

ОГАОУ ДПО «Белгородский институт развития»

Рассмотрено цикловой комиссией
«Автоматика и мехатронные комплексы»
протокол заседания № 17 от «29» июня 2019 г.
Председатель цикловой комиссии
_____ Недоступенко Д.А.

Методические указания
для выполнения лабораторных работ по изучению мобильных
роботизированных систем на базе мобильной платформы и робота –
манипулятора МПР-1.03 по ПМ 02. «Техническое обслуживание, ремонт и
испытание мехатронных систем»

специальность
15.02.10 «Мехатроника и мольная робототехника»

Недоступенко Дарья Александровна,
преподаватель ОГАПОУ
«Белгородский индустриальный колледж»

Белгород
2019

СОДЕРЖАНИЕ

Пояснительная записка.....	3
Лабораторная работа №1 Изучение конструкции и принципа работы мобильной роботизированной платформы «MRP-3».....	6
Лабораторная работа №2 Подключение к raspberry pi. Знакомство со средой разработки ПО для управления платформой.....	16
Лабораторная работа №3 Программирование простых алгоритмов по передвижению платформы.....	19
Лабораторная работа №4 Программирование алгоритмов получения информации от технологических датчиков платформы.....	23
Лабораторная работа №5 Получение снимков и видеопотока с камеры при помощи PICAMERA и OPENCV. Работа с терминалом.....	28
Лабораторная работа №6 Дистанционное управление платформой с мобильного устройства на платформе ANDROID. Дистанционное управление платформой при помощи ИК пульта.....	33
Лабораторная работа №7 Программирование алгоритмов движения платформы по заданной траектории.....	39
Лабораторная работа №8 Реализация алгоритмов самоориентации платформы с помощью вспомогательных датчиков.....	41
Приложение 1. Правила техники безопасности при работе обучающихся в лабораториях.....	43
Приложение 2. Критерии оценки выполнения лабораторной работы.....	45
Библиографический список.....	46

Пояснительная записка

Методические указания для выполнения лабораторных работ по изучению мобильных роботизированных систем на базе мобильной платформы и робота – манипулятора МПР-1.03 по ПМ 02. «Техническое обслуживание, ремонт и испытание мехатронных систем» предназначено для обучающихся 2 курса специальности 15.02.10 «Мехатроника и мобильная робототехника».

Методические указания составлены с учетом требований Федерального государственного образовательного стандарта среднего профессионального образования по специальности 15.02.10 «Мехатроника и мобильная робототехника» (далее – ФГОС СПО) [8], которые определяют актуальность представленных материалов, и на основании рабочей программы.

Разработка методических указаний обусловлена необходимостью учебно-методического обеспечения междисциплинарного курса МДК 02.01 «Техническое обслуживание, ремонт и испытание мехатронных систем» профессионального модуля ПМ 02. «Техническое обслуживание, ремонт и испытание мехатронных систем».

Целью представленных материалов является формирование общих (ОК 01-05, 10) и профессиональных (ПК 2.1-2.3) компетенций обучающихся при выполнении лабораторных работ.

Задачи методических указаний следующие:

- активизация мыслительной деятельности и познавательного интереса обучающихся;
- формирование умений применять полученные теоретические знания на практике, реализация единства теоретической и практической деятельности;
- выработка при решении поставленных задач профессионально значимых качеств: самостоятельность, точность, творческая инициатива.

Использование методических указаний способствует глубокому погружению обучающихся в процесс изучения мобильных роботизированных систем на базе мобильной платформы и робота-манипулятора МПР-1.03 в целом. Кроме того, работа с реальным оборудованием вызывает у обучающихся интерес и усиливает стремление к освоению специальных дисциплин.

В методических указаниях рассматривается мобильная робототехническая платформа МПР-1.03. Данный тип роботов создан для решения задач навигации, планирования, локализации. Изучение профессионального модуля ПМ.02 «Техническое обслуживание, ремонт и испытание мехатронных систем» имеет комплексный характер: базируется на знаниях и умениях, приобретенных обучающимися при изучении курсов электротехники, информатики, физики, математики и междисциплинарного

курса МДК 01.02 «Технология программирования мехатронных систем».

Комплекс на базе мобильной платформы и робота-манипулятора МПР-1.03 предназначен для:

- получения основных базовых знаний по направлению «Робототехника», изучения кинематики и принципа построения мобильных роботов;
- отработки навыков программирования,
- отладки и управления роботом,
- изучения устройства и способов управления самоходной роботизированной платформой.

Комплекс управляется контроллером на платформе Arduino с платой расширения и драйверами двигателей постоянного тока.

Разработка мобильных роботов является сложной задачей, связанной с требуемыми ресурсами и навыками во многих областях, такие как разработка программного обеспечения, искусственный интеллект, механическое проектирование, электротехника, сигнал обработки сенсорной техники или теории управления. Современные мобильные робототехнические системы часто являются монолитными. В настоящее время большое внимание уделяется разработке робототехнических комплексов с высокой приспособляемостью к движению по сложным траекториям и сложным поверхностям. Во всем мире ведутся разработки подобных аппаратов для реализации таких задач как, проведение работ по обезвреживанию взрывчатых устройств, локализация зоны чрезвычайной ситуации (ЧС), освоение космоса, ведение военных и разведывательных операций и т.д.

Принцип работы мобильной платформы основан на осевой передаче вращающего момента ведущим колесам (двум или четырем в зависимости от модификации изделия) через понижающие редуктора от двигателей постоянного тока.

Управление двигателями производится одноплатным микрокомпьютером Raspberry Pi через силовые драйверы на ключевых микросхемах L298N.

Структура представленных методических указаний представлена тремя блоками.

Первый блок, включает 8 лабораторных работ по изучению мобильной платформы и робота-манипулятора МПР-1.03. Лабораторные работы включают в себя следующие элементы: наименование темы, цель занятия, формируемые у обучающихся знания и умения, перечень формируемых общих и профессиональных компетенций, время выполнения работы, ход работы, порядок выполнения работы, краткие теоретические сведения, практическую часть (указания по выполнению), контрольные вопросы.

В теоретической части приводится структурированный материал, необходимый для подготовки обучающихся к лабораторным работам.

Практическая часть представляет собой проведение опытов и экспериментов. Вопросы к каждой лабораторной работе позволяют провести контроль обучающихся. На контрольные вопросы обучающиеся отвечают письменно в технических отчетах по лабораторным работам. Для проведения следующей лабораторной работы требуется сдача с устной защитой технического отчета по предыдущей работе.

Второй блок, включает в себя правила техники безопасности при работе обучающихся в лабораториях. Обучающиеся допускаются преподавателем к выполнению лабораторной работы только после инструктажа по технике безопасности работы.

