

ПЕРВАЯ НАУЧНАЯ МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ МОЛОДЕЖНОГО НАУЧНОГО СООБЩЕСТВА ДГТУ «ИНТЕГРАЦИЯ: ГОСУДАРСТВО. НАУКА. БИЗНЕС»



УДК 004.9

Система картографирования на базе ESP32-CAM: разработка, тестирование и применение

М.И. Азнабаев, Д.Ю. Донской

Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

Аннотация

Статья описывает мобильную систему картографирования на основе микроконтроллера ESP32-CAM, обеспечивающую сбор геоданных и фотографий местности с GPS-метками. Для повышения точности позиционирования выполнено сравнительное тестирование нескольких GPS-модулей в условиях плотной городской застройки. В статье уделено внимание программному обеспечению, включая многозадачность на базе FreeRTOS, парсинг NMEA-строк с помощью TinyGPSPlus и реализацию веб-интерфейса через ESPAsyncWebServer. Представлены фрагменты кода управляющей программы.

Ключевые слова: ESP32, картографирование, мобильная система, GPS, тестирование

Для цитирования. Азнабаев М.И., Донской Д.Ю. Система картографирования на базе ESP32-CAM: разработка, тестирование и применение. *Молодой исследователь Дона*. 2025;10(5):85–91.

ESP32-CAM Based Mapping System: Development, Testing and Application

Marat I. Aznabaev, Danila Yu. Donskoi

Don State Technical University, Rostov-on-Don, Russian Federation

Abstract

The article describes a mobile mapping system based on the ESP32-CAM microcontroller, which enables collection of geodata and photographs of the area with GPS tags. To improve the accuracy of positioning, comparative testing of several GPS modules was performed in the densely built-up urban areas. The article focuses on studying the software including analysis of multitasking based on FreeRTOS, parsing NMEA sentence strings using TinyGPSPlus, and implementation of the web interface via ESPAsyncWebServer. Fragments of the control program code have been provided.

Keywords: ESP32, mapping, mobile system, GPS, testing

For Citation. Aznabaev MI, Donskoy DYu. ESP32-CAM Based Mapping System: Development, Testing and Application. *Young Researcher of Don*. 2025;10(5):85–91.

Введение. В последние годы малогабаритные высокопроизводительные микроконтроллеры со встроенным модулем Wi-Fi и поддержкой камер открыли новые возможности для автономного картографирования и мониторинга различных объектов. В данной работе был использован модуль ESP32-CAM AI Thinker, совмещающий на одной плате микроконтроллер ESP32 и сенсор OV2640. Цель исследования — разработка и экспериментальная проверка прототипа автономного устройства на базе ESP32-CAM для фотосъёмки с геопривязкой, а также выбор оптимального GPS-модуля путём анализа точности и устойчивости его работы в условиях плотной городской застройки и на открытых пространствах. Для повышения точности позиционирования проведено тестирование различных GPS-модулей: N10HD Mobiletek, Ublox NEO-7M, Ublox NEO-M8N, Ublox M8030-KT (BN-220, BN-880).



Рис. 1. ESP32-CAM AI-Thinker [5]

