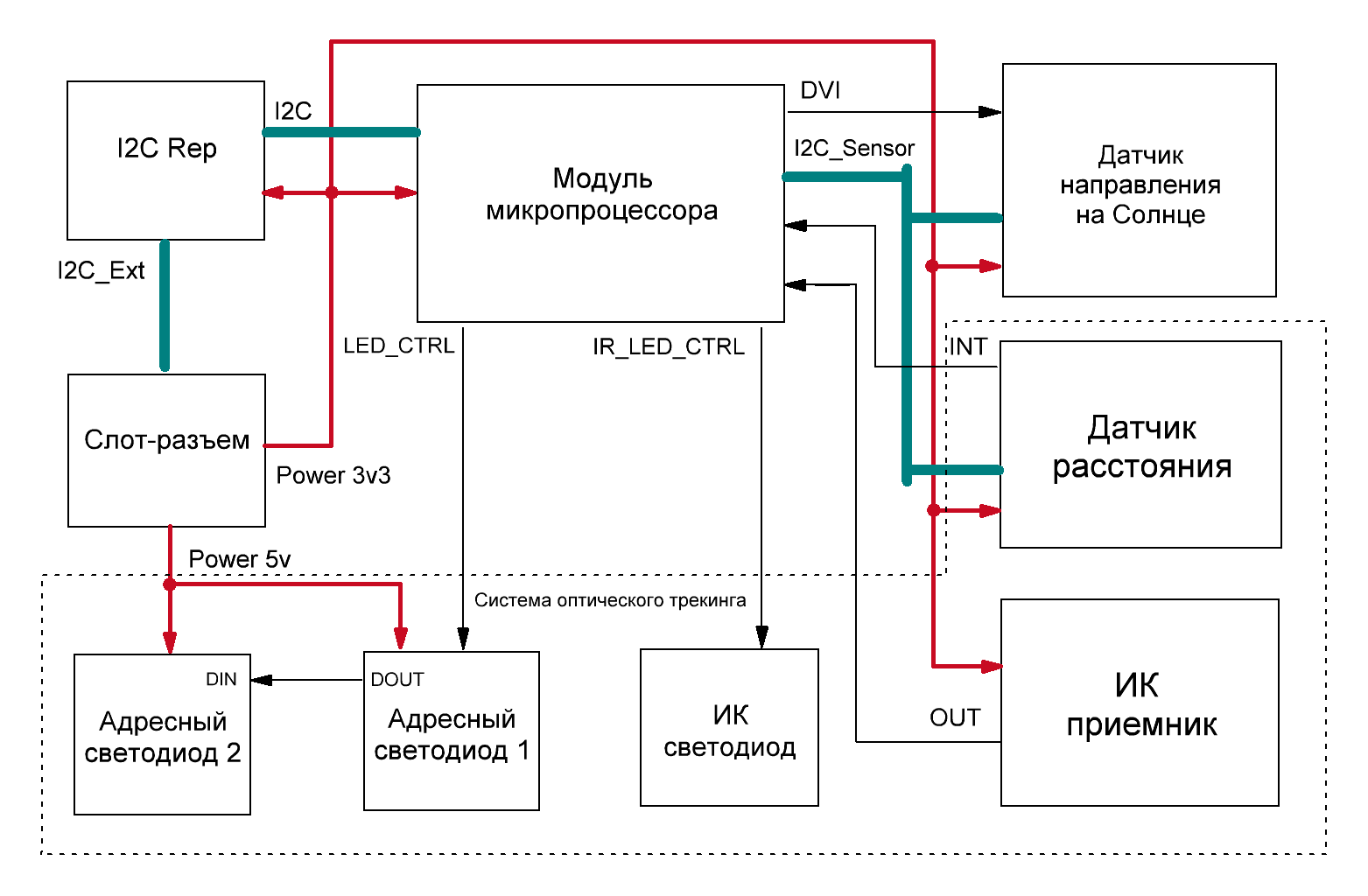


Рисунок 10 – Структурная схема модуля датчиков направления на Солнце и системы оптического трекинга

Рассматриваемый модуль размещается на шести гранях СМКА. Каждый модуль подключается шлейфом к разъему-слоту на плате бортовой системы энергоснабжения и коммуникации. Подключение производится к шинам питания и шине цифрового интерфейса I2C. Схема модуля содержит ретранслятор шины I2C, выход которого подключен к микропроцессорному модулю.

Основу модуля составляет управляющий компьютер, в качестве которого использован микроконтроллер STM32F030F4P6 компании ST Microelectronics, имеющий высокопроизводительное ядро ARM Cortex-M0 RISC разрядностью 32 бита и частотой 48 МГц, высокоскоростной памятью, широким выбором периферии и I/O. В нем имеются стандартные интерфейсы связи, один 12-битный АЦП, семь универсальных 16-битных таймеров и ШИМ таймер с расширенными возможностями управления.

Все устройства, встроенные в схему модуля датчиков направления на Солнце и системы оптического трекинга, управляются программно. Для управления используется шина I2C микроконтроллера или выделенные сигналы. Микроконтроллер программируется посредством специального программатора, подключаемого через разъем на плате модуля.

## Датчик направления на Солнце

В качестве дополнительных датчиков определения ориентации ИМ СМКА на всех его гранях установлены датчики направления на Солнце (всего – 6 штук), в качестве которых применяются цифровые датчики внешней освещенности на базе элемента BH1750FVI.

В датчике освещенности, помимо чувствительного элемента, имеется также 16-битный АЦП преобразования сигнала в цифровую форму. BH1750FVI, при совместном использовании с микроконтроллером, позволяет определять интенсивность света в пределах от 1 до 65535 Лк.

Основные технические характеристики датчика BH1750FVI:

– напряжение питания – (3 – 5) В;

– цифровой выход – I2C 16 бит;

– изменение чувствительности/усиления – в диапазоне (0,45 – 3,68);

– изменение разрешающей способности;

– калибровка – в диапазоне (0,96 – 1,44);

– прямой цифровой (числовой) выход, не требующий расчетов;

– диапазон измерения уровня яркости – (0 – 65535) Лк.

Примерные показатели уровня освещенности при разных условиях:

– ночь – (0,001 ~ 0,02) Лк;

– лунная ночь – (0,02 ~ 0,3) Лк;

– облачно (в помещении) – (5 ~ 50) Лк;

– облачно (под открытым небом) – (50 ~ 500) Лк;

– солнечный день (в помещении) – (100 ~ 1000) Лк;

– солнечный день (под открытым небом) – до 100000 Лк.

Ресурсы по программному управлению датчиком находятся по ссылке [4].

## Оптический датчик расстояния

Оптический (инфракрасный диапазон) датчик расстояния предназначен для измерения расстояния до объекта. Датчик преобразует изменение расстояния от датчика до объекта в изменение величины выходного сигнала. Он начинает работать при обнаружении объекта в его рабочей зоне и выключается при его отсутствии. Преимущественно работает по оси Х.

В качестве датчика расстояния в модуле используется элемент SI1143-A11-GM компании Silicon Labs. Это маломощная интегральная схема инфракрасного датчика приближения с открытым стоком, основанная на коэффициенте отражения, с интерфейсом I²C и выходом прерывания по программируемым событиям (выход INT на рисунке 10).

Бесконтактная сенсорная микросхема элемент SI1143-A11-GM включает в себя аналого-цифровой преобразователь, встроенные высокочувствительные видимые и инфракрасные фотодиоды, цифровой сигнальный процессор. Микросхема обеспечивает высокие характеристики в широком динамическом диапазоне и при различных источниках света, включая прямой солнечный свет. Реакция фотодиода и связанная с ним схема цифрового преобразования обеспечивают высокую устойчивость к мерцанию искусственного света и шуму флаттера естественного света.

При наличии двух или более светодиодов микросхема датчика способна поддерживать многоосное обнаружение движения. Высокая чувствительность к отражению – менее 1 мкВт/см². Имеется внутренняя и внешняя поддержка пробуждения, встроенный монитор напряжения питания и контроллер сброса при включении питания. Последовательная связь со скоростью передачи данных до 3,4 Мбит/с.

Основные технические характеристики датчика:

– метод измерения – оптический;

– измеряемое расстояние – от 10 до 2000 мм;

– выходная конфигурация – цифровой выход, открытый сток;

– рабочий ток источника питания – 4,3 мА;

– рабочее напряжение питания – от 1,71 до 3,6 В.

