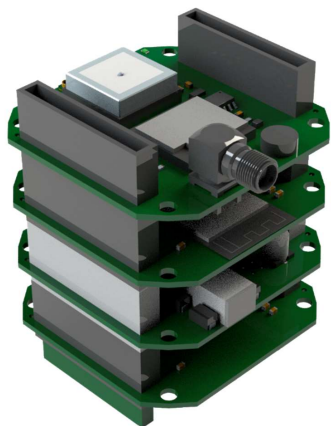


RaCEboard - электроника для твоего проекта



Это электронный мозг вашего устройства, будь то ровер, робот-манипулятор, модель ракеты или самолета. Это плата с кучей датчиков, которая отслеживает состояние устройства и реализует алгоритм, который вы запрограммировали.

Универсальный контроллер для практического обучения основам электроники, электротехники и программирования в рамках проектной деятельности.

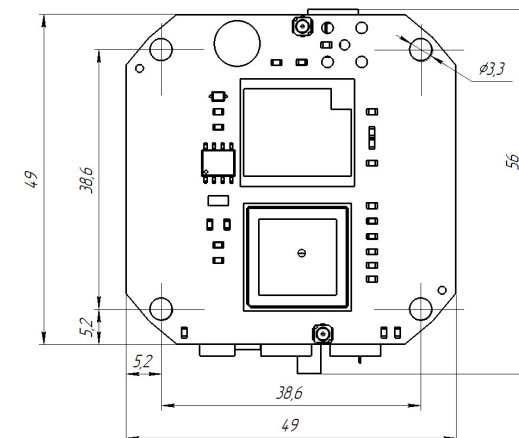
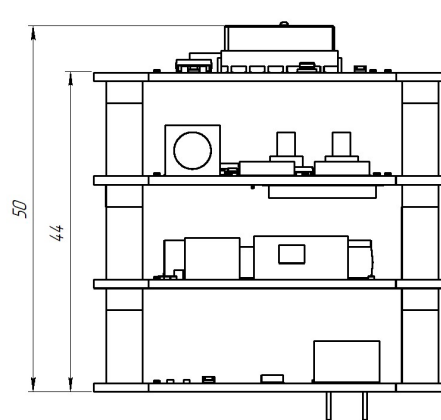
Программирование производится на языке Python.

Что умеет RaCEboard?

- определять свое местоположение по GNSS;
- определять ориентацию по инерциальным датчикам;
- определять свою высоту относительно уровня моря;
- управлять 2 сервоприводами и двигателем постоянного тока;
- передавать данные по воздуху до 5 км, используя радиомодуль;
- подзаряжать встроенную батарею;
- издавать звуковые сигналы;
- работать с внешними устройствами и датчиками;
- сохранять данные на внешнюю карту памяти.

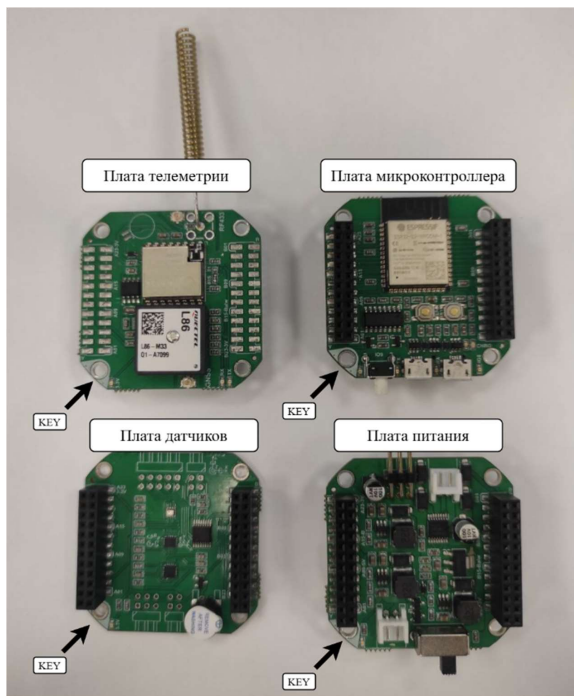
Заложена возможность использования Wi-Fi и Bluetooth.

Технические характеристики



1. Масса 74 гр.
2. Размер 49x49x50 мм.
3. MCU: ESP32-S3.
4. Барометр: BMP280.
5. Акселерометр: LSM6DSL.
6. Гироскоп: LSM6DSL.
7. Поддержка карты памяти до 64Гб.
8. GNSS: L86-M33.
9. Радиопередача: Ra01S.
10. Мощность радиопередачи 20dBm.
11. Частота радиопередачи: 433МГц.
12. Доступные интерфейсы: I2C, SPI, GPIO.
13. Рабочее напряжение 5V.
14. Совместимость с АКБ Li-Po/Li-Ion 1s.
15. Прошивка через microUSB – выполняется в виде загрузки файлов на внутренний накопитель.
16. Поддержка зарядки через microUSB.