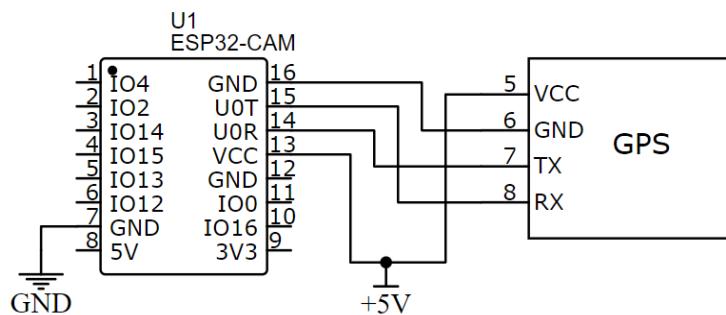


Рис. 2. Схема подключения GPS модулей к ESP32

Основная часть. Мобильная система картографирования разработана на основе микроконтроллера ESP32-S [1] и используется в виде модуля ESP32-CAM AI-Thinker [2] (рис. 1). Вместе с модулем используется 2-мегапиксельная камера OV2640 [3]. Модуль имеет 4 Мб PSRAM [4], который используется для буферизации изображений с камеры в потоковом видео. Такое решение позволяет использовать более качественную видеозапись без сбоев в работе микроконтроллера. Также есть возможность подключения внешней антенны для обеспечения связи на большее расстояние через разъем типа IPEX.

Данный модуль включает разведение схемотехники: под карту памяти microSD, 10 GPIO и контакты питания — как напрямую к контроллеру, так и через стабилизатор напряжения на 3,3 В. Питается устройство от 5 В через USB-разъем. Питание поступает на контакты VCC и GND. Для хранения данных используется внешняя SD карта, подключенная в разъем на плате. За определение координат отвечает GPS-модуль, взаимодействующий с ESP32 посредством интерфейса UART (рис. 2).

В угоду повышения точности фиксации координат проведём сравнение GPS-модулей и выберем лучший. Составим таблицу 1 для наглядности и удобства сравнения с характеристиками GPS-модулей DX-GP10 GPS, Ublox NEO-7M, Ublox NEO-M8N, BN-220(BN-880), участвовавших в тестах [6–10].

Таблица 1
Характеристики GPS-модулей

Параметр	DX-GP10	Ublox NEO-7M	Ublox NEO-M8N	BN-220 (BN-880)
Чипсет (Chipset)	N10HD Mobiletek	u-blox 7	u-blox M8N	u-blox M8030-KT
Формат приёма (Receiving Format)	GPS, GLONASS, BeiDou, A-GNSS	GPS, QZSS	GPS, GLONASS, BeiDou, Galileo	GPS, GLONASS, BeiDou
Частота (Frequency)	L1, 1575.42 МГц	L1, 1575.42 МГц	L1, 1575.42 МГц	L1, 1575.42 МГц
Каналы (Channels)	56	56	72	72
Горизонтальная точность позиционирования (с SBAS)	2.5 м	2.5 м	2.0 м	2.0 м
Отслеживание (Tracking)	-159 дБм	-161 дБм	-167 дБм	-167 дБм
Повторное определение (Reacquisition)	-157 дБм	-158 дБм	-160 дБм	-160 дБм
Холодный старт (Cold start)	-145 дБм	-148 дБм	-148 дБм	-148 дБм
Горячий старт (Hot start)	-155 дБм	-156 дБм	-157 дБм	-157 дБм
Холодный старт (Cold start)	29 с	30 с	26 с	26 с
Тёплый старт (Hot start)	5 с	5 с	3 с	3 с
Частота обновления (Max navigation update rate)	1 Гц	1 Гц	10 Гц	10 Гц
Протокол данных (Data Protocol)	NMEA	NMEA, UBX	NMEA, UBX, RTCM	NMEA, UBX
Напряжение питания (VCC)	3.3–5 В	3.0–5.5 В	2.7–3.6 В	2.7–3.6 В
Потребляемый ток	25 мА	25 мА	21 мА	21 мА