Ресурсы по программному управлению датчиком находятся по ссылке [5].

## ИК-приемник

ИК-приемник, показанный на рисунке 10, выполнен на элементе TSOP34833. Рассматриваемый ИК-приемник является миниатюрным модулем приемника кодированного оптического излучения, предназначенным для инфракрасных систем дистанционного управления с максимальной дальностью до 45 метров. Угол половинной дистанции приема – ±45°.

Чувствительный к излучению PIN-диод, защищенный инфракрасным фильтром, и предусилитель упакованы в общий трехвыводной корпус из эпоксидной смолы размерами 6х7х5,6 мм.

Специальная конструкция модуля TSOP34833 обеспечивает повышенную невосприимчивость к высокочастотным, радиочастотным и оптическим помехам, а также высокое подавление пульсаций и шумов напряжения питания, лежащего в диапазоне от 2,5 до 5,5 В. Выходным каскадом ИК-приемника является маломощный NPN-транзистор с резистивной нагрузкой в цепи коллектора, причем его максимальный втекающий ток не должен превышать 5 мА. Демодулированный выходной сигнал приемника (вывод OUT на рисунке 10) подан непосредственно на логический вход микроконтроллера для дальнейшей обработки.

## ИК-светодиод

На плате модуля датчиков направления на Солнце и системы оптического трекинга установлен планарный светодиод инфракрасного диапазона свечения KA-3528F3C. Его основные технические характеристики:

– мощность излучения P – 8 мВт;

– прямое напряжение – 1,5 В (при токе Iпр = 50 мА);

– длина волны – 940 нм;

– ширина спектра излучения – 50 нм;

– видимый телесный угол – 120 град;

– максимальное обратное напряжение – 5 В;

– максимальный прямой ток – 50 мА;

– максимальный импульсный прямой ток – 1200 мА.

## Адресные светодиоды

Каждая плата модуля имеет по два адресных светодиода WS2812B. Их основные технические характеристики:

– размер – 5х5 мм;

– частота ШИМ – 400 Гц;

– напряжение питания – 5 В;

– потребление при нулевой яркости (на 1 светодиод) – 1 мА;

– потребление при максимальной яркости (на светодиод) – 60 мА;

– цветность – RGB, 256 оттенков на канал, 16 миллионов цветов;

– размер данных – 24 бита на светодиод;

– скорость передачи данных – 800 кГц.

Пара светодиодов управляется от сигнала микроконтроллера Led\_CTRL. Светодиоды соединяются последовательно (рис. 10) согласно порядку применения.

Ознакомится с ресурсами по программному управлению адресными светодиодами можно по ссылке [6].

# Модуль системы радионавигации, датчиков расстояния и лазерного указателя

Структурная схема модуля радионавигации, датчиков расстояния и лазерного указателя представлена на рисунке 11.

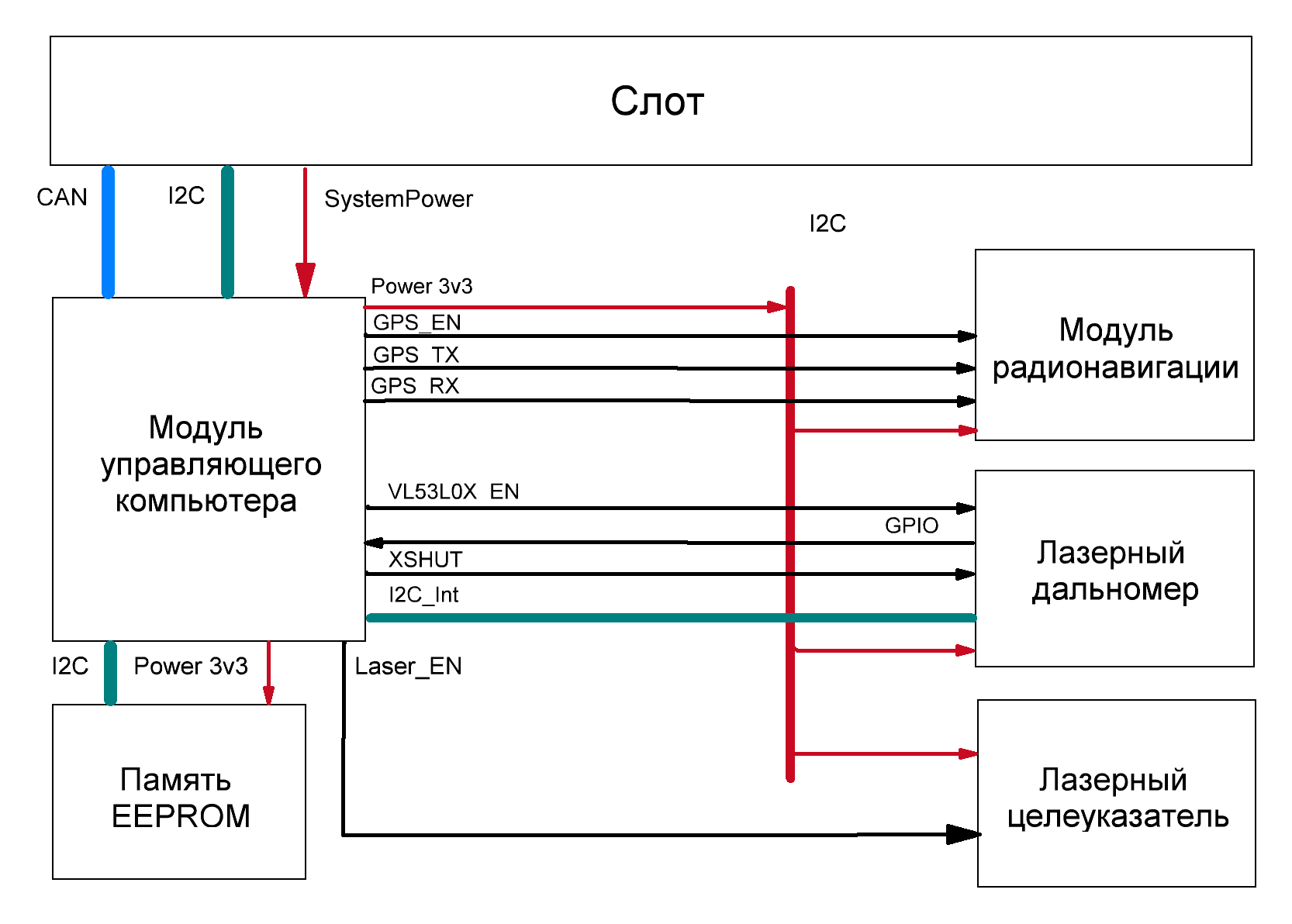


Рисунок 11 – Структурная схема модуля радионавигации, датчиков расстояния и лазерного указателя

Рассматриваемый модуль устанавливается в один из слотов бортовой системы энергоснабжения и коммуникации. К внешней стороне платы модуля фиксируются части каркаса для крепления антенны блока радионавигации, а также фиксируются несущие детали датчика расстояния и лазерного указателя.

В основе представленной схемы лежит модуль микропроцессора RP2040. Все устройства в схеме рассматриваемого модуля управляются программно. Для управления используется шины UART и I2C микроконтроллера или выделенные сигналы.

## Модуль радионавигации

На рисунке 12 представлена плата GPS GY-GPSV3 NEO-M8N. Она используется в качестве модуля радионавигации используется плата – GPS-модуль, позволяющий получать координаты устройств с помощью системы глобального позиционирования.

Модуль построен на базе GPS-чипа Ublox NEO-M8, имеет встроенную память и резервную батарею для хранения настроек. Его основные технические характеристики:

– напряжение питания – от 2,7 до 3,6 В;

– интерфейс – UART;

– поддержка навигационных систем – GPS/QZSS L1 / GLONASS L10F / BeiDou B1 / Galileo E1B/C;

– время старта – 26 с (холодный старт); 2 с (горячий старт);

– точность позиционирования – до 2 м;

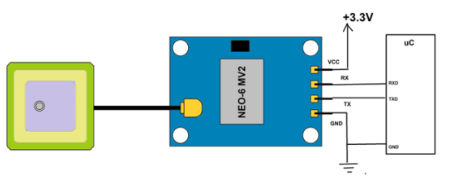
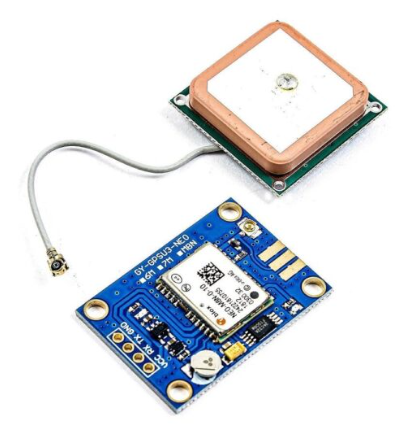


Рисунок 12 – Модуль GPS GY-GPSV3 NEO-M8N

– часы реального времени (RTC);

– поддержка активной и пассивной антенны.

