

СОДЕРЖАНИЕ ФАЙЛОВОГО АРХИВА, СОПРОВОЖДАЮЩЕГО НАБОР «БАЛАНСИРУЮЩИЙ РОБОТ НА БАЗЕ ESP32 В СРЕДЕ ARDUINO IDE »

Электронный архив к книгеложен на FTP-сервер издательства по адресу:
<ftp://ftp.bhv.ru/9785977566858.zip>. Ссылка доступна и со страницы книги на сайте www.bhv.ru.

Структура архива представлена в таблице:

Папки	Описание	Примечание
listing_2_1	Файл listing_2_1.ino соответствует листингу 2.1 — программа веб-сервера, демонстрирующего температуру внутри контроллера ESP32. Веб-сервер основан на стандартном примере для ESP32 (AdvancedWeb-Server), но вместо рандомного графика мы строим график температуры ядра ESP32	Глава 2*
listing_3_1	Файл listing_3_1.ino соответствует листингу 3.1 — программа запуска шагового мотора. Запускаем шаговый мотор, делаем 600 шагов, время между шагами 3 миллисекунды. Когда все шаги сделаны, двигатель останавливается, а через 500 миллисекунд с него снимается напряжение	Глава 3

listing_3_2	<p>Файл listing_3_2.ino соответствует листингу 3.2 — программа вращения шагового мотора с заданным ускорением и иными параметрами:</p> <pre>MAXi = 6000; // Количество шагов; min_steplong = 200; // Длительность самого короткого шага в микросекундах (максимальная скорость); max_steplong = 3000; // Длительность стартового шага, первый шаг после остановки имеет данную длину; acceleration = 10000; // Ускорение в шагах на секунду в квадрате. С данным ускорением раскручивается вал двигателя при старте и с данным отрицательным ускорением тормозит</pre>	Глава 3
listing_3_3	<p>Файл listing_3_3.ino соответствует листингу 3.3 — программа вращения колес по вводимым данным (управление мотором через последовательный порт).</p> <p>В отличие от предыдущего примера, этот пример позволяет в окне Монитора порта задать параметры: количество шагов, максимальную скорость, ускорение. Скорость порта устанавливается 115 200 бит/с.</p> <p>Параметры задаются путем передачи последовательности из трех параметров:</p> <p>S+число — задается скорость;</p> <p>M+число — задается количество шагов (знак минус изменяет направление);</p> <p>A+число — задает ускорение;</p> <p>символ E — завершает ввод команды и запускает цикл исполнения введенной команды.</p> <p>Пример: S3000M-6000A4000E. Здесь задаются максимальная скорость 3000 шагов/сек, количество шагов 6000 в инверсном направлении, ускорение 4000 шагов/сек²</p>	Глава 3

listing_4_1	<p>Файл listing_4_1.ino соответствует листингу 4.1 — тест вращения колес.</p> <p>Проводится тестирование правильности сборки робота путем генерации шагов с малой скоростью на оба колеса робота. При этом задействуются 5 контактов GPIO (15, 4, 13, 16, 2):</p> <ul style="list-style-type: none"> STEP_L 15 — генерация шагов левого колеса; STEP_R 4 — генерация шагов правого колеса; DIR_L 13 — направление вращения левого колеса; DIR_R 16 — направление вращения правого колеса; STEPPER_EN 2 — включение обоих моторов. 	Глава 4
listing_4_2	<p>Базовая программа управления роботом командами, поступающими по Bluetooth. Программа состоит из четырех файлов:</p> <ul style="list-style-type: none"> listing_4_2.ino — головной файл программы; motorstep.h — инициализация шаговых моторов; irq_robot.h — обработка прерываний, генерация шагов по таймеру; move_case.h — обработка команд, поступающих по Bluetooth. 	Глава 4
listing_5_1	<p>Файл listing_5_1.ino соответствует листингу 5.1 — пример работы с внутренней энергонезависимой памятью.</p> <p>Пример позволяет понять принципы работы с Flash-памятью, распаянной на плате контроллера ESP32</p>	Глава 5
listing_5_1b	<p>Файл listing_5_1b.ino соответствует листингу 5.1b — программа проверки работы кнопок.</p> <p>Этот простой пример считывания величины напряжения с контакта GPIO 32 демонстрирует изменение величины считываемого напряжения в зависимости от нажатия определенной кнопки</p>	Глава 5