Программная часть системы выполнена с использованием фреймворка Arduino для ESP32. Упор — на много задачность с использованием FreeRTOS [11], что позволило разделить работу с GPS, камеру и веб-интерфейс на независимые процессы. Инициализация камеры осуществляется в функции configInitCamera(). Эта функция настраивает пины и параметры камеры, включая частоту тактирования и формат изображения. Проверяется наличие внешней PSRAM, от чего зависит выбор разрешения и качества снимков. Если PSRAM доступна, используется HD-разрешение и качество JPEG = 10. Если нет, выбирается SVGA и JPEG = 12. Пример кода представлен на рис. 3.

```
// Initializes the camera
void configInitCamera() {
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;
    config.pin_sscb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 20000000;
    config.pixel_format = PIXFORMAT_JPEG; // YUV422,GRAYSCALE,RGB565,JPEG
    config.grab_mode = CAMERA_GRAB_LATEST;
    if (psramFound()) {
        config.frame_size = FRAMESIZE_HD;
        config.jpeg_quality = 10;           // 0-63 lower number means higher quality
        config.fb_count = 1;
    } else {
        config.frame_size = FRAMESIZE_SVGA;
        config.jpeg_quality = 12;
        config.fb_count = 1;
    }
    digitalWrite(PWDN_GPIO_NUM, LOW);
    delay(10);
    digitalWrite(PWDN_GPIO_NUM, HIGH);
    delay(10);
    esp_err_t err = esp_camera_init(&config);
    if (err != ESP_OK) {ESP.restart();return;}
}
```

Рис. 3. Функция configInitCamera()

При включении устройства активируется детектор brownout — он предотвращает перезагрузки при просадках питания. Затем инициализируются файловые системы SPIFFS, загружается конфигурация Wi-Fi, настраивается точка доступа и запускается веб-сервер. Wi-Fi инициализируется функцией initWiFi(), которая использует SSID и пароль из JSON-конфигурации в SPIFFS. Передача и приём изображений выполняются через HTTP-запросы к веб-серверу. Обработчик запроса на съёмку /photo реализован следующим образом и показан на рис. 4.

```
server.on("/photo", HTTP_GET, [](AsyncWebServerRequest *request) {
    vTaskResume(TaskHandlePhoto); request->send(200);});
```

Рис. 4. Обработчик запроса на съёмку

Параллельно с веб-интерфейсом работает задача чтения данных с GPS модуля. Она запускается через FreeRTOS-функцию xTaskCreate и непрерывно парсит входящий поток с UART. Обработка координат осуществляется библиотекой TinyGPSPlus. Пример кода задачи представлен на рис. 5.

```
void GPSRead(void *arg) {
    while (1) {
        esp_task_wdt_reset();
        if (Serial.available()) {
            char dataGPS = Serial.read();
            if (Tgps.encode(dataGPS)) {
                // displayInfo();
            }
            vTaskDelay(1 / portTICK_PERIOD_MS);
        }
    }
}
```

Рис. 5. Задача чтения данных с GPS модуля

Отдельная задача отвечает за съёмку фотографий с заданным интервалом. Этот интервал может быть задан пользователем через POST-запрос /pr. Периодическая съёмка осуществляется путём пробуждения задачи съёмки с помощью vTaskResume(TaskHandlePhoto) и последующего выполнения функции takeSavePhoto(). В этой функции выполняется получение кадра с камеры, генерация имени файла и сохранение как изображения, так и связанного с ним файла координат. Код функции takeSavePhoto() представлен на рис. 6.

```
void takeSavePhoto(fs::FS &fs, camera_fb_t *fb) {
    esp_camera_fb_return(fb); // dispose the buffered image
    fb = NULL; // reset to capture errors
    fb = esp_camera_fb_get();
    if (!fb) { ESP.restart(); }
    String path = getPictureFilename();
    File file = fs.open(path.c_str(), FILE_WRITE);
    if (!file) {
        Serial.printf("Failed to open file in writing mode");
    } else {
        file.write(fb->buf, fb->len); // payload (image), payload length
    }
    file.close();
    path = getGPSFilename(); // iterator
    File file_txt = fs.open(path.c_str(), FILE_WRITE);
    if (!file_txt) {
        infoPath = "ошибка сохранения.";
    } else {
        file_txt.printf("\n\n%d:%d\n%d/%d/%d\n%f|%f|%f,%f\n%f\n",
                        Tgps.time.hour(), Tgps.time.minute(), Tgps.time.second(),
                        Tgps.date.day(), Tgps.date.month(),
                        Tgps.date.year(), Tgps.location.lat(), Tgps.location.lng(),
                        Tgps.speed.kmph(), Tgps.course.deg());
        String data = "";
        serializeJson(GPSParam, data);
        file_txt.print("%s", data.c_str());
        infoPath = path;
    }
    file_txt.close();
}
```