Модуль программно управляется от микропроцессора по интерфейсу UART. Сигналы GPX\_RX, GPX\_TX представлены на рисунке 11. VL530X\_EN - сигнал разрешения, который коммутирует цепь питания.

Ознакомится с ресурсами по программному управлению модулем можно по ссылкам [7] – [9].

## Лазерный дальномер

На рисунке 13 представлена плата лазерного дальномера на основе датчика VL53L0X. Это один из самых малогабаритных датчиков для измерения расстояния.

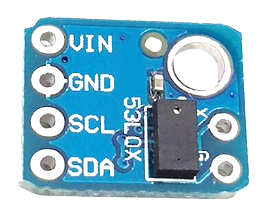


Рисунок 13 – Лазерный дальномер на основе VL53L0X

Принцип работы основан на отражении лазерного луча от препятствия. VL53L0X измеряет время, за которое лазерный луч достигнет объекта, отразится от него и вернётся в приёмник. Поскольку используется направленный источник света, он удобен для определения расстояния только от поверхности непосредственно перед ним.

В отличие от обычных датчиков дальности, VL53L0X обеспечивает точное измерение расстояния независимо от цвета и отражательной способности объекта, обеспечивая лучшую защиту от помех. Его основные технические параметры:

– рабочее напряжение – от 2,6 до 5 В;

– измеряемые расстояния – от 30 до 2000 мм;

– точность определения дальности – ± 5% (режим высокой скорости); ± 3% (режим высокой точности);

– минимальное время ранжирования – 20 мс (режим высокой скорости); 200 мс (режим высокой точности);

– угол обзора – 25 °;

– длина волны лазера – 940 нм;

– размер – 13,3x10,5 мм.

Датчик программно управляется от микропроцессора по интерфейсу I2C, имеет сигнал разрешения VL53L0X\_EN, который коммутирует цепь питания. Вывод GPIO – программируемый выход прерывания (логический уровень питания). Вывод XSHUT является активным-низким входом отключения, модуль подтягивает его до уровня напряжения питания (3,3 В), чтобы включить датчик по умолчанию. Низкий уровень этого вывода переводит датчик в аппаратный режим ожидания.

Ознакомится с ресурсами по программному управлению можно по ссылкам [10], [11].

## Лазерный целеуказатель

На рисунке 14 представлен модуль лазера KY-008. Модуль содержит лазерный светодиод с цилиндрическим радиатором и пассивные компоненты, обеспечивающие режим работы светодиода.

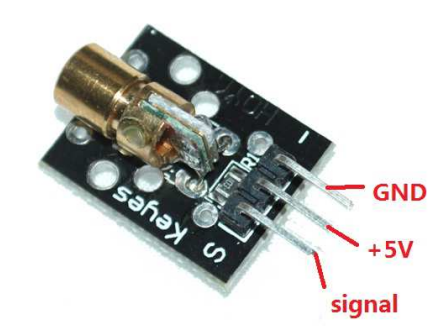


Рисунок 14 – Лазерный дальномер

Модуль лазера KY-008 создает небольшое световое пятно на противоположно расположенном объекте благодаря особым свойствам лазерного излучения. Луч виден в задымленном или пыльном помещении. Модуль лазера применяется как излучающий компонент в фотореле, в которых между источником света и фотоприемниках расстояние измеряется в метрах. Такая схема расположения фотоприборов используется при подсчете переместившихся крупных объектов между источником и приемником. В охранных системах это получило название световой барьер. Среди других применений – лазерная указка, установки световых эффектов, где световое пятно перемещается с помощью зеркала.

Основные технические характеристики KY-008:

– напряжение питания – 5 В, ток – 30 мА;

– длина волны излучения – 650 нм (оттенок света – рубиновый);

– светоотдача – 2–5 мВт;

– предельное расстояние до фотоприемника – 14 м.

Модуль лазера KY-008 чувствителен к статическому электричеству, даже импульсное превышение питающего или управляющего напряжений приводит к выходу из строя светодиода. При работе происходит нагрев светодиода, который в незначительной мере снижает имеющийся на светодиоде радиатор. Основной параметр питания лазерного светодиода – ток, а не напряжение. При нагреве снижается сопротивление светодиода, происходит увеличение потребляемого тока, в результате нагрев ускоряется и развивающийся процесс приводит к негативным последствиям.

В большинстве случаев не требуется работа модуля лазера KY-008 на предельной мощности. Снижать потребляемый ток можно с помощью последовательной установки в цепи питания резистора. Потребление максимального тока вызывает быстрый перегрев, работа в половинном режиме потребления замедляет процесс саморазогрева, но все же он будет происходить. Наиболее правильным способом обеспечения надежной работы является использование для питания модуля лазера KY-008 стабилизатора или ограничителя тока. В схеме подключения лазера используется стабилизированная цепь питания 3,3 В, источник которой располагается на плате модуля управляющего компьютера (микроконтроллера RP2040).

# Технология разработки программного обеспечения для микроконтроллера RP2040

На данный момент существуют следующие варианты для разработки программного обеспечения (ПО) для микроконтроллера RP2040:

– при помощи языков C, C++ и комплекта для разработки программного обеспечения «Raspberry Pi Pico SDK»;

– при помощи языка программирования MicroPython;

– при помощи языка программирования CircuitPython.

Варианты c использованием языков программирования, основанных на Python (CircuitPython, MicroPython), просты в применении и достаточны для быстрого прототипирования, но ограничены в скорости выполнения. C и С++, в свою очередь, обеспечивают наилучшую производительность, но требуют больших трудозатрат от разработчика ПО, по сравнению с языками на базе Python.

Микроконтроллер RP2040 имеет встроенную программу-загрузчик, которая позволяет загружать через интерфейс USB в энергонезависимую память микроконтроллера бинарный код прошивок в файлах формата UF2.

При первом подключении микроконтроллера RP2040 по интерфейсу USB к компьютеру разработчика Flash-память в RP2040 очищена, и программа-загрузчик активируется автоматически, предоставив компьютеру доступ к виртуальному USB-Flash накопителю для размещения в нем файлов прошивки. В случае же, когда прошивка уже записана во Flash-памяти, режим загрузчика можно активировать удержанием кнопки BOOTSEL на плате Raspberry Pi Pico одновременно с подключением платы по интерфейсу USB к компьютеру.

Для отладки программ можно подключиться к отладочному интерфейсу микроконтроллера Serial Wire Debug (SWD) при помощи внешнего аппаратного отладчика (например, SEGGER J-Link или Raspberry Pi Debug Probe).

В качестве более простого варианта для отладки можно использовать последовательный интерфейс UART, который выведен на отдельную колодку для подключения к внешним USB-UART адаптерам. Также, UART интерфейс может быть туннелирован в USB и использован без внешних устройств. В этом случае микроконтроллер PR2040 предоставляет компьютеру виртуальный последовательный порт через интерфейс USB.

## Разработка на языках программирования C и C++ при помощи Raspberry Pi Pico SDK

В случае разработки программного обеспечения при помощи комплекта для разработки Raspberry Pi Pico C/C++ SDK есть возможность использования аппаратных средств ммикроконтроллера более оптимальным образом.

Документация Raspberry Pi Pico по применению SDK может быть получена по ссылке [12].

Для операционной системы Linux процесс начальной подготовки к работе сводится к выполнению следующих команд:

1. Получение файлов Raspberry Pi Pico SDK посредством системы контроля версий Git:

git clone --recursive https://github.com/raspberrypi/pico-sdk.git

1. Получение файлов примеров через Git

git clone https://github.com/raspberrypi/pico-examples.git

1. Установка необходимых программ для сборки программного обеспечения  
   apt update && apt install cmake gcc-arm-none-eabi build-essential
2. Запись в переменную среды PICO\_SDK\_PATH пути к комплекту разработки ПО

export PICO\_SDK\_PATH=`pwd`/pico-sdk

1. Подготовка окружения к сборке исходного кода примеров

mkdir pico-examples/build && cd pico-examples/build && cmake ..