Третий блок носит теоретический характер и содержит критерии оценки выполнения лабораторной работы.

Таким образом, представленные материалы имеют практическую значимость, способствуют формированию общих и профессиональных компетенций обучающихся и могут быть рекомендованы преподавателям профессиональных образовательных организаций технических специальностей, работающим по программам подготовки специалистов среднего звена по специальностям 27.02.04 Автоматические системы управления и 15.02.10 Мехатроника и мобильная робототехника.

ЛАБОРАТОРНАЯ РАБОТА №1

Тема: «Изучение конструкции и принципа работы мобильной роботизированной платформы «MRP-3»

Цель работы: изучить конструкцию мобильной платформы, ее технические характеристики, принцип работы и управления.

При выполнении лабораторной работы обучающийся должен:

знать: конструкцию мобильной платформы, технические характеристики мобильной платформы, принцип работы и управления мобильной платформой.

уметь: читать электросхемы; осуществлять технический контроль качества технического обслуживания; анализировать задачу и/или проблему и выделять её составные части.

Формируемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие;

ПК 2.1. Осуществлять техническое обслуживание компонентов и модулей мехатронных систем и мобильных робототехнических комплексов в соответствии с технической документацией.

Время выполнения работы: 3 часа 00 минут.

Ход работы:

1. Проведение преподавателем инструктажа по технике безопасности.
2. Изучить краткие теоретические сведения и законспектировать в технический отчет.
3. Последовательно реализовать порядок выполнения работы.
4. Письменно ответить на контрольные вопросы.

Краткие теоретические сведения

В данном лабораторном комплексе рассматривается мобильная робототехническая платформа МРР-1.03 (рисунок 1). Данный тип роботов создан для решения задач навигации, планирования, локализации.

Многие исследовательские институты, университеты и промышленность вкладывают много ресурсов и времени на исследования в этой области. Они пытаются привлечь студентов и исследователей к изучению в этой области.

Разработка мобильных роботов является сложной задачей, связанной с требуемыми ресурсами и навыками во многих областях, такие как разработка программного обеспечения, искусственный интеллект, механическое проектирование, электротехника, сигнал обработки сенсорной техники или теории управления. Современные мобильные робототехнические системы часто являются монолитными. В настоящее время большое внимание уделяется разработке робототехнических комплексов с высокой приспособляемостью к движению по сложным траекториям и сложным поверхностям. Во всем мире ведутся разработки подобных аппаратов для реализации таких задач как,

проведение работ по обезвреживанию взрывчатых устройств, локализация зоны чрезвычайной ситуации (ЧС), освоение космоса, ведение военных и разведывательных операций и т.д.

Принцип работы мобильной платформы основан на осевой передаче вращающего момента ведущим колесам (двум или четырем в зависимости от модификации изделия) через понижающие редуктора от двигателей постоянного тока.

Управление двигателями производится одноплатным микрокомпьютером Raspberry Pi через силовые драйверы на ключевых микросхемах L298N.

Обратная связь осуществляется посредством получения информации с различных сенсоров (датчик вибрации, датчик линии, ультразвуковой датчик дистанции, акселерометр, магнетометр и гироскоп).

Основные эксплуатационные характеристики

1. Одноплатный микрокомпьютер– Raspberry Pi.
2. Количество ведущих двигателей: 2 или 4 в зависимости от модификации платформы.
3. Тип ведущего двигателя: ДПТ с редуктором, 12в, 6000 об/мин.
4. Понижающий коэффициент редуктора: 1:56.
5. Драйвер управления двигателями: двухканальный на базе L298N.
6. Датчик расстояния: ультразвуковой HC-SR04.
7. Модуль Bluetooth: HC-06-3.
8. Датчик линии (аналоговый): Инфракрасный с 4-канальной платой усиления.
9. Датчик акселерометр: 3-х осевой GY-291 (ADXL345).
10. Электронный компас: 3-х осевой HMC5883L.
11. Датчик вибрации: SW-420.
12. Аккумулятор: Литий-полимерный 3-х секционный 11,1В 2200 мА-ч.



Рисунок 1- Внешний вид мобильной платформы

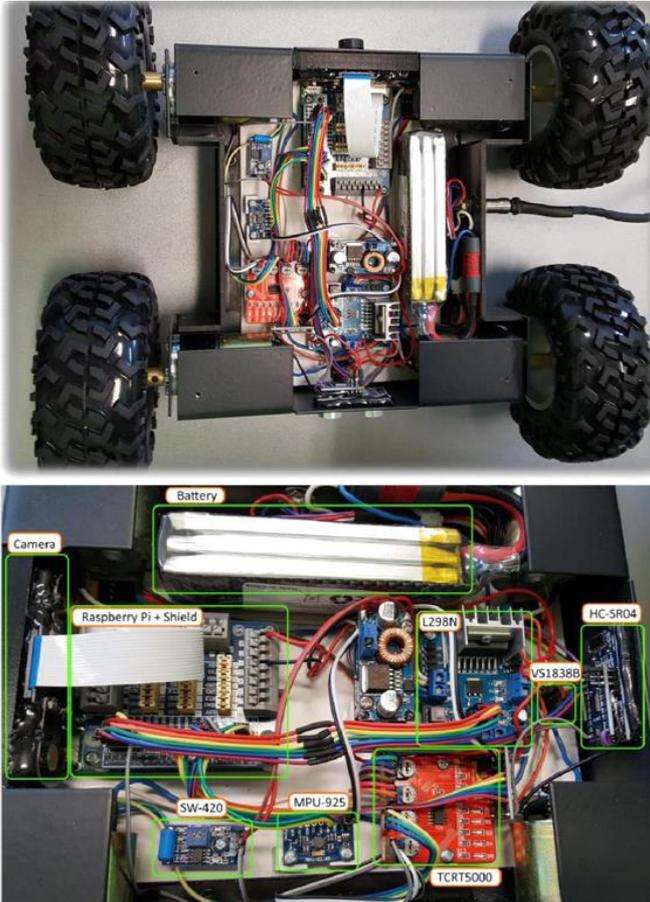


Рисунок 2- Вид элементов внутри корпуса платформы

На рисунке 3 представлена блок-схема основных элементов МРП, на рисунке 4 представлена принципиальная схема МРП.

