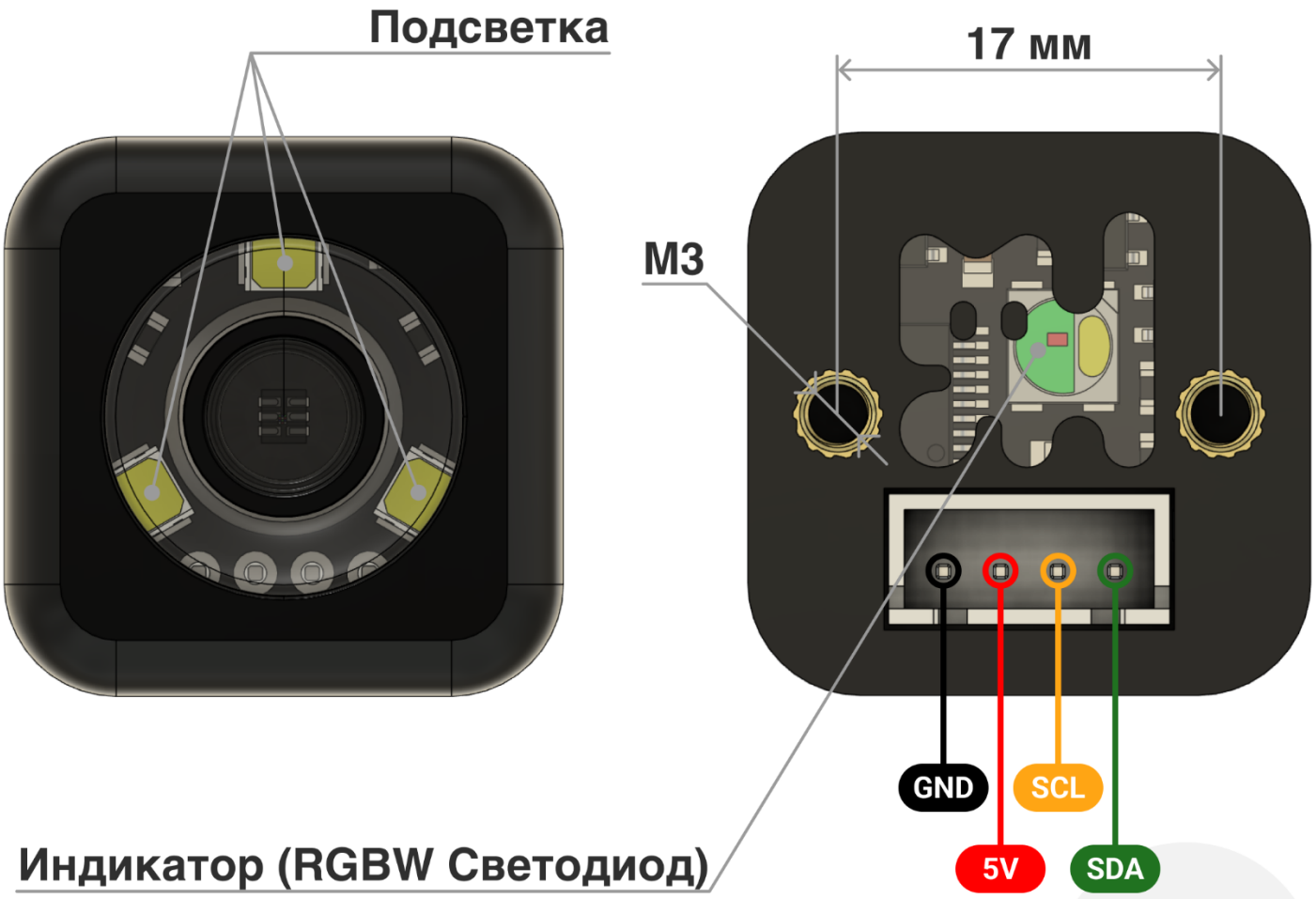


Датчик цвета Баригадам — это интеллектуальный цветовой модуль с интерфейсом I²C, предназначенный для робототехники, обучения и встраиваемых проектов, где критически важно надёжное распознавание цветов. Внутри используется качественное цветочувствительное ядро, работающее совместно с выделенным встроенным контроллером и интегрированным светодиодом RGBW. Датчик автоматически измеряет каналы RGBC (красный, зелёный, синий и Clear), фильтрует шум, компенсирует внешнее освещение в основных режимах и предоставляет как сырые данные, так и готовый к использованию индекс цвета (шесть базовых цветов плюс «пусто»).