Пример кода прошивки для мигания светодиодом с использованием Rasperry Pi Pico SDK, расположенный в каталоге pico-examples/blink:

#include "pico/stdlib.h"

int main() {

// Конфигурация пина со светодиодом

const uint LED\_PIN = 25;

gpio\_init(LED\_PIN);

gpio\_set\_dir(LED\_PIN, GPIO\_OUT);

// рабочий бесконечный цикл

while (true) {

// Переключить светодиод

gpio\_put(LED\_PIN, 1);

sleep\_ms(250);

gpio\_put(LED\_PIN, 0);

sleep\_ms(250);

}

}

Для того чтобы выполнить сборку исходного кода, необходимо выполнить следующую команду:

cd pico-examples/build/blink && make -j4

В случае успешной сборки, будет создано два файла: blink.elf и blink.uf2. ELF файл может быть использован программным обеспечением для отладки (например, GDB, LLDB, SEGGER Ozone) или для прошивки микроконтроллера посредством внешнего аппаратного отладчика. UF2 файл используется для прошивки микроконтроллера посредством внутренней программы-загрузчика. Этот файл можно загрузить на RP2040 через виртуальный USB-Flash накопитель, доступ к которому возможен в случае, если при подключении микроконтроллера через интерфейс USB была нажата кнопка BOOTSEL на плате. После копирования .uf2 – файла на виртуальный USB-Flash накопитель, микроконтроллер RP2040 автоматически перезагрузится и начнет выполнение программы.

Для создания начального шаблона для своего проекта существует [генератор проектов от Raspberry Pi](https://github.com/raspberrypi/pico-project-generator) Foundation, доступный по ссылке [13]. Генератор создаёт шаблоны проектов для Rasperry Pi Pico SDK, Visual Studio Code, а также позволяет подключить библиотеки для работы с периферией.

## Работа в среде MicroPython

MicroPython и CircuitPython достаточно близки друг к другу, но с некоторыми особенностями для отладки и API.

Установка MicroPython предполагает следующие действия:

1) скачивание .uf2 – файла прошивки по ссылке [14];

2) перевод микроконтроллера RP2040 в режим загрузчика удержанием кнопки BOOTSEL во время подключения к компьютеру через интерфейс USB;

3) копирование .uf2 – файла из первого шага на виртуальный USB-Flash накопитель.

После чего микроконтроллер перезагружается и на компьютере появляется новый виртуальный последовательный порт, к которому можно подключиться посредством ПО для работы с последовательными портами (Putty, minicom, TeraTerm, XShell). После подключения пользователю становится доступна интерактивная консоль Python. Для работы с исходном кодом прошивок на языке MicroPython предполагается использование интегрированной среды разработки Thonny, которую можно скачать по ссылке [15].

После установки Thonny, в настройках необходимо выбрать целевовое устройство (Raspberry Pi Pico). Данная операция производится через меню Tools / Options, на вкладке «Interpreter», как показано на рисунке 15, в выпадающем списке «Which interpreter or device» выбирается «MicroPython (Raspberry Pi Pico)», а в поле «Port» выбирается виртуальный последовательный порт, который предоставлен микроконтроллером RP2040 после подключения его к ПК по интерфейсу USB.

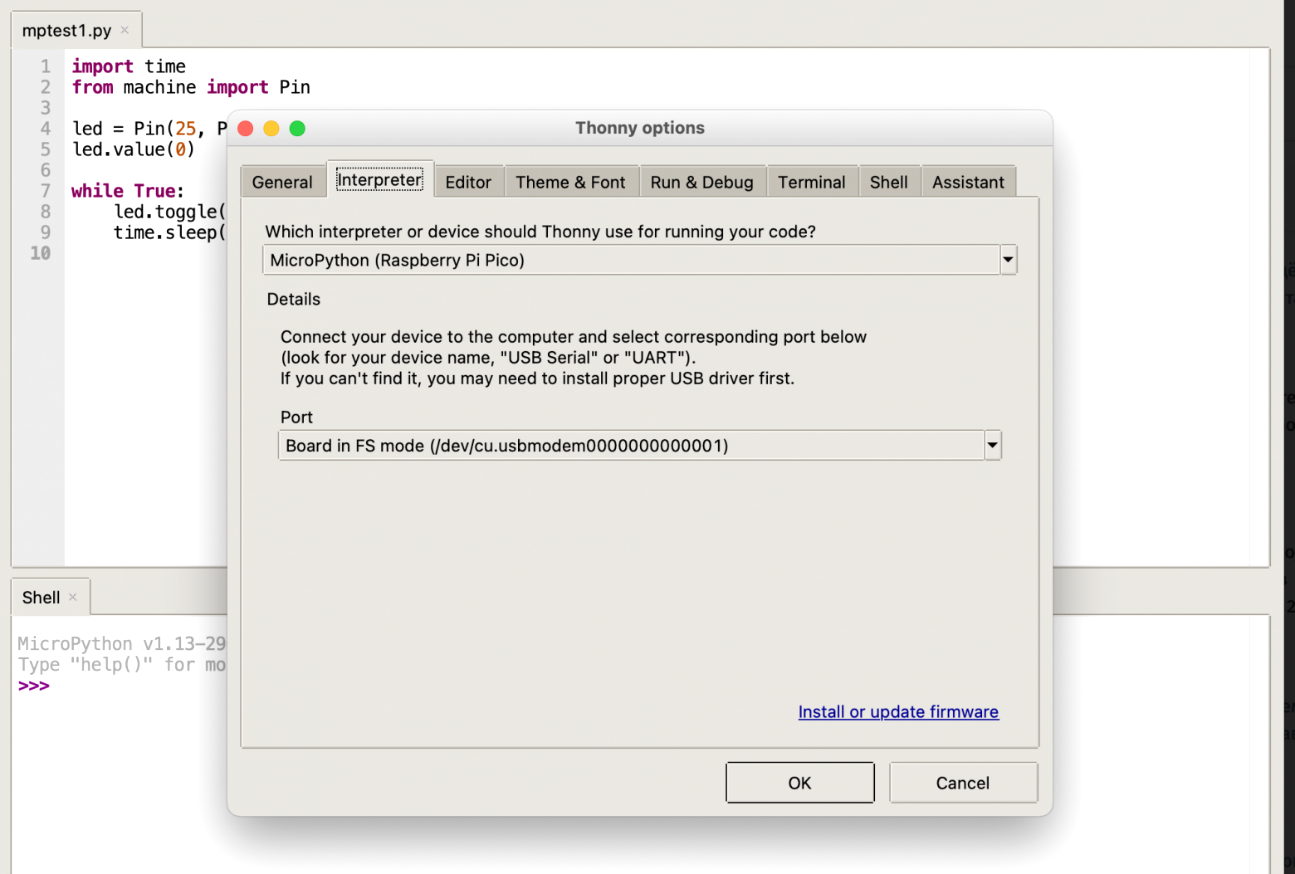


Рисунок 15 – Подключение к Raspberry Pi Pico в среде Thonny

Пример исходного кода для мигания светодиодом на языке программирования MicroPython:

# Импорт необходимых библиотек

import time

from machine import Pin

# Конфигурация пина со светодиодом как выходного и установка его в логический нуль

led = Pin(25, Pin.OUT)

led.value(0)

# Бесконечный цикл

while True:

led.toggle() # Переключить светодиод

time.sleep(1) # Подождать 1 секунду

## Работа в среде CircuitPython

CircuitPython является ответвлением MicroPython, основным отличием которого является создание виртуального USB флеш-накопителя со своей файловой системой, на котором можно редактировать скрипты, написанные на языке программирования CircuitPython, при помощи любого текстового редактора. При любой записи изменённого скрипта происходит автоматический перезапуск микроконтроллера и выполнение исходного кода.

Через виртуальный последовательный порт можно видеть результат выполнения или же запускать интерактивный режим для выполнения команд в нём, что также упрощает отладку в ПО.

Установка CircuitPython состоит из следующих шагов:

1) Получение .uf2 – файла прошивки с интерпретатором CircuitPython по ссылке [16];

2) Перевод микроконтроллера RP2040 в режим загрузчика удержанием кнопки BOOTSEL во время подключения к компьютеру через интерфейс USB;

3) Копирование файла, упомянутого в пункте 1) выше, на виртуальный USB-Flash накопитель, после чего микроконтроллер RP2040 перезапускается автоматически и начинает исполнение интерпретатора CircuitPython.