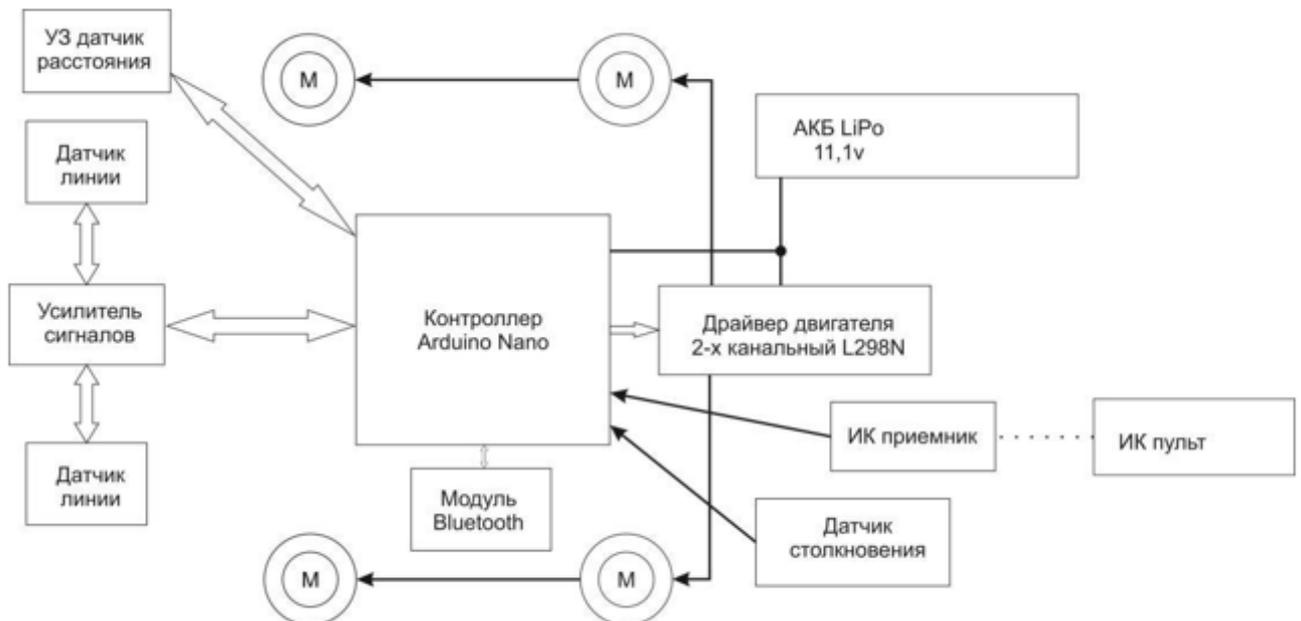


Рисунок 3- Блок-схема МРП

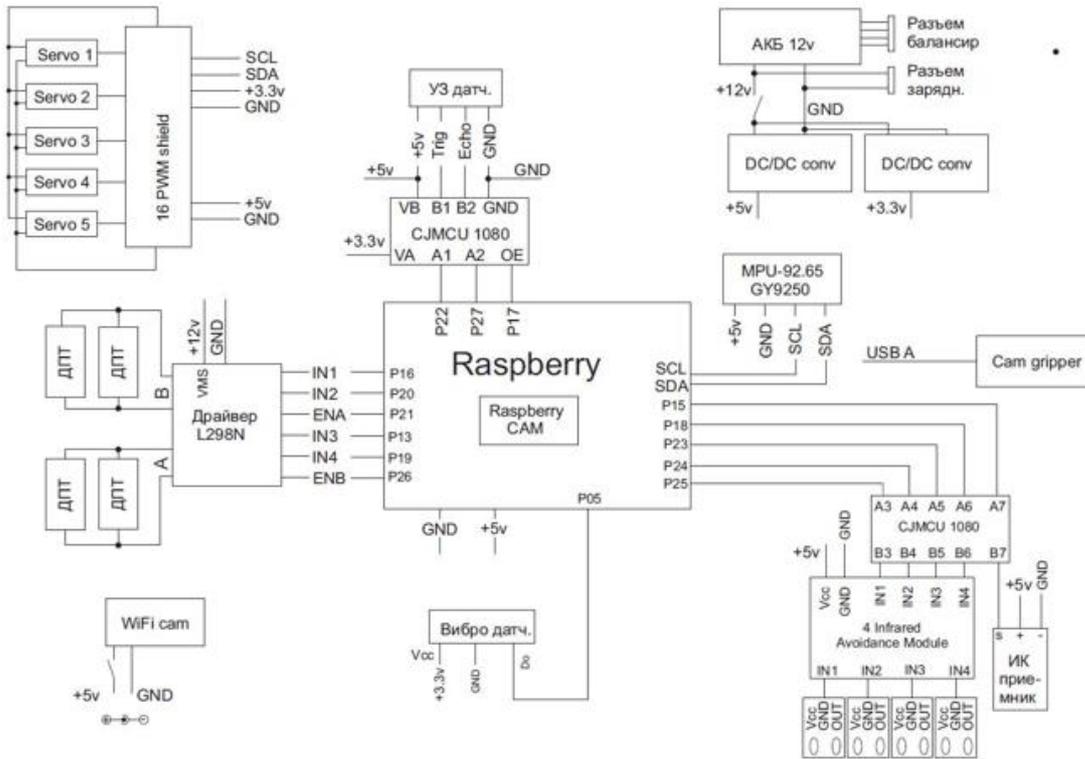


Рисунок 4- Принципиальная схема МРП

Ниже будут описаны характеристики и назначение основных элементов МРП. Взаимодействие с данными элементами на программном уровне будет рассмотрено в следующих лабораторных работах.