Состав RaCEboard и порядок сборки



RaCEboard состоит из 4 плат, которые собираются как "бутерброд". Вы легко можете поменять платы местами или исключить одну или несколько из сборки - на работоспособность это не повлияет!

Плата телеметрии содержит GNSS, радиомодуль, их антенны и часы реального времени.

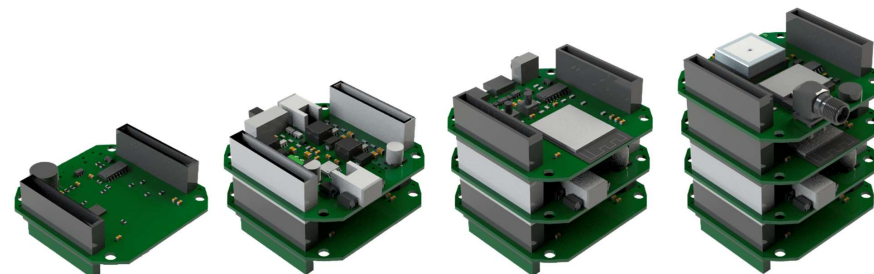
Плата микроконтроллера размещает МК ESP32-S3, карту памяти, кнопку, два разъёма microUSB для прошивки.

Плата питания имеет два канала управления сервоприводами, канал управления двигателем постоянного тока, кнопку питания и разъем для подключения аккумулятора.

Плата датчиков содержит барометр, акселерометр, гироскоп, магнетометр, пищалку, линию тих и выходы цифровых интерфейсов I2C и SPI.

Основные правила сборки

Мы рекомендуем собирать в следующей последовательности снизу-вверх:



плата датчиков – плата питания – плата микроконтроллера – плата телеметрии

- При сборке ориентируйтесь на ключ, он должен совпасть на всех 4 платах.
- Плату телеметрии рекомендуется размещать так, чтобы над антенной ничего не находилось.
- Плата питания устанавливается рядом с платой микроконтроллера.

Примеры работы с периферией

Запрограммировать RaCEboard очень легко с использованием Visual Studio Code.

Для этого необходимо предварительно установить [Python 3.10](#), а также дополнения в самом [Visual Studio Code](#): Python, Pylance и CircuitPython.

Для эффективной работы с периферией RaCEboard представлены следующие скетчи:

BMP280_work.py – пример работы барометра BMP280, позволяет получать данные:

- температура окружающей среды, $^{\circ}\text{C}$;
- атмосферное давление, кПа ;
- высота над уровнем моря в м .

Button_LED.py – пример работы светодиода, который загорается от нажатия кнопки.

Buzzer_work.py – пример работы пищалки, представлены функции:

- `BuzzerOn()` – включает пищалку с частотой `frequency_buzzer`;
- `BuzzerOff()` – выключает пищалку.

DCMotor_work.py – пример работы двигателя постоянного тока, представлены функции:

- `DCMotorSetSpeed(speed)` – функция задания скорости в процентах от -100 % до +100%.

GNSS_work.py – пример работы GNSS-модуля, позволяет получить данные:

- широта, градусы и минуты;
- долгота, градусы и минуты;
- высота, м ;
- скорость передвижения, узлы;
- азимут, градусы;
- расхождение по горизонтали.

Обязательное условие: необходимо как можно чаще вызывать `gnss.update()` в цикле `loop`, т.е. с частотой вызовов не менее 5-10 кГц.

LED_blink.py – пример мигания светодиодом.

LSM6DSL_work.py – пример работы акселерометра-гироскопа *LSM6DSL*, позволяет получить данные:

- линейные ускорения по трём осям (X, Y, Z), м/с^2 ;
- угловые скорости по трём осям (X, Y, Z), град/с^2 ;
- температура датчика, $^{\circ}\text{C}$.

MUX_work.py – пример работы аналогового переключателя 1 к 8 (1 двунаправленный вход с одной стороны, и 8 с другой стороны).

Линия 0 позволяет измерить напряжение аккумулятора;

Линии 1, 2 имеют подтяжку на *GND* 10 кОм ;

Линии 3-7 имеют состояние *float*.

Представлены следующие функции:

- `muxSelectLine(line)` – позволяет присоединить пин микроконтроллера к выбранной линии (0-7);
- `mux2In()` – позволяет установить, что пин является цифровым входом;
- `mux2Analog()` – позволяет установить, что пин является аналоговым входом;
- `mux2Out()` – позволяет установить, что пин является цифровым выходом.