После перезагрузки микроконтроллера, в системе появится новый виртуальный USB-Flash накопитель CIRCUITPY. На этом накопителе размещается файл code.py, с которого начинается выполнение исходного кода.

Файл code.py можно изменять непосредственно на виртуальном накопителе. Также доступна консоль на виртуальном последовательном порту с параметрами подключения 115200 8n1. При подключении можно получить доступ к интерактивной консоли Python, а также к тексту, который был выведен из исходного кода через встроенную функцию языка «print».

Программа управления светодиодами на CircuitPyhon выглядит следующим образом:

# Импорт необходимых библиотек

import board

import time

from digitalio import DigitalInOut, Direction

# Конфигурация пина со светодиодом как выходного

led = DigitalInOut(board.LED)

led.direction = Direction.OUTPUT

# Бесконечный цикл

while True:

led.value = not led.value # Переключить светодиод

time.sleep(1) # Подождать 1 секунду

# Технология разработки ПО для микроконтроллера STM32F030F4P6

Разработка программ для микроконтроллера STM32F030F4P6 рассматривается на примере работы в среде IDE, разработанной производителем микросхемы STM32 (компанией STMicroelectronics) – STM32CubeIDE.

## Установка и запуск среды STM32CubeIDE

Скачать и установить среду STM32CubeIDE можно по ссылке [17].

После запуска среды появляется диалоговое окно, приведенное на рисунке 16, с предложением изменить или оставить как есть путь к рабочей папке среды.

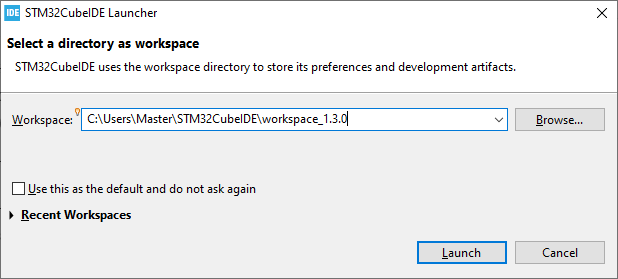


Рисунок 16 – Диалоговое окно выбора рабочей папки среды

Затем выбирается в меню пункт: File / New / STM32 Project. Появляется окно выбора микроконтроллера, как показано на рисунке 17, на котором планируется исполнение прошивки. В строке поиска вводится STM32F030F4P6 и в списке справа выбирается соответствующий вариант.

Далее на рисунке 18 показано как вводится имя проекта. По нажатию кнопки Next выполняется переход в окно первичных настроек проекта. Здесь нужно написать имя проекта в поле name. Нажатием кнопки Finish выполняется переход в главный интерфейс среды разработки. Изображение главного интерфейса среды разработки представлено на рисунке 19.

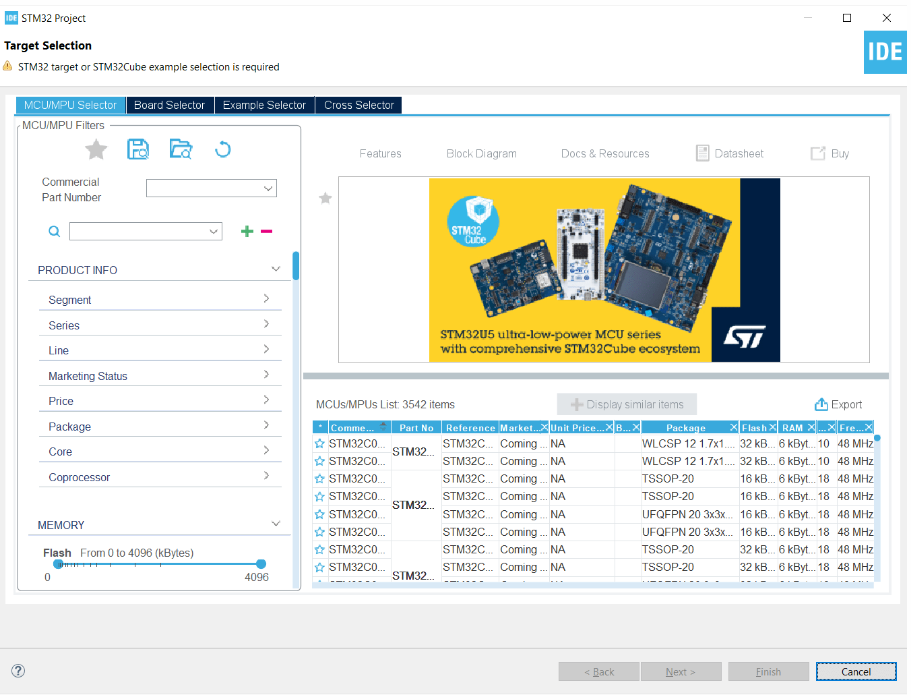


Рисунок 17 – Диалоговое окно выбора микроконтроллера

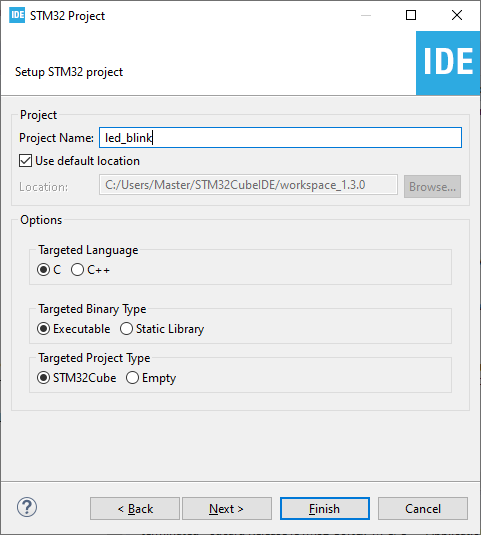
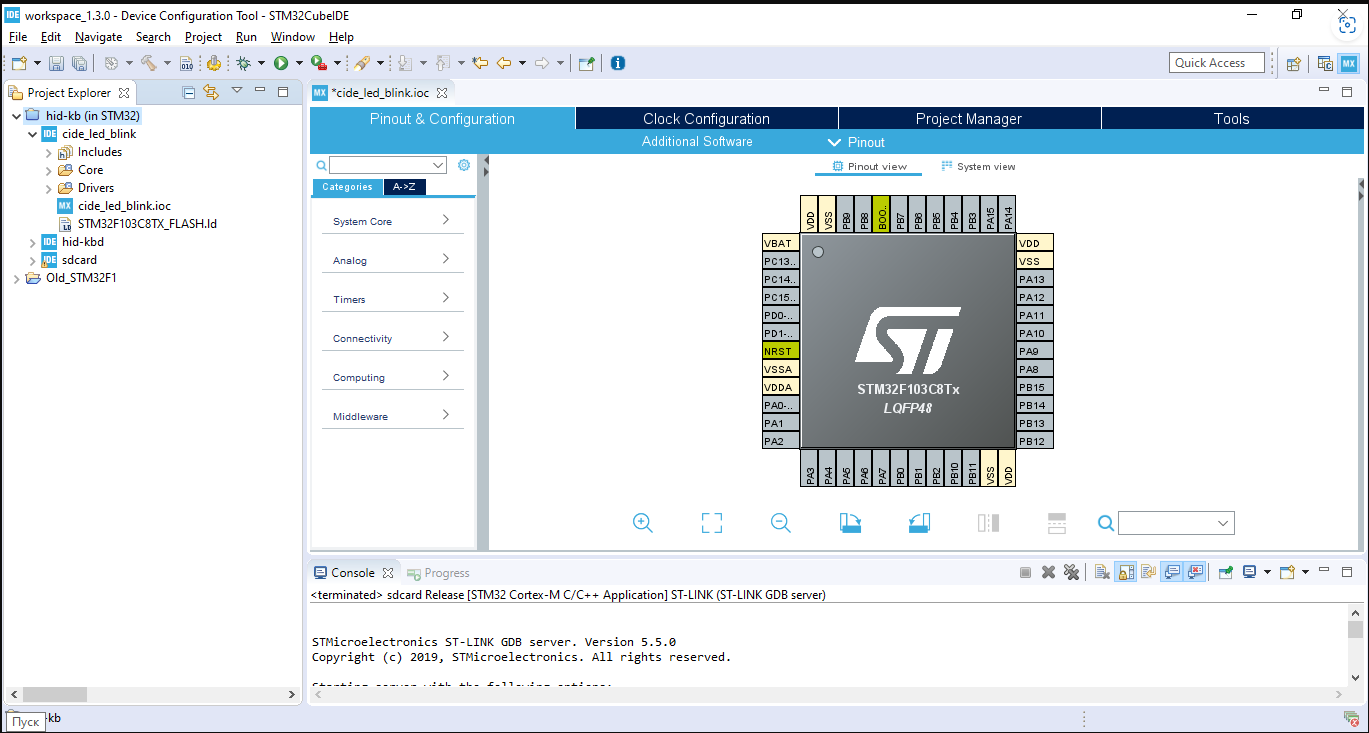


Рисунок 18 – Диалоговое окно первичных настроек проекта

Рисунок 19 – Диалоговое окно главного интерфейса среды разработки

Слева меню расположено дерево каталогов проекта. Оно дублирует реальные папки и файлы, которые находятся в рабочей папке проекта.