Raspberry Pi 3 model B

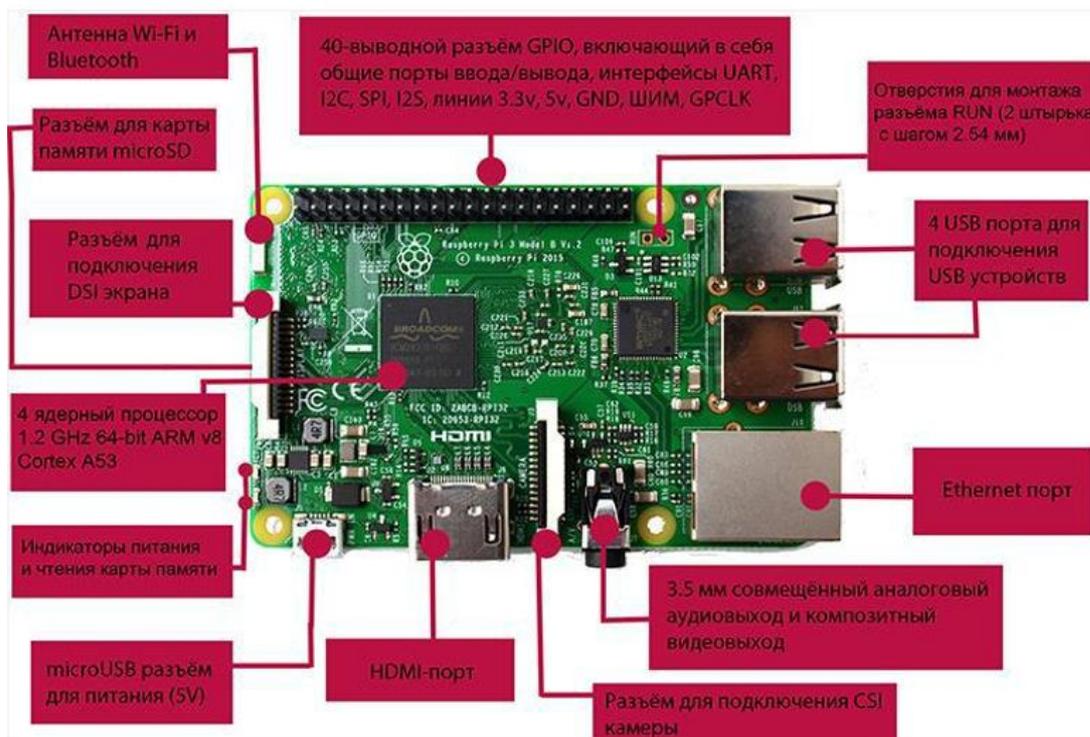


Рисунок 5- Raspberry Pi 3 model B

Характеристики:

- Процессор: 64-битный 4-ядерный ARM Cortex-A53 с тактовой частотой 1,2 ГГц на однокристальном чипе Broadcom BCM2837;
- Оперативная память: 1ГБ LPDDR2 SDRAM;
- Цифровой видеовыход: HDMI;
- Композитный выход: 3,5 мм (4 pin);
- USB порты: USB 2.0×4;
- Сеть: WiFi 802.11n, 10/100 Мб RJ45 Ethernet;
- Bluetooth: Bluetooth 4.1, Bluetooth Low Energy;
- Разъем дисплея: Display Serial Interface (DSI);
- Разъем видеоканеры: MIPI Camera Serial Interface (CSI-2);
- Карта памяти: MicroSD;
- Порты ввода-вывода: 40;
- Габариты: 85x56x17 мм.

Микрокомпьютер Raspberry Pi 3 model B миниатюрный и бесшумный компьютер, часто используемый в системах автоматизации, способный взаимодействовать с внешними компонентами с помощью системы ввода-вывода GPIO.

Сферы применения Raspberry Pi чрезвычайно широки и по большей степени ограничиваются фантазией пользователя. Ниже представлены примеры различных сфер использования Raspberry Pi:

- Домашний сервер для хранения данных;
- Робот под управлением WI-FI или с машинным зрением
- Игровая приставка или игровой автомат;
- Домашняя метеостанция;
- Охранная система с распознаванием лиц;
- «Умный» дом и многое другое.

В плату встроено четыре порта USB 2.0, HDMI, выход MIPI, microUSB, mini-jack, слот под карту microSD, которая заменяет встроенную память. Набор интерфейсов позволяет полноценно пользоваться собранной техникой и подключать к ней различные внешние устройства.

Raspberry Pi 3 model B основывается на чипе BCM2837. Его частота составляет 1.2 ГГц, используется 4 ядра. Дополняется система видеопроцессором Videocor 4.

ОЗУ основана на элементе типа SDRAM. Ее объем – 1 ГБ, чего будет достаточно для работы с несколькими задачами одновременно. Подключение к сети может осуществляться при помощи RJ-45 или Wi-Fi.

Модуль драйвера двигателей l298n

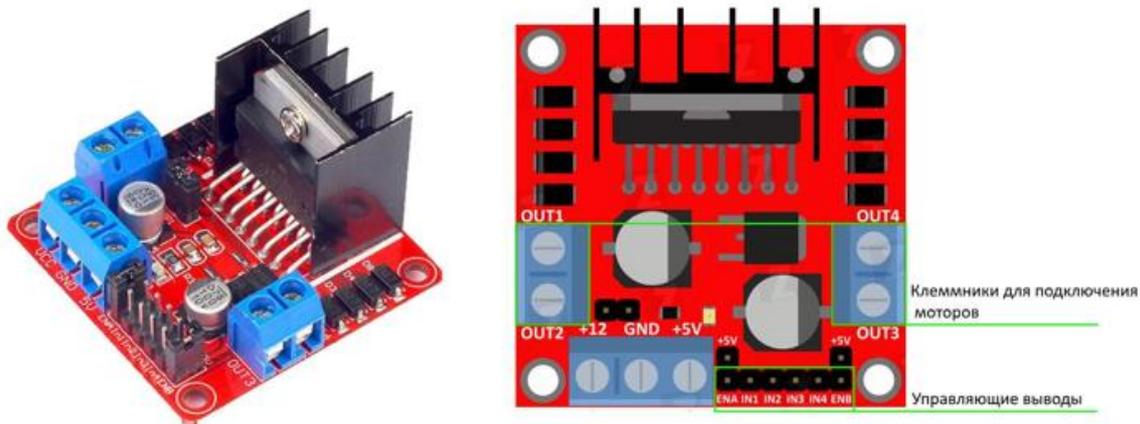


Рисунок 6- Модуль драйвера двигателей L298N

Характеристики:

- Входное напряжение логической части (V_d): от 6V до 12V;
- Входное напряжение управляемой части (V_s): от +5V до +35V;
- Рабочий ток внешней логической части (I_{ss}): 0 ~ 36mA;
- Ток нагрузки каждой управляемой части (I_o): 2A;
- Пиковый ток нагрузки каждой управляемой части (I_o): 3A;
- Максимальная потребляемая мощность: 20Вт (при температуре = 75°C);
- Высокий уровень входного управляющего сигнала (High): $2.3V = V_{in} = V_{ss}$;
- Низкий уровень входного управляющего сигнала (Low): $-0.3V = V_{in} = 1.5V$;
- Рабочая температура окружающей среды: от -25 до +130°C;
- Размеры модуля со стойками крепления и радиатором L298: 53 x 47 x 27 мм.

Модуль драйвера двигателей L298N позволяет управлять двигателями постоянного тока, либо шаговым двигателем с потребляемым током до 2 Ампер.

УЗ - датчик дистанции HC-SR04



Рисунок 7- УЗ - датчик дистанции HC-SR04

Характеристики:

- Напряжение питания: 5 В;
- Потребление в режиме тишины: 2 мА;
- Потребление при работе: 15 мА;

- Диапазон расстояний: 2–400 см;
- Эффективный угол наблюдения: 15°;
- Рабочий угол наблюдения: 30°.