Модуль поддерживает шесть режимов работы, включая стандартное распознавание 6 цветов с LED-индикацией, режим «Истинный цвет», в котором светодиод отображает фактически определённый цвет, а также высокоскоростные режимы сырых данных до 400 Гц для продвинутых сценариев использования. В фильтрованных режимах показания остаются стабильными и в значительной степени не зависят от внешнего освещения, что делает датчик отличным решением для соревновательных роботов.

© 2026, ООО «БАРИГАДАМ». Все права защищены.

Содержание

1. Обзор 2
2. Быстрый старт 2
   2.1. Что понадобится 2
   2.2. Питание и логические уровни 2
   2.3. Типичное подключение 3
   2.4. Настройки по умолчанию 3
3. I²C Интерфейс 4
   3.1. Адрес 4
   3.2. Протокол доступа к регистрам 4
4. Карта регистров 5
5. Значение цвета (регистр COLOR) 6
   5.1. Значения 6
   5.2. Режимы 6
6. Работа с режимами 6
   6.1. Сводка 6
   6.2. Подробности 7
7. Изменение I²C адреса и режима 8
   7.1. Изменение I²C адреса 9
   7.2. Изменение режима 9
8. Чтение данных в коде 10
   8.1. Стандартное чтение по регистрам 10
   8.2. Простое чтение первых N байт 10
9. Рекомендации по использованию 11
10. Arduino библиотека: Barigadam_ColorSensor 11
11. Важная информация и уведомление 14

© 2026, ООО «БАРИГАДАМ». Все права защищены.

1. Обзор

Датчик цвета Баригадам – это умный модуль:

- ✓ Встроенный светодиод RGBW для индикации цвета объекта
- ✓ Обработанные значения RGBC (красный / зелёный / синий / Clear)
- ✓ Встроенная классификация цвета (6 основных цветов + “пустота”)
- ✓ I²C интерфейс с регистровой картой (13 регистров)
- ✓ Настраиваемый I²C-адрес и режим работы (хранятся во flash-памяти и сохраняются после выключения)

Датчик сделан, чтобы **быть легким даже для начинающих**.
Вам только потребуется:

- Контроллер с I²C (EV3, Arduino, STM32, ESP32, Raspberry Pi и т. д.)
- 4 провода: **GND, VCC, SCL, SDA**

2. Быстрый старт

2.1. Что понадобится

- Датчик цвета Баригадам
- Любой контроллер с I²C:
  - Ардуино Уно / Мега / Нано
  - EV3
  - ESP32 / ESP8266
  - Платы STM32
  - Raspberry Pi и т. д.
- 4 провода (мама-папа или папа-папа, зависит от платы)

2.2. Питание и логические уровни

- **Напряжение питания:** 5 В рекомендовано.
- **I²C линии (SDA, SCL):** логический уровень соответствует напряжению питания.
  • 5 В питание → логика I²C 5 В.
  • 3.3 В питание → I²C будет 3.3 В (в таком случае яркость подсветки снизится).
- **Примечание:** При работе от **3.3 В** встроенные светодиоды подсветки будут светить тусклее, что снижает чувствительность датчика и общую точность измерений.
- **Скорость I²C: работает как на 100 кГц, так и на 400 кГц.**
- **Встроена защита от переполюсовки,** предотвращающая повреждение устройства при неправильном подключении питания**.**

© 2026, ООО «БАРИГАДАМ». Все права защищены.