Справа – система вкладок, где по умолчанию выбрана вкладка конфигуратора. На этой вкладке изображён микроконтроллер со всеми своими ножками, которые мы можем настраивать с помощью мышки.

Сверху – панель иконок с самыми часто-используемыми функциями.

## Конфигуратор STM32CubeMX

Настройка входов/выходов, таймеров и прочей периферии является обязательной процедурой. Чтобы упросить процесс разработки, используется специальный инструмент для настройки, который называется STM32CubeMX. Данный инструмент интегрирован в STM32CubeIDE.

Контакты в STM32 нумеруются по схеме: P[порт][индекс]. Например, PA10 – это контакт с индексом 10 в порту A.

В качестве примера, рассмотрим процесс подключения к этому контакту светодиода. Вначале следует настроить в конфигураторе контакт PA10 на режим выхода.

Нажатием левой кнопки мыши на контакте открывается выпадающее меню с разными вариантами использования контакта. Изображение меню представлено на рисунке 20. Выбирается GPIO\_Output. Контакт подсветится зелёным и получит подпись GPIO\_Output, как показано на рисунке 21.

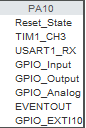
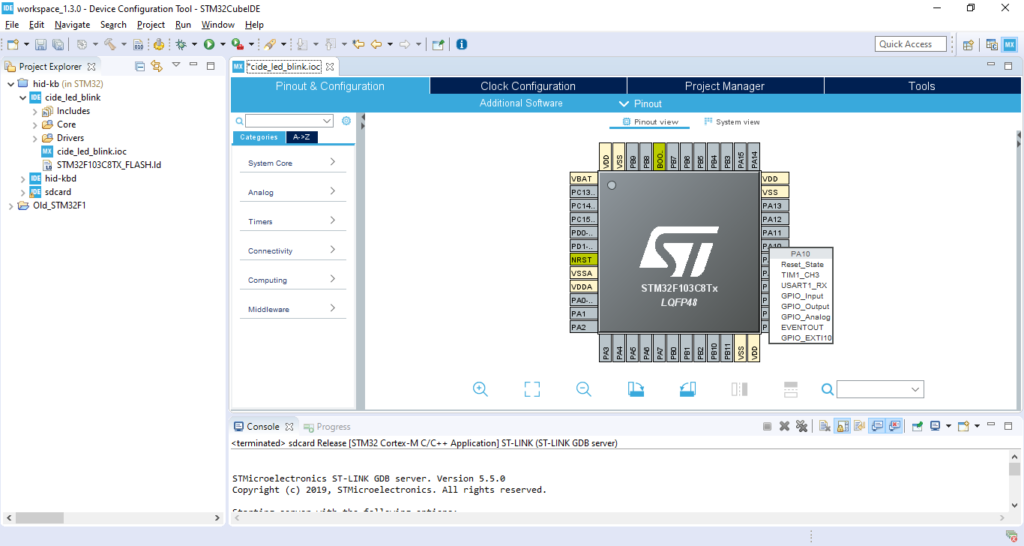


Рисунок 20 – Назначение функционального типа контакта

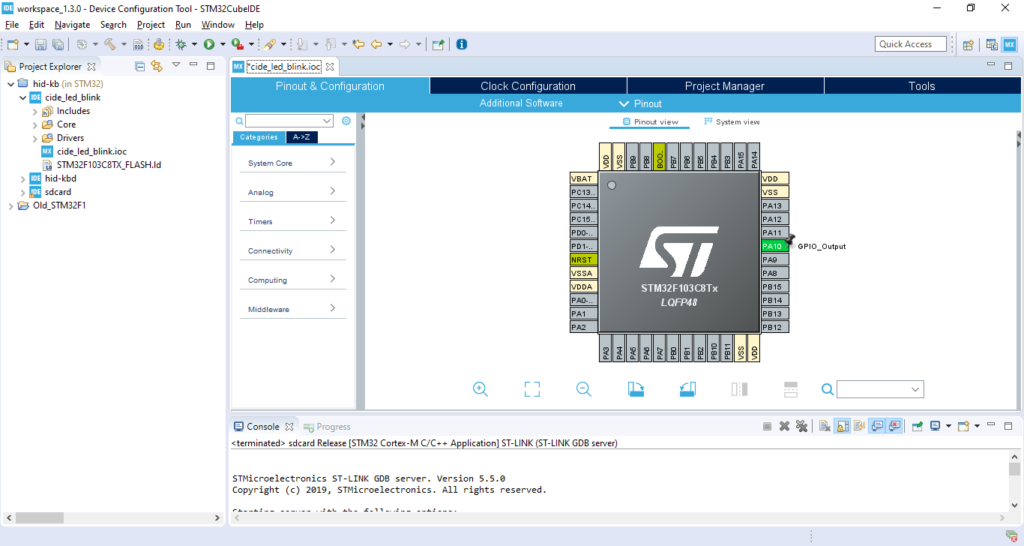


Рисунок 21 – Обозначение функционального типа контакта

Затем следует назначить псевдоним для этого контакта. На этой же вкладке, в правом меню, раскрывается пункт System core / GPIO. Изображение меню представлено на рисунке 22. Этот раздел меню отвечает за настройку параметров контактов общего назначения – GPIO, которые ранее были активированы. Для них можно назначать псевдонимы, прерывания, устанавливать подтяжки к питанию и земле, и другие настройки.

Далее выбираем контакт PA10. В поле User Label следует написать выбранный псевдоним, например слово LED. Псевдоним используется для удобства разработки – после его назначения в программе будет использовано мнемонически понятное слово LED.

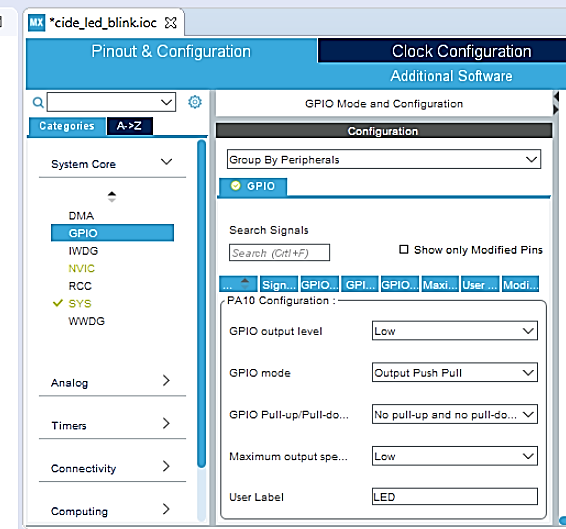
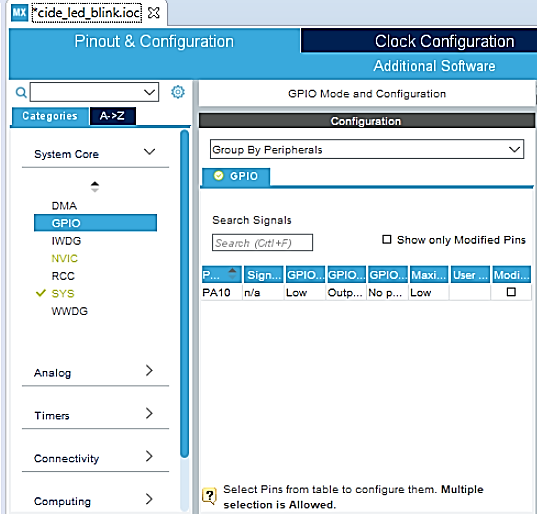


Рисунок 22 – Наименование контакта

## Настройка контактов программатора

Имеется ещё одна вспомогательная настройка, без использования которой, после первой прошивки программатором ST-Link, микроконтроллер может перестать отвечать по отладочному интерфейсу SWD. Это можно будет исправить, но лучше заранее сделать настройку.