УЗ датчик расстояния - бесконтактный направленный датчик HC-SR04. Используя ультразвуковые волны, он измеряет расстояние до объекта или просто обнаруживает препятствие на пути движения подвижной конструкции. На плате модуля размещены пьезоизлучатель ультразвука и воспринимающий отраженную волну микрофон. В отличие от инфракрасных дальномеров на ультразвуковой датчик HC-SR04 не влияют источники света или цвет препятствия. Могут возникнуть затруднения при определении расстояния до пушистых или тонких объектов. Обращаем внимание, что скорость звука в воздухе зависит от температуры. Это влияет на точность датчика. Типичными областями применения являются парковочные датчики, контроллеры уровня, устройства мониторинга местности и другие.

Датчик HC-SR04 отправляет ультразвуковые импульсы и слушает эхо. На вход Trig датчика подается импульс высокого уровня длительностью 10–15 микросекунд. Датчик отправляет ультразвуковой сигнал "chirp" из восьми коротких импульсов частотой выше предела диапазон слуха человека. Электроника датчика знает скорость звука в воздухе. Измеряя время между отправленным и принятым ультразвуком, ультразвуковой датчик HC-SR04 формирует выходной сигнал. Этот принцип эхолокации используют дельфины и летучие мыши. Спустя примерно микросекунду ультразвуковой датчик HC-SR04 выдает на выходе Echo импульс высокого уровня длительностью до 38 миллисекунд. Если препятствий не обнаружено, то на выходе будет сигнал с длительностью 38 мс. Таким образом, для работы с датчиком от электроники прибора требуется один цифровой управляющий выход и один вход для сигнала датчика.

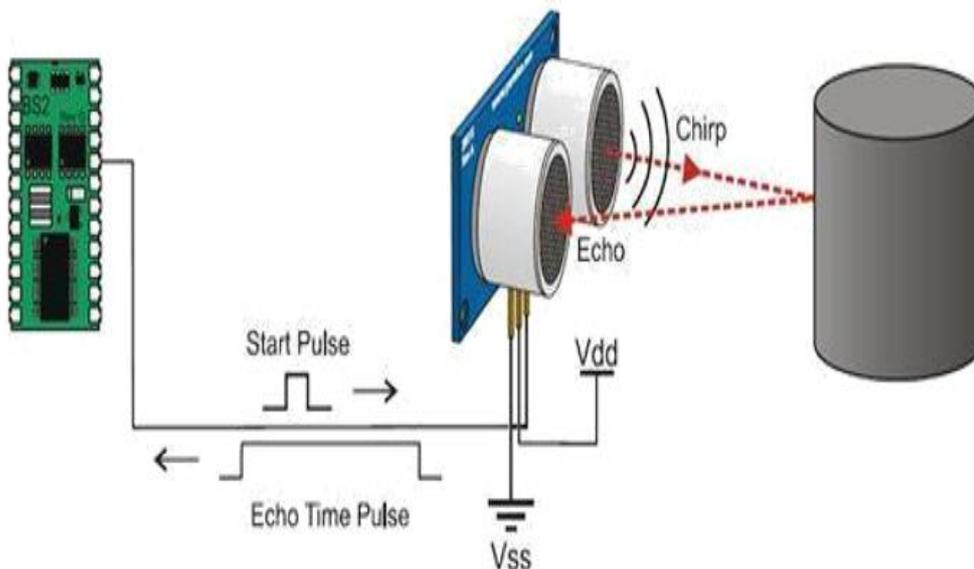


Рисунок 8- Принцип работы УЗ датчик дистанции HC-SR04

Датчик линии TCRT5000 4 канала



Рисунок 9- Датчик линии TCRT5000 4 канала

Характеристики:

- Напряжение питания: 5 В;
- Ток потребления: 10mA;
- Рабочее расстояние: от 1 мм до 20 см
- Диапазон температур: от -10° до +50°

Основой модуля является пара из ИК излучателя и ИК приемника "HS38BD" с рабочей частотой 38кГц. Чувствительность модуля регулируется с помощью потенциометров на компараторе "LM393". Принцип действия основан на определении цвета: черный цвет - на выходе датчика логическая "1", не черный - на выходе датчика логический "0". Расстояние, на котором работает TCRT5000 от 1 мм до 20 см. На плате присутствует светодиод, который включается при получении ИК-приемником сигнала.

Модуль GY-9250 на микросхеме MPU-925

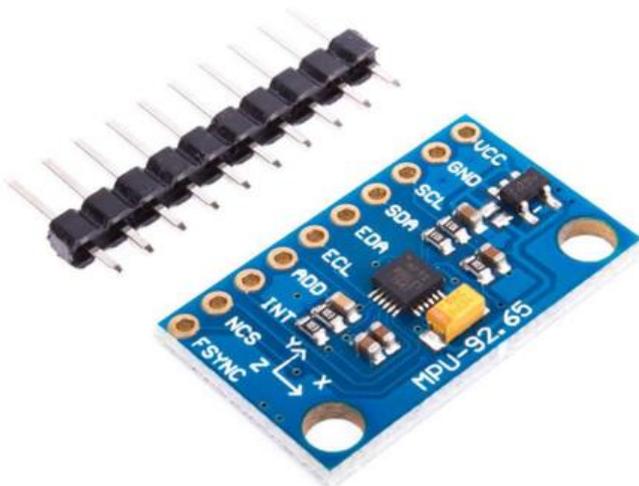


Рисунок 10- Модуль GY-9250 на микросхеме MPU 925

Характеристики:

- Микросхема: MPU9250
- Напряжение питания: 3 - 5 В
- Интерфейс: I2C (400кГц) / SPI (1 МГц)

- Диапазон измерения акселерометра: +/- 2G, +/- 4G, +/- 8G, +/- 16G
- Диапазон измерения гироскопа: +/- 250, +/- 500, +/- 1000, +/- 2000°/с
- Чувствительность гироскопа: 131, 65,5, 32,8, 16,4 LSB/°/с
- Диапазон измерения компаса магнитометра: +/- 4800 мкТл
- Буфер: FIFO 512В
- Рабочий ток: гироскоп – 3,2 мА, акселерометр – 450 мкА, магнитометр – 280 А
- Ток в режиме сна: 8+8 мкА (гироскоп+акселерометр).
- Габариты: 25 x 16 x 3 мм

Модуль GY-9250 на микросхеме MPU-9250 это 9-ти осевое устройство 3-х осевой гироскоп, 3-х осевой акселерометр и 3-х осевой магнитометр (магниторезистивный компас).

MPU-9250 – девятиосевой сенсор движения второго поколения компании InvenSense. Микросхема предназначена для применения в смартфонах, трекерах активности и прочих портативных потребительских изделиях. Добиться столь малых размеров микросхемы при высокой функциональности удалось с помощью патентованной технологии InvenSense CMOS MEMS, позволяющей изготавливать на одном кристалле МЭМС-устройства и КМОП-электронику.