Рис. 6. Функции takeSavePhoto()

Имя файла формируется на основе GPS-времени (при его доступности). Все данные записываются на SD-карту, что обеспечивает автономность системы и независимость от сетевых подключений. Веб-интерфейс на базе ESPAsyncWebServer — это простая страница, доступная через Wi-Fi, которая предоставляет пользователю основные функции: запуск и остановку периодической съёмки, ручной снимок, просмотр последних координат и настройку параметров устройства. Визуально интерфейс состоит из нескольких кнопок и текстовых полей, обеспечивая интерактивное взаимодействие с системой. Для пользователя реализованы запросы типов GET и POST, а также вывод текущих координат, получаемых от GPS-модуля, в реальном времени. Скриншот интерфейса приведён на рис. 7.

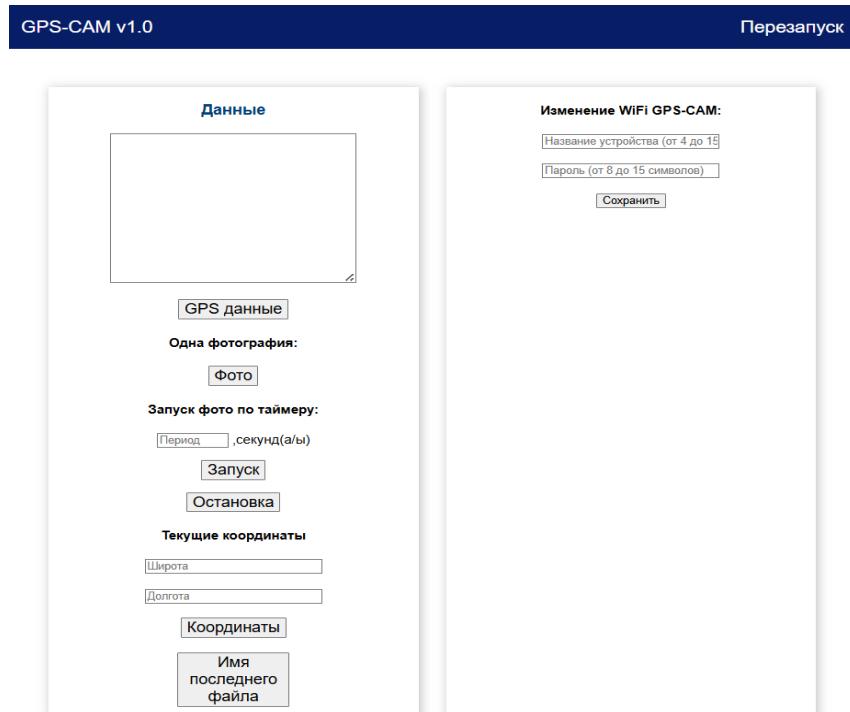


Рис. 7. Пользовательский интерфейс

Для наглядного представления взаимодействия программных и аппаратных компонентов системы была составлена блок-схема алгоритма, представленная на рис. 8.