PCF8563_work.py – пример работы часов реального времени, представлены следующие функции:

- `PCF8563.setDate(YYYY,MM,DD)` – задать начальную дату;
- `PCF8563.setTime(HH,MM,SS)` – задать начальное время;
- `PCF8563.getDate(YYYY,MM,DD)` – получить текущую дату;
- `PCF8563.getTime(HH,MM,SS)` – получить текущую время.

QMC5883L_work.py – пример работы цифрового магнетометра QMC5883L, представлены следующие функции:

`qmc5883l.magnetometer()` – получить значение индукции магнитного поля в мТл .

Ra01S_TX_work.py – пример работы радиопередачи с помощью радиомодуля *Ra01S*, представлены следующие функции:

`Ra01S.on()` – включение радио;

Ra01S.SetLowPower() – задание минимальной мощности радиопередатчика для отладки внутри помещения;

Ra01S.SetMaxPower() – задание максимальной мощности радиопередатчика для полёта;

Ra01S.SetChannel(channel) – задание канала (0-6);

Ra01S.SendS(string) – отправка строки по радиоканалу.

Обязательное условие: Чтобы использовать радио, необходимо инициализировать CS-пин карты памяти – даже если не планируется её использование.

Ra01S_RX_work.py – скетч для проверки приёма радио.

Обязательное условие: перед проверкой радиопередачи, в данном скетче необходимо назначить свой канал.

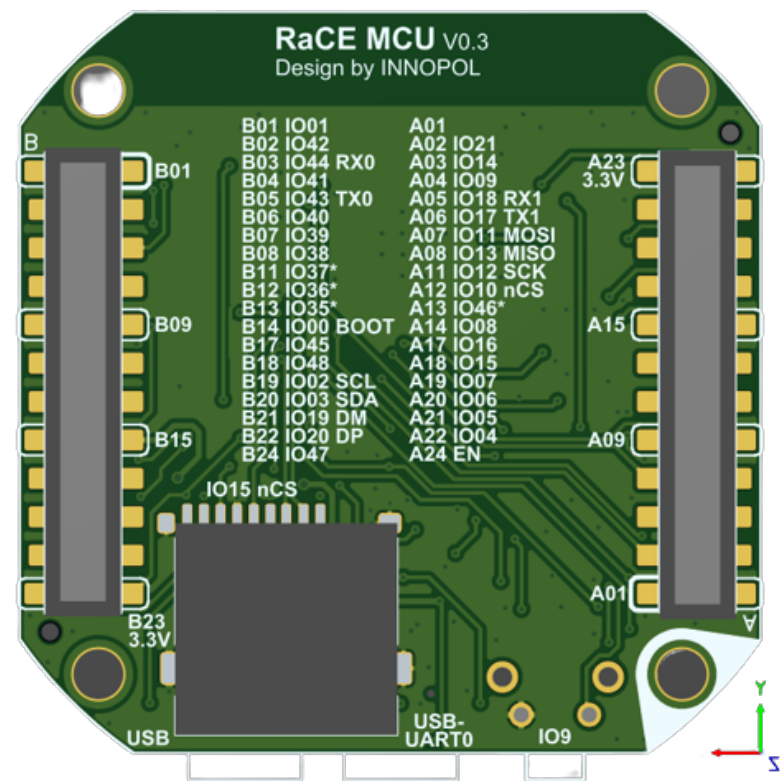
SD_files.py – пример работы SD-карты в режиме чтения и записи файлов. После успешного монтирования карты памяти, работа с файлами идёт, как на стандартном *Python*.

SD_folders.py - пример работы SD-карты в режиме чтения и записи файлов. Показывает, как создавать и удалять папки с файлами. Поддерживается только один уровень вложенности – нельзя удалить папку в папке.

Servo_work.py – пример работы сервопривода, представлены следующие функции:

- *ServoSetAngle(servo, angle)* – функция задания угла поворота сервы (от 0 до 180 градусов).

RaCEboard Pinout



Распиновка межплатного коннектора

Название устройства	Пин устройства	Пин микроконтроллера
Светодиод	-	IO8
Кнопка	-	IO9
Бuzzer	-	IO4
ШИМ для первого сервопривода	-	IO48
ШИМ для второго сервопривода	-	IO45
ШИМ для DC мотора	-	IO21 & IO47
BMP280	SDA	IO3
	SCL	IO2
LSM6DSL	SDA	IO3
	SCL	IO2
QMC5883L	SDA	IO3
	SCL	IO2
PCF8563	SDA	IO3
	SCL	IO2
GNSS	TX	IO17
	RX	IO18
SDCard	MOSI	IO11
	MISO	IO13
	SCK	IO12
	CS	IO15
Ra01SC	MOSI	IO11
	MISO	IO13
	SCK	IO12
	CS	IO7
	nReset	IO6
	nInt	IO5
MUX	MUX_OUT	IO1
	MUX_S0	IO46
	MUX_S1	IO0
	MUX_S2	IO39