

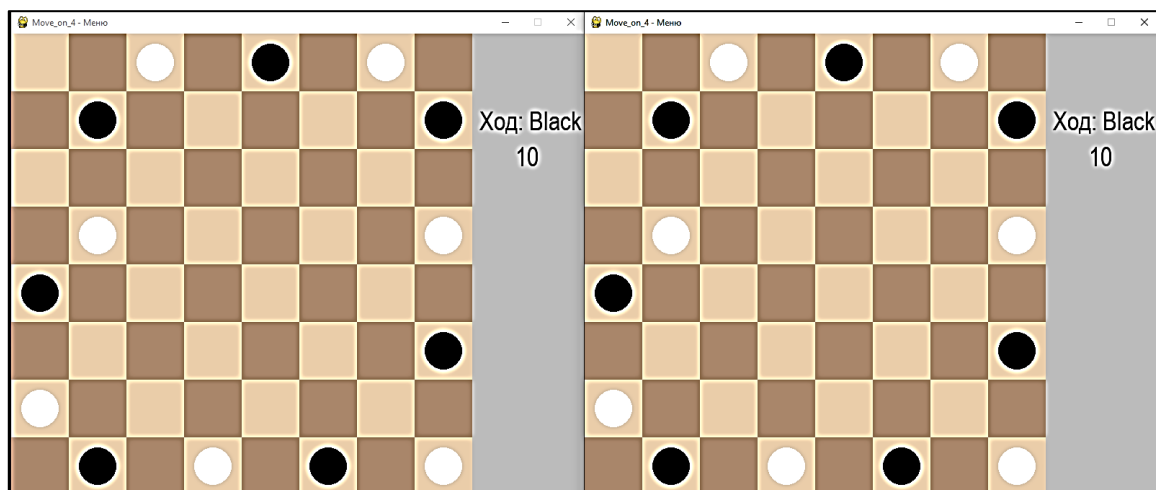


Рисунок 5 – Игра по сети

Одной из интересных особенностей является добавление таймера на каждый ход. Игроку отводится 15 секунд на размышление, после чего (в одиночной игре) управление временно передаётся компьютеру, совершающему случайный ход. Это поддерживает динамику игры и побуждает игрока к активному участию.

Поскольку проект разрабатывался с учётом кроссплатформенности, итоговое приложение может быть скомпилировано в исполняемый файл (например, .exe), не требующий установки Python и дополнительных библиотек у конечного пользователя [3]. Это расширяет область применения и делает игру доступной для более широкой аудитории.

Таким образом, в ходе работы была создана полноценная цифровая реализация настольной игры с двумя режимами: одиночным и сетевым. Проект может служить основой для дальнейшего развития, включая улучшение алгоритма соперника, добавление новых игровых режимов и более сложной логики поведения.

Литература

- 1 Свиггарт, Э. Програмируем игры на Python / Э. Свиггарт. – М. : ДМК Пресс, 2019. – 384 с.
- 2 Шоу, З. Научитесь программировать на Python / З. Шоу. – СПб. : Питер, 2021. – 432 с.
- 3 Тим, Я. PyInstaller. Использование PyInstaller для создания исполняемых файлов на Python / Я. Тим. – М. : Издательство Хабр, 2021. – 128 с.

УДК 004.9

Я. П. Ходанович

ПОЛУЧЕНИЕ И АНАЛИЗ КАРДИОДАННЫХ С ИСПОЛЬЗОВАНИЕМ ESP32

В статье рассматриваются вопросы анализа больших данных с использованием микроконтроллера ESP32. Для анализа выбраны кардиоданные. Подробно рассмотрены аппаратные средства для сбора и передачи данных. Отмечены особенности различных датчиков. Указаны пути предварительной обработки и сбора данных. Перечислены возможные способы визуализации и анализа данных.

Аппаратная часть. ESP32 – это мощный микроконтроллер с двухъядерным процессором, встроенным Wi-Fi и Bluetooth, что делает его отличной платформой для

работы с биосенсорами [1]. Он имеет несколько аналоговых входов (ADC), цифровые порты и поддерживает распространённые протоколы связи, такие как I2C и SPI. Это позволяет подключать к нему разнообразные датчики, включая те, что используются для измерения сердечного ритма.