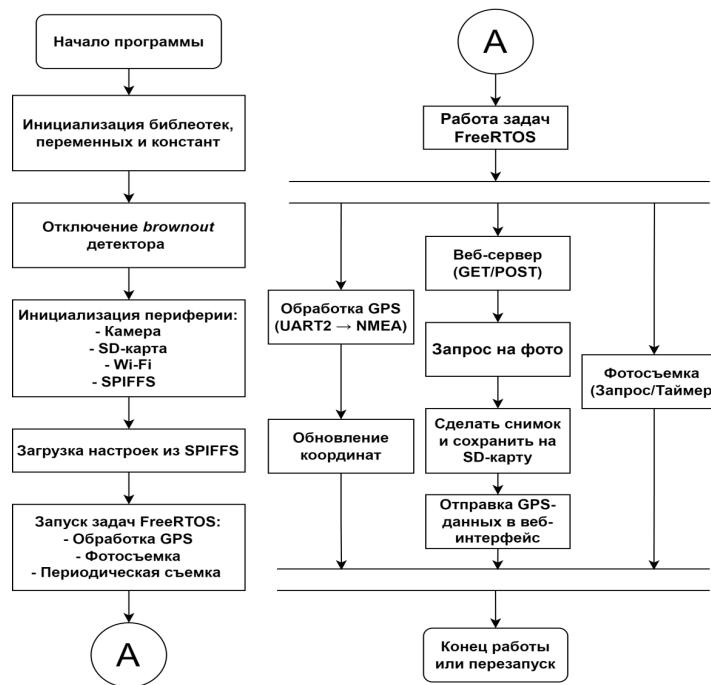


Рис. 8. Алгоритм управляющей программы для микроконтроллера ESP32

Тестирование проводилось в городской среде, где качество сигнала ограничено. Система размещалась на стекле автомобиля, а съёмка велась каждые 10 секунд. Модули BN-220 и BN-880 тестировались при движении в разных направлениях одного и того же маршрута. Такой подход позволил объективно сравнить их устойчивость и точность. Испытания показали, что именно эти два модуля обеспечили наиболее стабильные результаты в условиях застройки. Остальные модули — DX-GP10, Ublox NEO-7M и NEO-M8N — демонстрировали большую погрешность. Визуализация данных выполнялась в среде QGIS [12], результаты представлены на рис. 9.

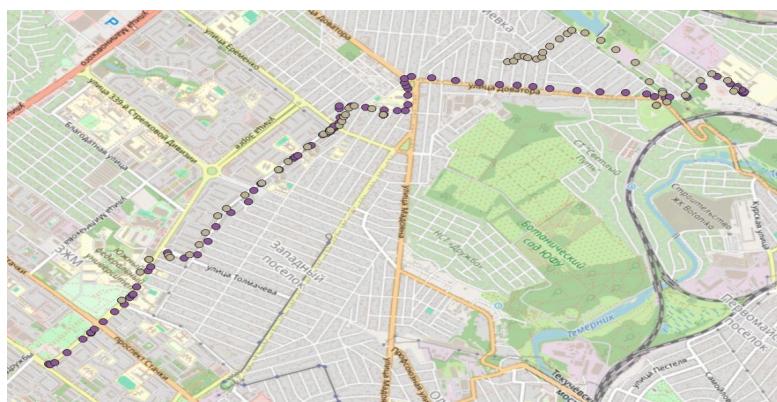


Рис. 9. Полученные координаты с BN-220 (серые точки) и BN-880 (фиолетовые точки)

В ходе испытаний были собраны и систематизированы экспериментальные данные, представленные в таблице 2.

Таблица 2
Экспериментальные данные GPS модулей

Параметр	BN-220	BN-880	Ublox NEO-7M	DX-GP10 GPS	Ublox NEO-M8N
Количество спутников	14	13	8	10	12
Время холодного старта	4 мин	3 мин	16 мин	8 мин	5 мин
Погрешность	2–5 м	2–3 м	4–14 м	6–10 м	2–4 м
Потребление тока	21 мА	21 мА	25 мА	25 мА	22 мА

Пример работы функции фотографирования приведён на рис. 10, внешний вид системы — на рис. 11.



Рис. 10. Пример функции фотографирования



Рис. 11. Внешний вид мобильной системы картографирования

Заключение. Разработанная система на базе микроконтроллера ESP32 обеспечивает выполнение задач картографирования, включая автоматическую фотосъёмку и фиксацию GPS-координат с заданным интервалом. Конструкция компактна, питается от интерфейса USB. По результатам испытаний наилучшие показатели точности и устойчивости продемонстрировал модуль BN-880, обеспечив минимальную погрешность 2–3 м. Визуальный анализ маршрута в среде QGIS подтвердил эффективность данного решения в условиях городской застройки. Проведённые исследования показали, что разработанная система отличается надёжностью и может применяться в задачах мониторинга и картографирования.