2.3. Типичное подключение

Типичный 4-контактный разъём *(*точный порядок контактов смотрите на маркировке вашей платы*)*:

Пин датчика	Подключение к плате микроконтроллера
GND	GND
VCC	5 В (или 3,3 В, но подсветка будет тусклее)
SCL	пин I²C SCL
SDA	пин I²C SDA

Примеры:

- **Ардуино Уно**
  - SDA → A4
  - SCL → A5
- **Ардуино Мега**
  - SDA → D20
  - SCL → D21
- **ESP32**
  - Любые I²C-совместимые GPIO, настраиваемые в прошивке.

2.4. Значения по умолчанию

- **7-битный I²C-адрес: 0x40**
- **Режим:** 1 (“6 цветов + индикация”)
- **Частота обновления данных на выходе: ~120 Гц**

Сразу после подключения вы можете:

1. Просканировать I²C шину → ваше устройство будет найдено по адресу 0x40.
2. Считать измеряемые регистры → получите RGBC + значение цвета.
3. Индикатор на датчике будет показывать цвета в режиме 1.

© 2026, ООО «БАРИГАДАМ». Все права защищены.

3. I²C-интерфейс

3.1. Адрес

- Датчик использует **7-битные I²C адреса**.
- Адрес по умолчанию (из коробки): **0x40**
- Вы можете менять адрес на любое значение из диапазона **0x40-0x60** (33 возможных 7-битных адреса, смотрите раздел 5).

В библиотеке Wire для Arduino всегда используется 7-битный адрес напрямую (например, Wire.beginTransmission(0x40);).

3.2. Протокол доступа к регистрам

Датчик ведет себя как маленькое устройство I²C на основе регистров.

Чтобы считать регистры:

1. **Укажите регистр**
   - Начните запись на адрес устройства
   - Отправьте 1 байт**:** адрес регистра (**0x00–0x0C**)
   - Остановите передачу (или начните снова)
2. **Считайте значение**
   - Начните чтение с того же адреса устройства
   - Считайте 1 или несколько байт
   - Внутренний указатель регистра автоматически увеличивается после каждого байта
   - После последнего регистра (0x0C) указатель переходит обратно к 0x00

Это позволяет считать все данные измерения за один приём.

Пример шаблона:

- Запись: [DeviceAddr + W] [0x00] → устанавливает указатель на первый регистр
- Чтение 8 байт: [R_L, R_H, G_L, G_H, B_L, B_H, C_L, C_H]

Важно: Регистры датчика доступны только для чтения. Обычная запись используется только для установки указателя регистра (кроме специальной конфигурации, описанной в разделе 5).

© 2026, ООО «БАРИГАДАМ». Все права защищены.

4. Карта регистров

Датчик цвета Баригадам включает 13 регистров (0x00–0x0C):

Адрес	Имя	Тип	Описание
0x00	R_L	uint8_t	Красный канал, младший байт
0x01	R_H	uint8_t	Красный канал, старший байт
0x02	G_L	uint8_t	Зелёный канал, младший байт
0x03	G_H	uint8_t	Зелёный канал, старший байт
0x04	B_L	uint8_t	Синий канал, младший байт
0x05	B_H	uint8_t	Синий канал, старший байт
0x06	C_L	uint8_t	Clear-канал, младший байт
0x07	C_H	uint8_t	Clear-канал, старший байт
0x08	COLOR	uint8_t	Значение цвета (см. раздел 5.1)
0x09	MODE	uint8_t	Текущий режим (0–5, см. раздел 6)
0x0A	STATUS	uint8_t	Статус (0xAC в текущей прошивке датчика)
0x0B	DEV_ID	uint8_t	ID устройства (0xBA для датчика цвета Баригадам)
0x0C	FW	uint8_t	Версия прошивки (0x01)

- Каналы RGBC являются 16-битными и передаются в формате little-endian:
  - Красный канал = R_L + (R_H << 8)
  - Зелёный канал = G_L + (G_H << 8)
  - Синий канал = B_L + (B_H << 8)
  - Clear-канал = C_L + (C_H << 8)
- STATUS, DEV_ID и FW могут быть использованы для:
  - Проверки, что датчик подключен и отвечает,
  - Проверить совместимость прошивки с вашей версией библиотеки.