Наиболее простой вариант – это оптический датчик пульса Pulse Sensor (рисунок 1). Он использует светодиод и фотодетектор, чтобы регистрировать изменения объёма крови в капиллярах при каждом ударе сердца. Такой датчик подключается к аналоговому входу ESP32, например, к пину GPIO34. Его питание можно взять с 3.3 V или 5 V (в зависимости от модели), а землю – на общий GND. Данные с датчика будут приходить в виде аналоговых сигналов, которые нужно оцифровывать встроенным АЦП ESP32.

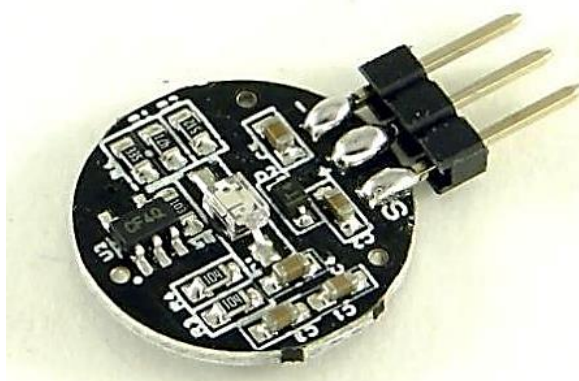


Рисунок 1 – Оптический датчик пульса Pulse Sensor

Если хочется чего-то более функционального, можно использовать датчик MAX30100 (рисунок 2) или MAX30102 (рисунок 3). Эти модули совмещают в себе фотоплетизмографию (PPG) для измерения пульса и могут также определять насыщение крови кислородом (SpO_2). Они работают по цифровому протоколу I2C, поэтому нужно подключить SDA и SCL к соответствующим пинам ESP32, например, GPIO21 и GPIO22. Также требуется питание 1.8–3.3 V и земля. Эти датчики довольно чувствительны, и желательно подключать их через дополнительные резисторы pull-up на линии I2C, если их нет на плате.



Рисунок 2 – Датчик MAX30100



Рисунок 3 – Датчик MAX30102

Для регистрации ЭКГ можно использовать модуль AD8232 (рисунок 4). Это специализированный аналоговый фронтенд для биопотенциалов. Он подключается по аналоговой линии – один из выходов соединяется с аналоговым входом ESP32, питание можно брать от 3.3 V. Датчик использует три электрода, которые прикрепляются к телу – два для измерения сигнала, один для заземления. Важно использовать экранированные провода и располагать провода подальше от источников шумов, так как ЭКГ-сигналы очень слабые.



Рисунок 4 – Модуль AD8232

Сбор данных. Сбор данных с кардиодатчиков на базе ESP32 – это ключевой этап, который определяет точность и надёжность всей системы мониторинга. В этом процессе важно учитывать тип используемого датчика, способ передачи данных и особенности аналоговых или цифровых сигналов.

Если используется аналоговый датчик, например Pulse Sensor или выход модуля AD8232, данные поступают на аналоговый вход ESP32 в виде изменяющегося напряжения. Это напряжение отражает биосигналы – изменения объёма крови при ударах сердца или электрическую активность сердца. ESP32 использует встроенный аналогово-цифровой преобразователь (ADC) для преобразования этого напряжения в цифровые значения. Эти значения нужно считывать с высокой частотой – желательно от 100 до 500 Гц, чтобы сохранить достаточную детализацию сигнала. Однако ADC на ESP32 может быть подвержен шумам и не всегда даёт идеальную точность, особенно без фильтрации.

Если применяется цифровой датчик, такой как MAX30100 или MAX30102, данные передаются в виде цифровых значений по шине I2C (рисунок 5). В этом случае сбор данных происходит через обмен с регистрами датчика. Датчик самостоятельно формирует поток чисел, соответствующих интенсивности отражённого инфракрасного или красного света. Эти значения отражают колебания объёма крови в капиллярах. Частота считывания и глубина данных задаются при инициализации – можно настроить, например, частоту 100 Гц, разрешение 16 бит и режим работы (например, только пульс, или пульс и SpO₂ одновременно).