listing_5_2	<p>Программа самостоятельного повторения роботом пройденного пути. Программа состоит из четырех файлов:</p> <ul style="list-style-type: none"> <code>listing_5_2.ino</code> — головной файл программы (в том числе работа с энергонезависимой памятью и обработка нажатий кнопок); <code>motorstep.h</code> — инициализация шаговых моторов; <code>irq_robot.h</code> — обработка прерываний, генерация шагов по таймеру, запись в массив шагов, сделанных роботом; <code>move_case.h</code> — обработка команд, поступающих по Bluetooth. 	Глава 5
listing_6_1	<p>Программа обхода препятствий и прохождения лабиринта. Программа состоит из четырех файлов:</p> <ul style="list-style-type: none"> <code>listing_6_1.ino</code> — головной файл программы (анализ наличия препятствий и генерация команд управления скоростью моторов); <code>motorstep.h</code> — инициализация шаговых моторов; <code>irq_robot.h</code> — обработка прерываний, генерация шагов по таймеру; <code>parallelonar.h</code> — работа с ультразвуковыми датчиками измерения расстояния от датчиков до препятствий. 	Глава 6
listing_7_1	<p>Программа управления роботом телеприсутствия (в качестве камеры на роботе — смартфон) — является адаптированной копией программы из листинга 4.1 с измененной системой обработки внешних команд. Программа состоит из четырех файлов:</p> <ul style="list-style-type: none"> <code>listing_7_1.ino</code> — головной файл программы; <code>motorstep.h</code> — инициализация шаговых моторов; <code>irq_robot.h</code> — обработка прерываний, генерация шагов по таймеру; <code>move_case.h</code> — обработка команд, поступающих по Bluetooth. 	Глава 7

listing_7_2	<p>Программа управления роботом телеприсутствия (в качестве камеры на роботе — камера ESP32-CAM). Робот принимает команды от UART-порта и формирует поступательные движения и повороты. Программа состоит из четырех файлов:</p> <ul style="list-style-type: none"> listing_7_2.ino — головной файл программы; motorstep.h — инициализация шаговых моторов; irq_robot.h — обработка прерываний, генерация шагов по таймеру; move_case.h — обработка команд, поступающих от UART-порта. 	Глава 7
CameraWeb-ServerEx	<p>Программа для загрузки в модуль ESP32-CAM (модернизированная программа управления камерой через сайт). Программа состоит из четырех файлов:</p> <ul style="list-style-type: none"> CameraWebServerEx.ino — головной файл программы (запуск камеры и веб-сервера); app_httpd.cpp — обработка видеосигнала, трансляция видео, поддержка страницы сайта камеры; camera_index.h — содержит массив со сжатым HTML-кодом страницы сайта камеры; camera_pins.h — содержит описание контактов GPIO для подключения различных модулей камеры к контроллерам ESP32. 	Глава 7

CameraRobot	<p>Программа для загрузки в модуль ESP32-CAM (модернизированная программа управления роботом через сайт). Программа состоит из четырех файлов:</p> <ul style="list-style-type: none"> CameraRobot.ino — головной файл программы (запуск камеры и веб-сервера); app_htpd.cpp — обработка видеосигнала, трансляция видео, поддержка страницы сайта камеры, передача команд управления роботом по UART; camera_index.h — содержит массив со сжатым HTML-кодом страницы сайта камеры; camera_pins.h — содержит описание контактов GPIO для подключения различных модулей камеры к контроллерам ESP32. <p>Дополнительно в папке в качестве примеров сохранены файлы, сформированные из массивов файла camera_index.h:</p> <ul style="list-style-type: none"> index_ov2640.html — распакованная страница сайта для модуля камеры ov2640; index_ov2640.html.gz — сжатая страница сайта для модуля камеры ov2640; index_ov2640shot.html — распакованная измененная страница сайта для модуля камеры ov2640; index_ov2640shot.html.gz — сжатая измененная страница сайта для модуля камеры ov2640; index_ov3660.html — распакованная страница сайта для модуля камеры ov3660. 	Глава 7
from_neco_apk	<p>Содержит файл video_robot_v_1.1.6.apk — установочный файл программы VideoRobot для смартфона на Android. Программа VideoRobot устанавливается на смартфон, который будет использоваться в качестве камеры робота</p>	Глава 7