© 2026, ООО «БАРИГАДАМ». Все права защищены.

5. Значение цвета (COLOR регистр)

Прошивка датчика выполняет распознавание цвета внутри самого модуля.
Результат выдаётся в виде **значения цвета** в регистре **0x08 (COLOR)**.

5.1. Значения

Значение COLOR	**Что означает**
0xFF	**Пустота** / цвет не обнаружен
0x01	Чёрный
0x02	Синий
0x03	Зелёный
0x04	Жёлтый
0x05	Красный
0x06	Белый

5.2. Режимы

- В **режимах 0, 1, 2, 3**, COLOR обновляется на ~120 Гц.
- В режимах 4 и 5 датчик работает в режиме сырых данных, а COLOR всегда равен 0xFF (цвет в этих режимах не определяется).

6. Работа с режимами

Датчик цвета Баригадам поддерживает 6 режимов (mode = 0…5).
Текущий режим сохраняется в памяти датчика и может быть считан из регистра MODE (0x09).

6.1. Сводка

Режим	**Имя**	**Поведение подсветки**	**Частота обновления данных**	Значения RGBC	Регистр COLOR
0	Без индикации	Индикатор (светодиод RGBW) выключен	~120 Гц	Отфильтрованные, не зависят от фона	6 цветов + пустота
1	Основной (6 цветов + индикация)	Индикатор показывает обнаруженный цвет / «пусто»	~120 Гц	Отфильтрованные, не зависят от фона	6 цветов + пустота
2	Только цветные (без чёрного и белого)	Индикатор показывает только цветные объекты (красный/зелёный/синий/жёлтый), без чёрного/белого	~120 Гц	Отфильтрованные, не зависят от фона	{красный, синий, зелёный, жёлтый}
3	Истинный цвет	Индикатор показывает истинный цвет объекта	~120 Гц	Отфильтрованные, не зависят от фона	6 цветов + пустота

© 2026, ООО «БАРИГАДАМ». Все права защищены.

4	Супербыстрые значения	Индикатор показывает «огненный» цвет (оранжевый)	~400 Гц	Сырые RGBC	Всегда 0xFF (цвета нет)
5	Сырой фон	Индикатор фиолетовый, измерительная подсветка отключена	~400 Гц	Сырые RGBC (только фон)	Всегда 0xFF (цвета нет)

`6.2. Подробности

Режимы 0–3: отфильтрованные значения, не зависящие от внешней засветки/фона

Значения RGBC фильтруются и не зависят от внешнего освещения.

Это идеально подходит для:

- роботов, следующих по цветным меткам,
- соревновательных полей с нестабильным освещением,
- надёжного распознавания напечатанных цветных участков.

Режим 0 – Без индикации

- Такой же алгоритм распознавания и фильтрации, как в режиме 1,
- Но встроенный светодиод **RGBW** выключен.
- Используется, когда не нужна никакая видимая индикация от датчика.

Режим 1 – Основной режим

- Стандартный режим, рекомендуемый для большинства пользователей.
- Светодиод RGBW показывает, какой цвет обнаружен:
  - Красная поверхность → индикатор горит красным
  - Зеленая поверхность → индикатор горит зеленым
  - Синяя поверхность → индикатор горит синим
  - Желтая поверхность → индикатор горит желтым
  - Белая поверхность → индикатор горит белым
  - Черная поверхность / пустота → индикатор горит тускло / индикатор не горит
- COLOR = один из {пусто, чёрный, синий, зелёный, жёлтый, красный, белый}.