Зайдя в раздел SYS, в поле Debug, следует выбрать Serial Wire, после чего контакты PA13 и PA14 будут зарезервированы за программатором и их случайное повторное использование в исходном коде будет затруднено. Изображение выбранных контактов представлено на рисунке 23

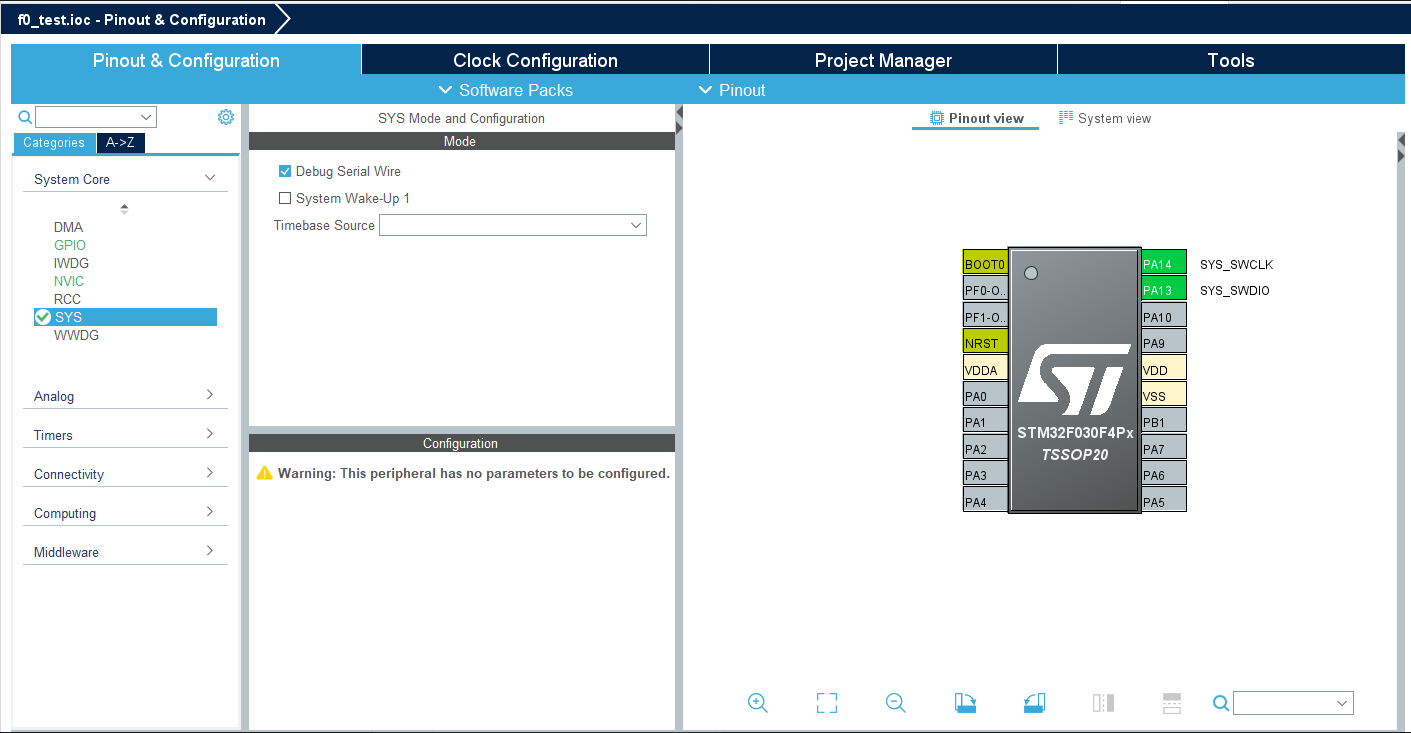


Рисунок 23 – Резервирование контактов для программатора

## Программа изменения состояния светодиода

Основной код программы начинает исполнение с файла main.c. При открытии файла справа появится вкладка с редактором кода выбранного файла Изображение окна редактора представлено на рисунке 24.

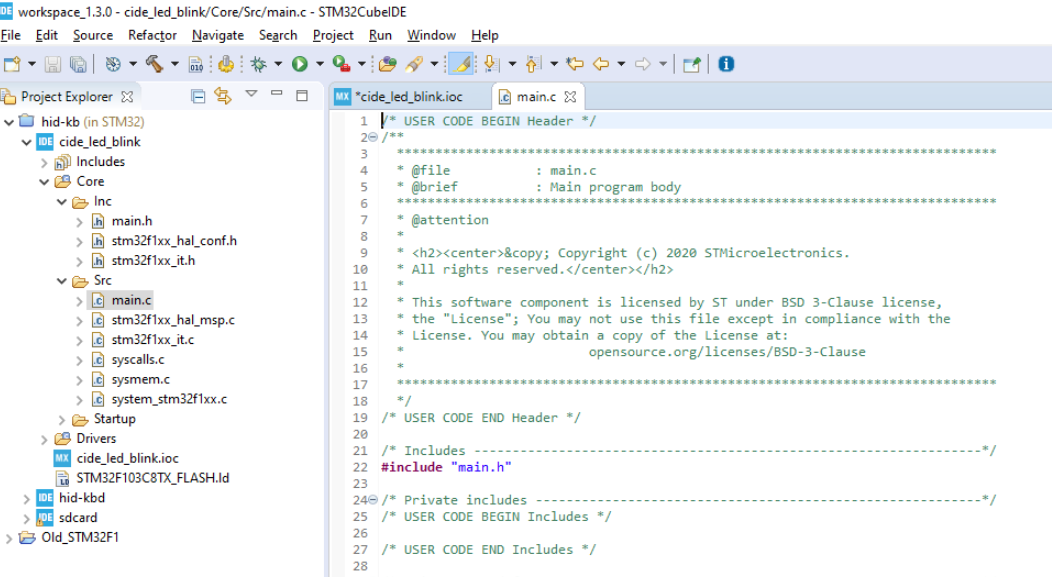


Рисунок 24 – Открытие вкладки с основным кодом программы

По умолчанию, код состоит из фрагментов, которые генерируются автоматически при любом изменении в конфигураторе, и фрагментов, куда разработчик может вставить свой код. Фрагменты разработчика помечены комментарием, начинающимся со слов «USER CODE». Любой текст, написанный за пределами блоков USER CODE, может быть удален средой разработки при изменении настроек проекта.

Главный цикл программы, выглядят следующим образом:

/\* USER CODE BEGIN 3 \*/

/\* USER CODE END 3 \*/

Для смены состояния контакта используется функция из библиотеки HAL:

HAL\_GPIO\_WritePin(порт, индекс, состояние)

Здесь порт – имя порта, которому принадлежит контакт. Поскольку был назначен псевдоним, имя будет LED\_GPIO\_Port. Затем идёт индекс контакта в порту LED\_Pin. Третий параметр – состояние: GPIO\_PIN\_SET – высокий уровень, GPIO\_PIN\_RESET – низкий уровень.

После включения и выключения светодиода следует добавить некоторое ожидание, для чего следует использовать функцию:

HAL\_Delay(время),

где время – количество миллисекунд, которые нужно подождать.

Таким образом, чтобы изменить состояние светодиода необходимо написать:

HAL\_GPIO\_WritePin(LED\_GPIO\_Port, LED\_Pin, GPIO\_PIN\_SET);

HAL\_Delay(50);

HAL\_GPIO\_WritePin(LED\_GPIO\_Port, LED\_Pin, GPIO\_PIN\_RESET);

HAL\_Delay(500);

В редакторе это будет выглядеть, как показано на рисунке 25.