Финансирование. Работа проведена в рамках выполнения проекта «Математическое моделирование и алгоритмы моделирования роста растений на основе автоматизированной картографической системы» (FZNE2024-0006).

Список литературы

1. *ESP32 Series. Datasheet Version 5.2(2.4 GHz Wi-Fi + Bluetooth® + Bluetooth LE SoC)*. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (accessed: 27.10.2025).
2. *ESP32-CAM Module. Product Specification*. URL: https://github.com/raphaelbs/esp32-cam-ai-thinker/blob/master/assets/ESP32-CAM_Product_Specification.pdf (accessed: 27.10.2025).
3. *OV2640 Datasheet*. URL: https://github.com/raphaelbs/esp32-cam-ai-thinker/blob/master/assets/OV2640_Datasheet.pdf (accessed: 27.10.2025).
4. *IPUS IPS6404 Datasheet*. URL: https://github.com/raphaelbs/esp32-cam-ai-thinker/blob/master/assets/IPUS_IPS6404_Datasheet.pdf (accessed: 27.10.2025).
5. *ESP32-CAM AI-Thinker*. URL: <https://www.daraz.com.np/products/esp32-camera-module-i122735676.html> (accessed: 27.10.2025).
6. *N10HD GNSS Module Datasheet. Shanghai Mobiletek Communication Ltd*. URL: <https://285624.selcdn.ru/syms1/iblock/97c97c353743871f5b4cedfd08c60247ece/20b05ade7f3ccf1811264c0c8e5f5910.pdf> (accessed: 27.10.2025).
7. *NEO-7. U-blox 7 GNSS Datasheet*. URL: https://content.u-blox.com/sites/default/files/products/documents/NEO-7_DataSheet_%28UBX-13003830%29.pdf (accessed: 27.10.2025).
8. *NEO-M8. U-blox M8 Concurrent GNSS Modules Datasheet*. URL: https://content.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_UBX-15031086.pdf (accessed: 27.10.2025).
9. *UBX-M8030. Versatile U-blox M8 GNSS Chips Datasheet*. URL: https://content.u-blox.com/sites/default/files/products/documents/UBX-M8030_ProductSummary_%28UBX-15029937%29.pdf (accessed: 27.10.2025).
10. *BN-220 GPS Module + Antenna Datasheet*. URL: <https://files.banggood.com/2016/11/BN-220%20GPS+Antenna%20datasheet.pdf> (accessed: 27.10.2025).
11. *FreeRTOS Overview*. URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/freertos.html> (accessed: 27.10.2025).
12. *QGIS Documentation. Руководство пользователя QGIS*. URL: https://docs.qgis.org/3.40/ru/docs/user_manual/index.html (accessed: 27.10.2025).

Об авторах:

Марат Игоревич Азнибаев, магистр кафедры «Автоматизация производственных процессов и производств» Донского государственного технического университета (344003, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1), ttrtr3330@gmail.com

Данила Юрьевич Донской, ассистент и аспирант кафедры «Автоматизация производственных процессов и производств» Донского государственного технического университета (344003, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1), dand22@bk.ru

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the Authors:

Marat I. Aznabaev, Bachelor's Degree Student of the Automation of Production Processes Department, Don State Technical University (1, Gagarin Square, Rostov-on-Don, 344003, Russian Federation), ttrtr33300@gmail.com

Danila Yu. Donskoi, Assistant and Post Graduate Student of the Automation of Production Processes Department, Don State Technical University (1, Gagarin Square, Rostov-on-Don, 344003, Russian Federation), dand22@bk.ru

Conflict of Interest Statement: the authors declare no conflict of interest.

All authors have read and approved the final manuscript.