Режим 2 – Только цветные (без черного/белого)

- Специальный режим для задач, где требуются только цветные объекты.
- Черный и белый цвета не распознаются.
- COLOR может принимать значения:
  - пусто (0xFF),
  - синий (0x02),
  - зелёный (0x03),
  - жёлтый (0x04),
  - красный (0x05).
- Светодиод показывает только эти цвета; чёрный и белый не отображаются.

© 2026, ООО «БАРИГАДАМ». Все права защищены.

Режим 3 — Истинный цвет

- Встроенный светодиод стремится воспроизвести реальный цвет объекта.
- Этот режим удобен для визуальной отладки:
  - Поместите объект под датчик,
  - светодиод отобразит цвет, близкий к тому, который «видит» датчик.

При этом регистр COLOR по-прежнему выдаёт классифицированный цвет (6 цветов + «пусто»).

Режимы 4–5: высокоскоростные режимы сырых данных

Это расширенные режимы для пользователей, которым требуется реализовать собственную обработку данных.

- Частота дискретизации: ~400 Гц (≈ 2,5 мс на выборку).
- Вычитание фона и глубокая фильтрация не выполняются.
- COLOR всегда равен 0xFF.

Режим 4 — сверхбыстрое сканирование

- Светодиод горит ярким «огненным»/оранжевым цветом.
- Регистры RGBC содержат сырые данные, полученные при такой подсветке.
- Используйте этот режим, когда требуется максимальная скорость и вы готовы самостоятельно обрабатывать сырые данные.

Режим 5 — сырой фон

- Светодиод показывает фиолетовый цвет, но подсветка, используемая для измерения, выключена
  (фактически это режим измерения фоновой/внешней освещённости).
- Регистры RGBC содержат сырые значения внешней освещённости без подсветки датчика.

7. Изменение I²C-адреса и режима

Датчик цвета Баригадам позволяет изменять:

- **I²C-адрес (сохраняется, диапазон 0x40–0x60)**
- **режим (0–5, сохраняется)**

Эти настройки записываются во внутреннюю flash-память и сохраняются после отключения питания.

7.1. Изменение I²C-адреса

- Текущий/заводской адрес: A_old (например, 0x40).
- Новый адрес: A_new (в диапазоне 0x40–0x60).

© 2026, ООО «БАРИГАДАМ». Все права защищены.

Последовательность действий:

1. Подайте питание на датчик.
2. В течение 0,5 с выполните запись по I²C на адрес A_old, передав ровно один байт данных: A_new.

Если команда выполнена успешно:

- датчик сохраняет A_new во flash-памяти,
- светодиод мигает зелёным 3 раза,
- начиная со следующего обмена (и при всех последующих включениях питания), датчик будет использовать адрес A_new.

Если командный байт некорректен (не входит в диапазон 0x40–0x60 и не равен одному из 0xE0–0xE5):

- светодиод мигает красным 3 раза,
- адрес и режим остаются без изменений.

Чтобы сбросить адрес на значение по умолчанию, отправьте 0x40 в течение окна загрузки.

7.2. Изменение режима

- Командные байты режима: 0xE0–0xE5.
- Номер режима = командный байт − 0xE0.

Командный байт	Новый режим
0xE0	0
0xE1	1
0xE2	2
0xE3	3
0xE4	4
0xE5	5

Последовательность действий:

1. Подайте питание на датчик.
2. В течение 0,5 с запишите один байт на текущий I²C-адрес: 0xE0 + mode.

Если команда выполнена успешно:

- датчик сохраняет новый режим во flash-памяти,
- светодиод мигает зелёным 3 раза,
- датчик продолжает работу в этом режиме и сохраняет его для следующего включения питания.

© 2026, ООО «БАРИГАДАМ». Все права защищены.

Если байт не входит в диапазон 0xE0–0xE5 или 0x40–0x60:

- светодиод мигает красным 3 раза,
- изменения не сохраняются.