В корпусе микросхемы MPU-9250 объединены два кристалла: на одном размещён трёхосевой гироскоп и трёхосевой сенсор, а на втором – трёхосевой магнитометр Asahi Kasei Microdevices (AK8963). Данные с сенсоров оцифровываются 16-битными АЦП, обрабатываются с применением алгоритмов Motion Fusion сигнальным процессором DMP (Digital Motion Processor) и передаются внешнему микроконтроллеру по шине I2C/SPI.

Алгоритмы MotionFusion обрабатывают все внутренние сенсоры и собирают полный набор данных.

Модуль цифрового датчика вибрации на БАЗЕ SW-420



Рисунок 11- Модуль цифрового датчика вибрации на базе SW-420

Характеристики:

- Напряжение питания: DC 3,3...5 В;
- Встроенный светодиодный индикатор;
- Размеры устройства: 38 x 15 x 10 мм;

Модуль цифрового датчика вибрации на базе SW-420. Выдает единицу при срабатывании, иначе ноль. Можно настроить как на срабатывание от мелкой вибрации, так и только на сильные удары, для регулировки на плате датчика размещен триммер.

Модуль инфракрасного приёмника VS1838B

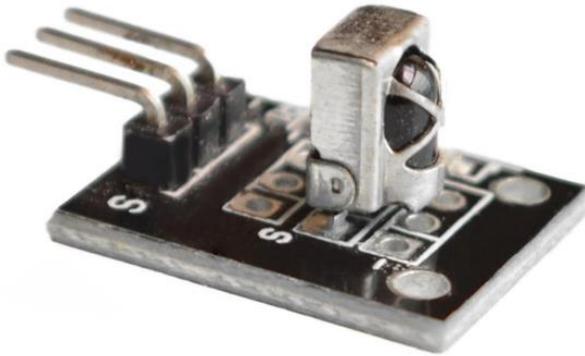


Рисунок 12- Модуль инфракрасного приёмника VS1838B

Характеристики:

- Потребляемый ток: 0,5 мА
- Рабочая частота: 38 КГц
- Расстояние приема сигнала: 20 м
- Эффективный угол приема сигнала: 90°
- Рабочее напряжение (приемник): 2,7 - 5,5 В
- Рабочее напряжение (модуль): 2,7 - 5,5 В

Модуль инфракрасного приёмника VS1838B предназначен для приема сигнала от пульта дистанционного управления бытовой техники, который передает команды посредством инфракрасного излучения. Воспринимающий элемент модуля – VS1838B. Сигнал модуля ориентирован на обработку микроконтроллером. При оснащении прибора управляемого дистанционно ИК приемником VS1838B можно использовать имеющиеся дома пульты от телевизора, кондиционера и т.д.

Контрольные вопросы

1. Из каких основных элементов состоит мобильная платформа MRP?
2. Какие функции выполняет микрокомпьютер Raspberry Pi в мобильной платформе?
3. Какие датчики обратной связи расположены в мобильной платформе? Расскажите о принципах их работы.
4. Исходя из изученного материала, предположите в каких областях можно применять данную платформу?

ЛАБОРАТОРНАЯ РАБОТА №2

Тема: «Подключение к Raspberry Pi. Знакомство со средой разработки ПО для управления платформой»

Цель работы: осуществить подключение к Raspberry Pi. Произвести знакомство со средой разработки ПО для управления платформой.

При выполнении лабораторной работы обучающийся должен:

знать: последовательность подключения к Raspberry Pi; классификацию и виды отказов оборудования; технологическую последовательность разборки, ремонта и сборки узлов и механизмов мехатронных систем.

уметь: читать электросхемы; кратко обосновывать и объяснить свои действия (текущие и планируемые); составлять план действия.

Формируемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие;

ПК 2.1. Осуществлять техническое обслуживание компонентов и модулей мехатронных систем и мобильных робототехнических комплексов в соответствии с технической документацией.

Время выполнения работы: 1 час 30 минут.

Ход работы:

1. Проведение преподавателем инструктажа по технике безопасности.
2. Изучить краткие теоретические сведения и законспектировать в технический отчет.
3. Последовательно реализовать порядок выполнения работы.
4. Письменно ответить на контрольные вопросы.

Краткие теоретические сведения

Подключиться к платформе Raspberry Pi можно при помощи HDMI кабеля, который непосредственно вставляется в соответствующий разъем на Raspberry Pi. После подачи питания на экране вашего монитора появится экран загрузки ОС (при таком подключении не забудьте выбрать необходимый источник на вашем мониторе). Однако HDMI разъем не всегда может быть физически доступен в виду конструкторских особенностей системы. В связи с этим рассмотрим подключение к Raspberry Pi при помощи VNC viewer.

Порядок выполнения работы

1. Скачайте и установите VNC viewer на ваш компьютер (режим доступа: <https://www.realvnc.com/en/connect/download/viewer/>)
2. Включите МРП и подождите окончания загрузки ОС Raspberry Pi (1-2 мин.)
3. Со своего компьютера подключитесь к wi-fi сети rasp-webgui, пароль: ChangeMe. IP-адрес будет автоматически присвоен ПК.

4. Откройте VNC viewer, нажмите File->New connection, укажите следующий IP адрес Raspberry Pi: 192.168.100.1. Нажмите дважды на созданное подключение, введите логин с паролем: login: pi, pass: raspberry (рисунок 13).

5. Если все шаги выполнены верно, то вы увидите рабочий стол ОС Raspberry Pi.

6. Нажмите на круглую кнопку с изображением малины  -> Programming - Python 2 (IDLE) (рисунок 14).

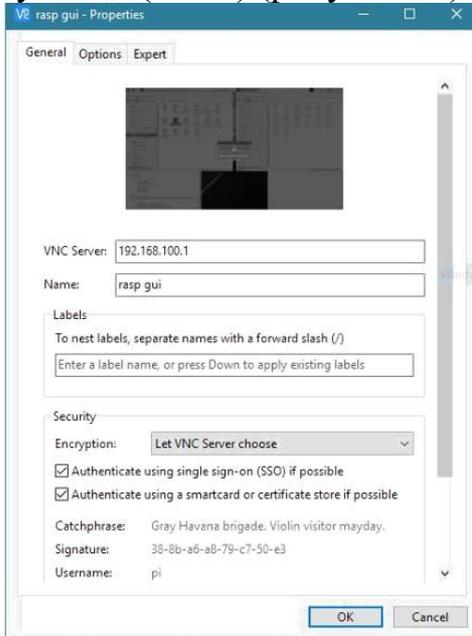


Рисунок 13- Свойства нового подключения



Рисунок 14- Python 2 (IDLE)

Все скрипты, которые вы будете писать в будущем, используют язык Python. Если вы не знакомы с данным языком программирования необходимую документацию можно найти на <https://www.python.org/>.