listing_8_1	<p>Программа опрашивает модуль MPU-6050, измеряет и передает на последовательный порт компьютера рассчитанный с учетом комплементарного фильтра угол наклона робота относительно оси Y</p> <p>При использовании плоттера по последовательному порту (встроен в Arduino IDE) можно получить три графика: кривую изменения угла на основании только угловой скорости, на основании данных акселерометра, расчет комплементарного фильтра по данным акселерометра и гироскопа.</p> <p>Программа состоит из двух файлов:</p> <ul style="list-style-type: none"> <code>listing_8_1.ino</code> — головной файл программы (расчеты и передача данных); <code>gyro_acsel.h</code> — работа с MPU-5060, инициализация и опрос: получение данных об угловой скорости и величины ускорения. 	Глава 8
listing_8_2	<p>Программа подбора коэффициентов ПД-регулятора балансирующего робота. Программа состоит из пяти файлов:</p> <ul style="list-style-type: none"> <code>listing_8_2.ino</code> — головной файл программы (расчеты, управление скоростями колес); <code>defin.h</code> — описание переменных, задействованных в расчете балансировки робота; <code>gyro_acsel.h</code> — работа с MPU-5060, инициализация и опрос: получение данных об угловой скорости и величине ускорения; <code>motorstep.h</code> — инициализация шаговых моторов; <code>irq_robot.h</code> — обработка прерываний, генерация шагов по таймеру. 	Глава 8

listing_8_3	<p>Программа подбора коэффициентов ПИД-регулятора балансирующего робота (робот учится стоять на месте). Программа состоит из пяти файлов:</p> <ul style="list-style-type: none"> <code>listing_8_3.ino</code> — головной файл программы (расчеты, управление скоростями колес); <code>defin.h</code> — описание переменных, задействованных в расчете балансировки робота; <code>gyro_acsel.h</code> — работа с MPU-5060, инициализация и опрос: получение данных об угловой скорости и величине ускорения; <code>motorstep.h</code> — инициализация шаговых моторов; <code>irq_robot.h</code> — обработка прерываний, генерация шагов по таймеру. 	Глава 8
listing_8_4	<p>Программа управляемого движения балансирующего робота — дает возможность роботу не только самостоятельно балансировать, но и двигаться с заданной скоростью и совершать повороты. Более того, робот получил сервомотор с рычагом для самостоятельного подъема. Программа состоит из семи файлов:</p> <ul style="list-style-type: none"> <code>listing_8_4.ino</code> — головной файл программы (расчеты, управление скоростями колес); <code>defin.h</code> — описание переменных, задействованных в расчете балансировки робота; <code>gyro_acsel.h</code> — работа с MPU-5060, инициализация и опрос: получение данных об угловой скорости и величине ускорения; <code>motorstep.h</code> — инициализация шаговых моторов; <code>irq_robot.h</code> — обработка прерываний, генерация шагов по таймеру; <code>move_case.h</code> — обработка команд, поступающих по Bluetooth; <code>servo_hand.h</code> — управление сервомотором подъемного рычага. 	Глава 8
listing_8_4s4	<p>Робот, управляемый ранее с помощью листинга <code>listing_8_4</code>, получает ультразвуковой датчик, который позволяет ему не натыкаться на стены. см. описание файлов <code>listing_8_4</code>.</p> <p><code>irq_sonar.h</code> — измерение расстояния до объекта перед роботом</p>	Руково-дство

ESP32ServoTest	Тестовая программа для правильного позиционирования рычага подъема на серводвигателе.	Руководство
model robot	<p>Детали корпуса робота в формате STL для печати на 3D-принтере и их количество для сборки робота:</p> <p>борт.stl — 2 шт.;</p> <p>колесо.stl — 2 шт.;</p> <p>крепление аккумуляторов.stl — 1 шт.;</p> <p>крепеж поворотного колеса.stl — 1 шт.;</p> <p>палуба.stl — 1 шт.;</p> <p>проставка.stl — 1 шт.;</p> <p>рычажок.stl — 1 шт.;</p> <p>связка бортов.stl — 3 шт.;</p> <p>стойка палубы зад.stl — 2 шт.;</p> <p>стойка палубы перед.stl — 2 шт.;</p> <p>стойка палубы центр.stl — 1 шт.;</p> <p>фиксатор датчиков.stl — 1 шт.</p>	Глава 4–8

* Главы из книги М.Момота «Мобильные роботы на базе ESP32 в среде Arduino IDE», которая входит в набор.