Если записанный байт совпадает с текущим сохранённым адресом/режимом:

- ничего не изменяется,
- запись во flash-память не выполняется,
- светодиод не мигает (действие не выполняется),
- датчик продолжает работу в штатном режиме.

8. Чтение данных в коде

В библиотеке или проекте можно читать данные с датчика двумя способами.

8.1. Стандартное чтение по регистрам

Это классическая схема доступа к регистрам по I²C, совместимая с будущими версиями прошивки.

1. **Выберите начальный регистр (например, 0x00 для R_L):**
   - Запишите адрес устройства (7-битный, например 0x40)
   - Отправьте один байт: 0x00 (начальный адрес)
2. **Считайте N байт:**
   - Для RGBC считайте 8 байт:
     R_L, R_H, G_L, G_H, B_L, B_H, C_L, C_H
   - При необходимости считайте ещё 1 байт для COLOR (регистр 0x08) и т. д.

Затем восстановите 16-битные значения:

```
R = R_L + (R_H << 8)
G = G_L + (G_H << 8)
B = B_L + (B_H << 8)
C = C_L + (C_H << 8)
```

Используйте COLOR для получения готовой классификации цвета.

8.2. Простое чтение «первых N байт»

Для удобства прошивка датчика цвета Баригадам устроена так, что:

Каждое чтение по I²C всегда начинается с регистра 0x00.
После каждого чтения внутренний указатель автоматически возвращается к 0x00.

© 2026, ООО «БАРИГАДАМ». Все права защищены.

Это означает, что можно просто выполнить:

- Считать 8 байт → всегда получите RGBC (R_L, R_H, G_L, G_H, B_L, B_H, C_L, C_H)
- Считать 9 байт → RGBC + COLOR
- Считать 12–13 байт → полный набор регистров (RGBC + COLOR + MODE + STATUS + DEV_ID + FW)

9. Рекомендации по использованию

- Для лучших результатов в **режимах 0–3**:
  - Держите дистанцию от датчика до поверхности объекта в диапазоне 0-5 см.
  - Избегайте глянцевых поверхностей.
- В режиме 2 датчик оптимален для применений, где используются только красные/зелёные/синие/жёлтые объекты.
- Для собственной обработки значений:
  - Используйте режимы 4 или 5, чтобы получить сырые RGBC на высокой скорости,
  - Применяйте собственные алгоритмы на используемой плате/контроллере.

10. Библиотека Arduino: Barigadam_ColorSensor

Barigadam_ColorSensor — это вспомогательная библиотека для Arduino-совместимых контроллеров. Она написана для API Arduino (Arduino.h, Wire.h, Serial) и работает с:

- Классическими контроллерами Ардуино (Уно, Мега, Нано и т. п.)
- Платами STM32 / ESP32, использующими ядро Arduino

Библиотека оборачивает протокол I²C, описанный в этом руководстве, в простые функции, поэтому можно считывать RGBC, HSV и индекс цвета без ручной работы с регистрами.

Для использования подключите заголовочные файлы в скетче:

```
#include <Wire.h>
#include <Barigadam_ColorSensor.h>
```

Перед использованием функций убедитесь, что в setup() вызваны Wire.begin() и Serial.begin(...).

Barigadam_readRGBC_All(uint8_t addr, uint16_t &R, uint16_t &G, uint16_t &B, uint16_t &C)
Низкоуровневая вспомогательная функция, которая считывает первые 8 байт датчика (регистры 0x00–0x07) и возвращает все четыре 16-битных канала:

- R, G, B — красный, зелёный и синий каналы
- C — Clear-канал (общая интенсивность)

© 2026, ООО «БАРИГАДАМ». Все права защищены.