7. В качестве тестового скрипта напишите скрипт, который считает сумму чисел от 0 до 100 и выводит результат на экран.

Контрольные вопросы

1. Как осуществляется подключение к Raspberry Pi?
2. На какой ОС системе работает данная Raspberry Pi?
3. Найдите окно терминала и расскажите, зачем оно нужно.

ЛАБОРАТОРНАЯ РАБОТА №3

Тема: «Программирование простых алгоритмов по передвижению платформы»

Цель работы: написать скрипт для запуска ДПТ МРП.

При выполнении лабораторной работы обучающийся должен:

знать: последовательность подключения к Raspberry Pi; классификацию и виды отказов оборудования; основы программирования; алгоритмы по передвижению платформы.

уметь: понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые); понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы.

Формируемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам;

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

ПК 2.1. Осуществлять техническое обслуживание компонентов и модулей мехатронных систем и мобильных робототехнических комплексов в соответствии с технической документацией.

Время выполнения работы: 1 час 30 минут.

Ход работы:

1. Проведение преподавателем инструктажа по технике безопасности.
2. Изучить краткие теоретические сведения и законспектировать в технический отчет.
3. Последовательно реализовать порядок выполнения работы.
4. Письменно ответить на контрольные вопросы.

Краткие теоретические сведения

Для управления ДПТ МРП будем работать с модулем драйвера двигателей L298N. Как вы узнали из л.р. №1 данный модуль дает возможность управлять двигателями постоянного тока.

Выходы IN1, IN2 – задают направление вращения 1го мотора.

Выходы IN3, IN4 – задают направление вращения 2го мотора.

Направление вращения ротора двигателя управляется сигналами HIGH или LOW на каждый привод (или канал). Например, для первого мотора, HIGH на IN1 и LOW на IN2 обеспечит вращение в одном направлении, а LOW и HIGH заставит вращаться в противоположную сторону.

Выходы ENABLE A, B (ENA относится к IN1, IN2, ENB к IN3, IN4) отвечают за раздельное управление каналами. Могут использоваться в двух режимах:

Условно "пассивном" режиме, задав выводам "EN" высокий уровень (+5V). Для этого на плате, рядом с выводами ENA и ENB находятся пины соединенные с +5V. Замыкаются выводы с помощью джамперов. В данном

режиме нельзя регулировать скорость двигателей, они будут всегда вращаться на полную скорость.

Условно "активном" режиме, когда ими будет управлять контроллер - высокий логический уровень разрешает вращение моторов, низкий запрещает вне зависимости от состояния выводов "IN". Для регулировки скорости моторов, на "EN" выводы подается ШИМ (PWM) сигнал.

В данной лабораторной работе используется "активный" режим, физически с платой драйвера ничего делать не надо!

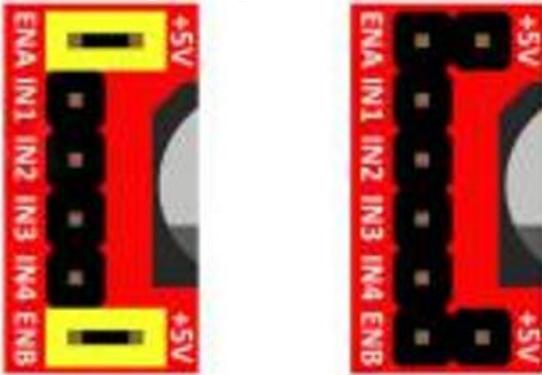


Рисунок 15- Конфигурация режимов работы драйвера

Порядок выполнения работы

1. Подключитесь к Raspberry Pi. Запустите Python 2.
2. Создайте новый скетч и импортируйте библиотеку GPIO **import RPi.GPIO as GPIO**

3. Задайте номера выводов драйвера в переменные

```
# setup
motors IN1 =
```

16

```
IN2 = 20
```

```
ENA =
```

21

```
# посмотрите номера в схеме лабораторной работы №1
```

```
IN3 =
```

```
? IN4
```

```
= ?
```

```
ENB =
```

```
?
```

4. Задайте режим нумерации выводов на Raspberry Pi и установите состояние выводов в режим записи.

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(IN1,GPIO.OUT)
```

```
GPIO.setup(IN2,GPIO.OUT)
```

```
GPIO.setup(IN3,GPIO.OUT)
```

```
GPIO.setup(IN4,GPIO.OUT)
```

```
GPIO.setup(ENA,GPIO.OUT)
```

```
GPIO.setup(ENB,GPIO.OUT)
```

5. Укажите частоту

ШИМ. p1 =

```
GPIO.PWM(ENA,50)
```

```
p2 = GPIO.PWM(ENB,50)
```

6. Напишите функции для движения МРП прямо *MotorAB_Direction_Forward(x,y)*, назад *MotorAB_Direction_Backward(x,y)*, разворот на месте по часовой стрелке *MotorAB_Direction_Right(x,y)*, разворот на месте против часовой стрелки *MotorAB_Direction_Left(x,y)*, остановка *MotorAB_Brake()*.

В качестве примера функции *MotorAB_Brake()* и *MotorAB_Direction_Forward(x,y)* для вас уже написаны.

```
def MotorAB_Brake(): global
```

```
IN1,IN2,IN3,IN4,p1,p2
```

```
GPIO.output(IN1, GPIO.HIGH)
```

```
GPIO.output(IN2, GPIO.HIGH)
```

```
GPIO.output(IN3, GPIO.HIGH)
```

```
GPIO.output(IN4, GPIO.HIGH)
```

```
p1.start(0)
```

```
p2.start(0)
```

```
def
```

```
MotorAB_Direction_Forward(x,y)
```

```
: try:
```

```
global IN1,IN2,IN3,IN4,p1,p2
```

```
GPIO.output(IN1, GPIO.HIGH)
```

```
GPIO.output(IN2, GPIO.LOW)
```

```
GPIO.output(IN3, GPIO.HIGH)
```

```
GPIO.output(IN4, GPIO.LOW)
```

```
p1.start(x)
```

```
p2.start(y)
```

```
except:
```

```
print x,y
```

7. Импортируйте библиотеку `time` **import time** и используйте ее для запуска написанных функций на определенный промежуток времени