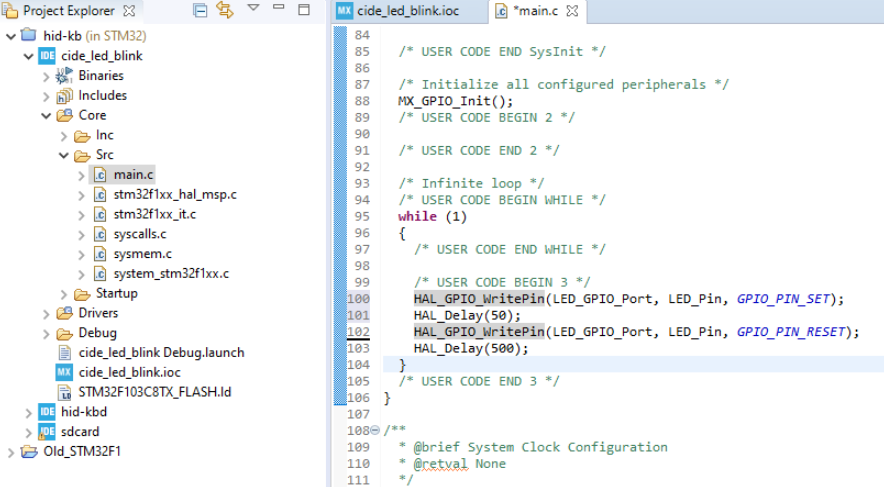


Рисунок 25 – Представление кода программы в окне редактора

## Компиляция программы

Чтобы собрать выполнить компиляцию и компоновку исходного кода прошивки, необходимо запустить процесс сборки. Для этого следует выбрать в меню пункт: Project / Build Project. В консоли в нижней части окна среды разработки отображаться процесс сборки, который завершается текстом:

22:05:23 Build Finished. 0 errors, 0 warnings. (took 25s.625ms)

Если рядом с errors стоит не 0, значит в процессе сборки произошла ошибка. До устранения всех ошибок результирующий бинарный файл прошивки не создается. Warnings – это предупреждения, на которые тоже следует обратить внимание, но они не препятствуют полной сборке программы.

## Загрузка программы

Для загрузки программы на микроконтроллер используется внешний аппаратный отладчик (программатор), который переносит прошивку в энергонезависимую память микроконтроллера, а также может осуществлять передачу отладочной информации на ПК. Внешний вид программатора ST-Link V2 mini представлен на рисунке 26.



Рисунок 26 – Модуль ST-Link V2 mini

Подключение ST-LINK к плате с микроконтроллером STM32 производится с помощью разъемного соединения.

У программатора имеется 10 контактов, из которых используется только 6. На плате рядом с микроконтроллером STM32 имеется шестиконтактный разъём со следующими наименованиями сигналов:

– GND – земля;

– SWCLK – тактовый сигнал интерфейса SWD;

– SWDIO – линия данных интерфейса SWD;

– NRST – линия перезагрузки микроконтроллера;

– 3.3 – питание 3,3 В.

Следует подключить соответствующие контакты ST-Link и платы микроконтроллера шлейфом проводов, а затем подключить программатор к USB-порту компьютера. Затем следует возвратиться к среде STM32CubeIDE и нажать зеленую стрелку на панели иконок, или выбрать в меню Run / Run.

В появившемся окне, изображение которого представлено на рисунке 27, можно настроить параметры отладки/загрузки. На вкладке «Debugger», как показано на рисунке 28, должен быть выбран программатор ST-Link и интерфейс SWD.

После чего следует нажать ОК и дождаться завершения процедуры.

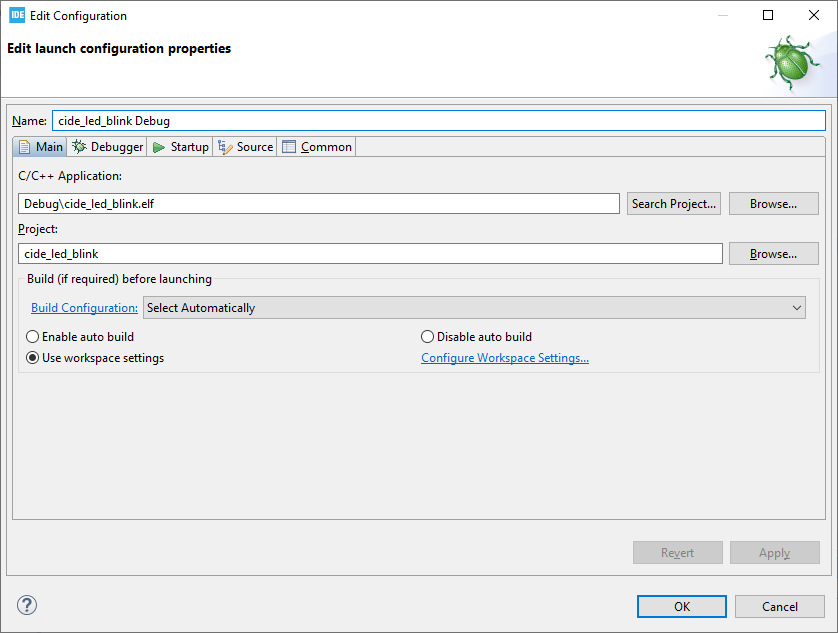


Рисунок 27 – Диалоговое окно настройки параметров отладки/загрузки

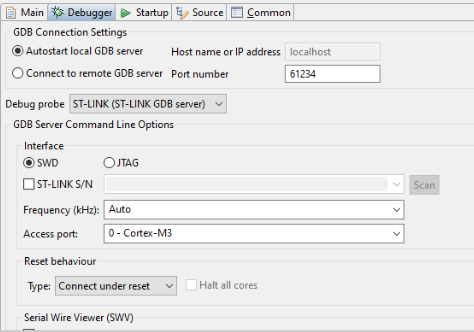


Рисунок 28 – Внешний вид вкладки Debugger

Ниже представлен листинг выполнения процедуры прошивки микроконтроллера:

STMicroelectronics ST-LINK GDB server. Version 5.5.0

Copyright (c) 2019, STMicroelectronics. All rights reserved.

Starting server with the following options:

Persistent Mode : Disabled

Logging Level : 1

Listen Port Number : 61234

Status Refresh Delay : 15s

Verbose Mode : Disabled

SWD Debug : Enabled

InitWhile : Enabled

Waiting for debugger connection…

Debugger connected

----------------------------------------------------------------

STM32CubeProgrammer v2.4.0

----------------------------------------------------------------

ST-LINK SN : 54FF6A064887535326211187

ST-LINK FW : V2J36S7

Voltage : 3.23V

SWD freq : 4000 KHz

Connect mode: Under Reset

Reset mode : Hardware reset

Device ID : 0x410

Device name : STM32F030 Medium-density

Flash size : 16 KBytes

Device type : MCU

Device CPU : Cortex-M0

Memory Programming …

Opening and parsing file: ST-LINK\_GDB\_server\_a02420.srec

File : ST-LINK\_GDB\_server\_a02420.srec

Size : 4624 Bytes

Address : 0x08000000

Erasing memory corresponding to segment 0:

Erasing internal memory sectors [0 4]

Download in Progress:

File download complete

Time elapsed during download operation: 00:00:01.109

Verifying …

Download verified successfully

После завершения процедуры прошивки микроконтроллер начинает исполнение и светодиод изменяет своё состояние с выключенного и включенное и обратно.

## Список использованных источников

1. https://docs.circuitpython.org/projects/ov7670/en/latest/index.html.
2. https:// [www.waveshare.com/wiki/MLX90640-D55\_Thermal\_Camera](http://www.waveshare.com/wiki/MLX90640-D55_Thermal_Camera).
3. <https://files.waveshare.com/upload/e/ef/UV_Sensor_%28B%29_User_Manual_EN.pdf>.
4. <https://github.com/enjoyneering/BH1750FVI>.
5. [www.silabs.com/sensors/optical/si114x/device.si1143-a11-gm?tab=softwareandtools](http://www.silabs.com/sensors/optical/si114x/device.si1143-a11-gm?tab=softwareandtools).
6. <http://3d-diy.ru/wiki/components/adresnye-svetodiody-ws2812b-ws2812s/>.
7. https://download.mikroe.com/documents/datasheets/NEO-M8-FW3\_DataSheet\_%28UBX-15031086%29.pdf.
8. https://github.com/SimpleMethod/STM32-GNSS.
9. <https://github.com/mikalhart/TinyGPSPlus/tree/master>.
10. <https://www.st.com/resource/en/datasheet/vl53l0x.pdf>.
11. <https://robotchip.ru/obzor-lazernogo-dalnomera-vl53l0x/>
12. <https://datasheets.raspberrypi.org/pico/getting_started_with_pico.pdf>.
13. <https://github.com/raspberrypi/pico-project-generator>.
14. <https://micropython.org/download/RPI_PICO/>.
15. <https://thonny.org/>
16. <https://circuitpython.org/board/raspberry_pi_pico/.\>
17. https://www.st.com/en/development-tools/stm32cubeide.html