Эта функция использует шаблон чтения «первых N байт» , описанный в разделе 8. 2, и служит базой для большинства остальных функций.

uint16_t readRGBC(uint8_t addr, uint8_t channel)
Вспомогательная функция, возвращающая один канал RGBC:

- channel = 1 → красный канал (Red)
- channel = 2 → зелёный канал (Green)
- channel = 3 → синий канал (Blue)
- channel = 4 → Clear-канал
- Любое другое значение → возвращает 0

Внутри функция вызывает Barigadam_readRGBC_All() и выбирает один из четырёх каналов.

int readColor (uint8_t addr)
Считывает регистр **COLOR** (0x08) и возвращает классифицированный индекс цвета:

- Возвращает 1…6 — корректный индекс цвета (см. раздел 5.1: чёрный, синий, зелёный, жёлтый, красный, белый)
- Возвращает -1 — цвет не обнаружен (COLOR = 0xFF, «пусто»)
- Возвращает -2 — ошибка I²C (данные недоступны)

Это самый простой способ получить готовую классификацию цвета напрямую из прошивки датчика.

uint8_t readRegister (uint8_t addr, uint8_t reg)
Общая вспомогательная функция для чтения одного 8-битного регистра:

- Устанавливает внутренний указатель регистра на reg
- Считывает один байт и возвращает его

Используйте эту функцию, если требуется прямой доступ к MODE, STATUS, DEV_ID, FW или любому другому регистру из карты регистров.

float readHSV(uint8_t addr, uint8_t c)
Вычисляет приблизительные значения **HSV** по измеренным данным RGBC:

- Внутри считывает все каналы RGBC с помощью Barigadam_readRGBC_All()
- Нормализует R, G, B по каналу Clear
- Выполняет расчёт:
  - оттенок H в градусах (0…360)
  - насыщенность S (0…1)
  - яркость V (0…1, ограничена внутренней константой)

Возвращаемое значение зависит от c:

- c = 1 → возвращает H (оттенок, градусы)
- c = 2 → возвращает S (насыщенность, 0–1)
- любое другое значение → возвращает V (яркость/значение, 0–1)

© 2026, ООО «БАРИГАДАМ». Все права защищены.

Эта функция полезна, если удобнее работать в цветовом пространстве HSV вместо сырых RGBC.

void sendToSensor(uint8_t currentAddr, uint8_t dataToSend)
Вспомогательная функция для изменения **I²C-адреса** или **режима** в течение 0,5-секундного окна загрузки (см. раздел 7):

- Если dataToSend находится в диапазоне 0…5
  → он интерпретируется как номер режима.
  Функция автоматически добавляет 0xE0 и отправляет соответствующий командный байт (0xE0…0xE5).
- Если dataToSend находится в диапазоне 0x40…0x60
  → он интерпретируется как новый I²C-адрес.

Функция выводит диагностические сообщения в Serial и многократно отправляет выбранный байт на currentAddr.

Типовой порядок использования:

1. Загрузите скетч, вызывающий sendToSensor(...) в setup().
2. Откройте монитор порта (Serial Monitor).
3. Перезапустите питание датчика, чтобы он получил команду во время окна загрузки.

Если параметры некорректны (адрес вне диапазона, совпадает с текущим и т. д.), функция выводит сообщение об ошибке и ничего не выполняет.

void I2Cscan()
Простой сканер шины I²C для отладки:

- Сканирует все 7-битные адреса от 0x01 до 0x7E
- Выводит все обнаруженные устройства в монитор порта (Serial Monitor)
- Повторяет сканирование каждые 5 секунд в бесконечном цикле

Это удобно для проверки, что датчик цвета Баригадам присутствует на ожидаемом адресе (по умолчанию 0x40 или любой пользовательский адрес от 0x40 до 0x60).

Эти функции можно свободно комбинировать в скетчах Arduino:
например, один раз использовать I2Cscan() для поиска устройства, затем sendToSensor() для настройки адреса и режима, а после этого readColor() или readHSV() для высокоуровневого определения цвета в роботе или проекте.

© 2026, ООО «БАРИГАДАМ». Все права защищены.