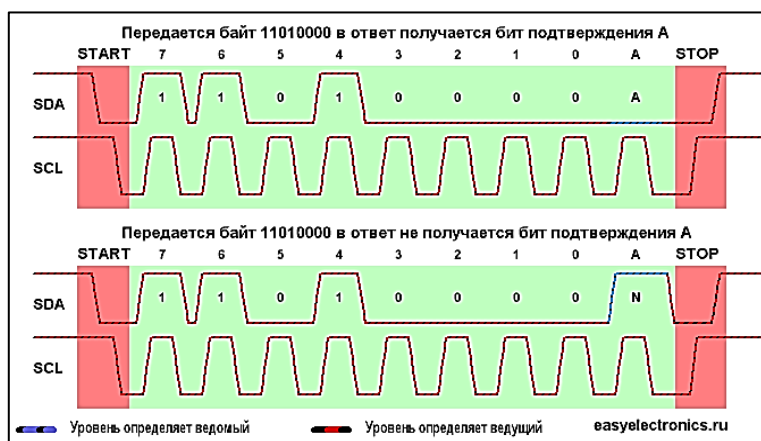


Рисунок 5 – Процесс передачи данных по шине I2C

После получения сигнала, будь то аналоговый или цифровой, данные чаще всего буферизуются в массив или кольцевой буфер. Это нужно для того, чтобы иметь временное окно сигнала, по которому можно делать анализ – например, искать пики, считать частоту, сглаживать шумы. На этом этапе также могут применяться простейшие цифровые фильтры, например, скользящее среднее или пороговая фильтрация.

Сбор данных может быть организован в виде таймера или прерывания, особенно если необходимо обеспечить стабильную частоту выборки. Это особенно важно при измерении ЭКГ или при вычислении вариабельности сердечного ритма (HRV), где даже незначительное нарушение интервалов может исказить результат.

Полученные данные можно сразу анализировать на ESP32, либо временно сохранять и отправлять для последующей обработки на внешний сервер или устройство через Wi-Fi, Bluetooth или по UART. Таким образом, сбор данных представляет собой не просто считывание показаний, а организованный поток цифровой информации, с которым можно работать в реальном времени или асинхронно.

Предобработка и анализ кардиосигнала. Предобработка сигнала – ключевой этап в системе анализа кардиоданных. Сырые данные с датчиков (аналоговых, таких как Pulse Sensor и AD8232, или цифровых, например MAX30102) содержат шумы, артефакты движения, дрейф базовой линии и другие искажения. Для очистки применяют цифровую фильтрацию: полосовые фильтры пропускают только нужный диапазон частот (обычно 0,5–4 Гц для ЧСС), фильтры нижних и верхних частот устраняют шумы и дрейф, а метод скользящего среднего дополнительно сглаживает сигнал. При резких выбросах используют пороговую фильтрацию, заменяя некорректные значения предыдущими корректными. Нормализация амплитуды упрощает визуализацию и передачу данных. Для сложных сигналов, таких как ЭКГ, применяются более продвинутые методы: медианные фильтры, фильтры Баттерворта, коррекция базовой линии и контроль равномерности выборки. Эти методы позволяют преобразовать искажённый сигнал в стабильные данные, пригодные для дальнейшего анализа.

Анализ кардиоданных. Анализ начинается с определения частоты сердечных сокращений (ЧСС) путём поиска пиков (например, R-пиков в ЭКГ) и расчёта интервалов между ними ($bpm = 60 / \text{интервал в секундах}$). Для повышения надёжности ЧСС усредняют по нескольким ударам.

Более глубокий уровень анализа – вариабельность сердечного ритма (HRV), отражающая работу вегетативной нервной системы. HRV рассчитывается на основе последовательности RR-интервалов, с использованием показателей SDNN, RMSSD и частотного анализа (LF / HF).

Дополнительно форма ЭКГ-сигнала позволяет выявлять аритмии, экстрасистолы, тахикардии и другие патологии, анализируя морфологию комплексов QRS и интервалы между ними.

Передача и визуализация данных. Обработанные данные можно передавать и визуализировать через Wi-Fi или Bluetooth. ESP32 отправляет данные на серверы по MQTT, HTTP или WebSocket, обеспечивает отображение графиков в реальном времени либо передаёт данные на смартфоны через BLE. При необходимости ESP32 может работать как локальный веб-сервер, отображая пульс и другие показатели без доступа в интернет.

Литература

1 ESP32 – Википедия. – URL: <https://ru.wikipedia.org/wiki/ESP32> (дата обращения: 15.04.2025).

УДК 004.9

Д. С. Чернокал

О СОЗДАНИИ ТЕЛЕГРАМ-БОТА ДЛЯ ИНТЕРНЕТ-МАГАЗИНА

В статье рассматриваются вопросы разработки телеграм-бота для интернет-магазина. Описываются основные функции бота. Приводится схема работы