11. Важная информация и уведомление

Информация в этом документе отражает текущие знания и понимание компании Баригадам на дату публикации. Она частично основана на данных, полученных от производителей компонентов и других третьих сторон. Баригадам не гарантирует, что такая информация от третьих сторон является полной, точной или актуальной. Мы постоянно работаем над улучшением нашей документации, однако можем не проводить разрушающие испытания, полный химический анализ или другие исчерпывающие проверки для каждой партии поступающих материалов.

Часть информации считается конфиденциальной для Баригадам и её поставщиков; поэтому некоторые идентификаторы (например, внутренние номера деталей) могут не раскрываться.

Уведомление

Баригадам оставляет за собой право в любое время и без предварительного уведомления изменять конструкцию, функции или технические характеристики любого продукта, описанного в данном документе, а также прекращать выпуск любого продукта. Перед размещением заказа или проектированием новых систем с использованием продукции Баригадам клиентам следует всегда получать у Баригадам самую актуальную информацию о продукте и документацию.

Баригадам не несёт ответственности за использование каких-либо схем, примеров или рекомендаций, содержащихся в данном документе, а также за проектирование или работу любых изделий клиента, в состав которых входят компоненты Баригадам. Продажа или использование продукции Баригадам не предоставляет никакой лицензии — ни явной, ни подразумеваемой — на какие-либо патенты, авторские права, права на топологии микросхем или иные права интеллектуальной собственности. Баригадам также не гарантирует, что описанные здесь схемы или конструкции свободны от претензий третьих лиц в отношении интеллектуальной собственности.

Продукция **Баригадам** предоставляется «как есть» без каких-либо явных или подразумеваемых гарантий, включая, помимо прочего, гарантии товарной пригодности, пригодности для конкретной цели или отсутствия нарушений прав третьих лиц. **Баригадам** не несёт ответственности за какие-либо прямые, косвенные, случайные, специальные или последующие убытки, возникающие в результате использования либо невозможности использования любой продукции **Баригадам** или документации.

Применения, критически важные для безопасности

ПРОДУКЦИЯ БАРИГАДАМ НЕ РАЗРАБАТЫВАЕТСЯ, НЕ ТЕСТИРУЕТСЯ И НЕ ПРЕДНАЗНАЧЕНА ДЛЯ ИСПОЛЬЗОВАНИЯ В ПРИЛОЖЕНИЯХ, ГДЕ ОТКАЗ ИЛИ НЕИСПРАВНОСТЬ МОГУТ РАЗУМНО ОЖИДАЕМО ПРИВЕСТИ К ТРАВМАМ, СМЕРТИ ИЛИ СЕРЬЁЗНОМУ УЩЕРБУ ИМУЩЕСТВУ. СЮДА ОТНОСЯТСЯ, В ЧАСТНОСТИ, СИСТЕМЫ ЖИЗНЕОБЕСПЕЧЕНИЯ, ИМПЛАНТИРУЕМЫЕ МЕДИЦИНСКИЕ УСТРОЙСТВА, АВАРИЙНОЕ ИЛИ КРИТИЧЕСКИ ВАЖНОЕ ДЛЯ БЕЗОПАСНОСТИ ОБОРУДОВАНИЕ, А ТАКЖЕ ЛЮБЫЕ ДРУГИЕ ПРИЛОЖЕНИЯ ТИПА «FAIL-SAFE».

ЛЮБОЕ ТАКОЕ ИСПОЛЬЗОВАНИЕ ПРОДУКЦИИ БАРИГАДАМ ОСУЩЕСТВЛЯЕТСЯ ИСКЛЮЧИТЕЛЬНО НА РИСК КЛИЕНТА, И КЛИЕНТ НЕСЁТ ЕДИНОЛИЧНУЮ ОТВЕТСТВЕННОСТЬ ЗА ВСЕ НЕОБХОДИМЫЕ МЕРЫ БЕЗОПАСНОСТИ, РЕЗЕРВИРОВАНИЯ И СОБЛЮДЕНИЯ ТРЕБОВАНИЙ.

© 2026, ООО «БАРИГАДАМ». Все права защищены.