```
# ждем 1 сек
```

```
time.sleep(1)
```

8. Прокатите тележку вперед – 2 сек, назад – 2 сек., разворот на месте по часовой стрелке на 360° , *разворот на месте против часовой стрелки на 360° .*

Контрольные вопросы

1. В каких режимах можно управлять ДПТ при помощи драйвера двигателей L298N?
2. Что такое ШИМ, скважность? Для чего используется ШИМ?
3. Как при движении МРП вперед / назад сделать плавный поворот влево/вправо?

ЛАБОРАТОРНАЯ РАБОТА №4

Тема: «Программирование алгоритмов получения информации от технологических датчиков платформы»

Цель работы: научить считывать информацию с датчиков.

При выполнении лабораторной работы обучающийся должен:

знать: основы программирования; алгоритмы получения информации при программировании; основные характеристики датчиков платформы.

уметь: понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые); понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы.

Формируемые компетенции:

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие;

ПК 2.3. Производить замену и ремонт компонентов и модулей мехатронных систем и мобильных робототехнических комплексов в соответствии с технической документацией.

Время выполнения работы: 1 час 30 минут.

Ход работы:

1. Проведение преподавателем инструктажа по технике безопасности.
2. Последовательно реализовать порядок выполнения работы с занесением данных в технический отчет.
3. Письменно ответить на контрольные вопросы.

Порядок выполнения работы

1. Подключение УЗ датчик дистанции HC-SR04

Укажите номера пинов УЗ датчика TRIGGER в переменную GPIO_TRIGGER и ECHO в переменную GPIO_ECHO (номера пинов указаны в схеме лабораторной работы №1). Установите пин GPIO_TRIGGER в состояние записи (GPIO.OUT), а пин GPIO_ECHO в состояние чтения (GPIO.IN).

Скопируйте и изучите функцию получения расстояния с датчика.

```
def distance(GPIO_TRIGGER, GPIO_ECHO):
```

```
# взводим пин GPIO_TRIGGER
GPIO.output(GPIO_TRIGGER,
True)
```

```
# устанавливаем пин GPIO_TRIGGER в 0 через
0.01ms time.sleep(0.00001)
GPIO.output(GPIO_TRIGGER, False)
```

```
StartTime = time.time()
```

```
StopTime = time.time()
```

```
# указываем начальное время while  
GPIO.input(GPIO_ECHO) == 0:
```

```
StartTime = time.time()
```

```
# указываем конечное время while  
GPIO.input(GPIO_ECHO) == 1:
```

```
StopTime = time.time()
```

```
# находим разницу
```

```
TimeElapsed = StopTime - StartTime
```

```
# перемножаем со скоростью звука (34300 см/с)
```

```
# и делим на 2, потому, что звук идет до предмета и обратно
```

```
distance = (TimeElapsed * 34300) / 2
```

```
return distance
```

Выведите значения расстояния на экран в бесконечном цикле с частотой 2 Гц.
2. Подключение датчика линии TCRT5000

Укажите номера информационных пинов датчика линии TCRT5000 и задайте им необходимое состояние.

```
# color
```

```
sensors
```

```
TrackingPin1 =
```

```
? TrackingPin2
```

```
= ?
```

```
TrackingPin3 =
```

```
? TrackingPin4
```

```
= ?
```

```
GPIO.setup(TrackingPin1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

```
GPIO.setup(TrackingPin2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

```
GPIO.setup(TrackingPin3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

```
GPIO.setup(TrackingPin4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

Для чтения состояния датчика используйте следующий код:

```
if GPIO.input(TrackingPin1) == GPIO.HIGH: c1 = 1
```

```
if GPIO.input(TrackingPin2) == GPIO.HIGH: c2 = 1
```

```
if GPIO.input(TrackingPin3) == GPIO.HIGH: c3 = 1
```

```
if GPIO.input(TrackingPin4) == GPIO.HIGH: c4 = 1
```

Выведите состояние датчиков на экран в бесконечном цикле с частотой 2 Гц. При необходимости чувствительность модуля регулируется с помощью потенциометров.

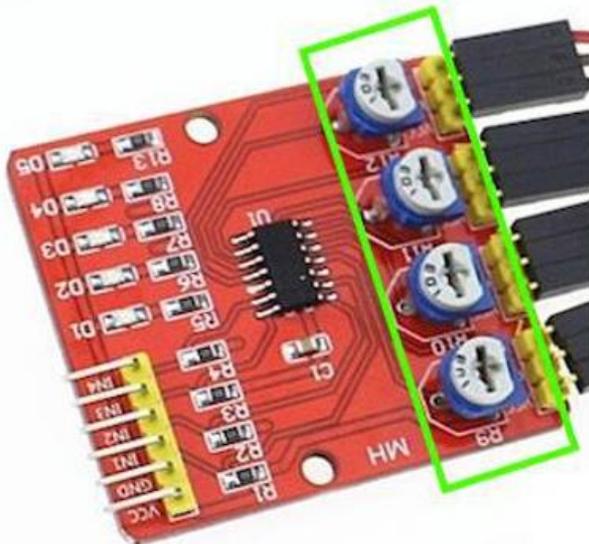


Рисунок 16- Регулировочные потенциометры

2. Подключение модуля GY-9250

Скопируйте файлы библиотеки из папки `mpu9250` и подключите их:
from mpu9250 import *

Создайте функцию инициализации датчика

```
# Инициализация акселерометра, магнетометра и гироскопа  
(GY9250) def setupGY9250():
```

```
global mpu, compas  
mpu = MPU9250()  
mpu.initialize()
```

```
compas = AK8963()
```

```
compas.initialize()
```

Скопируйте код и изучите как происходит чтение данных с датчика.

```
# Съем данных с GY9250 gyro_data =  
mpu.get_gyro() accel_data = mpu.get_accel()  
compas_data = compas.get_calibrated() roll =  
mpu.get_x_rotation(mpu.get_accel()) pitch =  
mpu.get_y_rotation(mpu.get_accel()) if  
roll>180: roll=roll-360
```

```
heading = compas.heading()
```

```
# Данные акселерометра
```

```
accDataOut =
'ax'+str(round(accel_data['x'],4))+',ay'+str(round(accel_data['y'],4))+',az'+str(round(accel_data['z'],4))+';'
```

```
# Данные магнетометра magDataOut =
'mx'+str(round(compas_data['x'],2))+',my'+str(round(compas_data['y'],2))+',mz'+str(round(compas_data['z'],2))+';'
```

```
# Данные гироскопа gyrDataOut =
'gx'+str(round(gyro_data['x'],2))+',gy'+str(round(gyro_data['y'],2))+',gz'+str(round(gyro_data['z'],2))+';'
```

```
# Данные крена, тангажа и рысканья
```

```
pryDataOut =
',pi'+str(round(pitch,2))+',ro'+str(round(roll,2))+',ya'+str(round(heading,2))+';'
```

Выведите состояние датчиков на экран в бесконечном цикле с частотой 2 Гц.

3. Подключение модуля цифрового датчика вибрации

Инициализируйте датчик и создайте callback при наличии вибрации

```
# setup sw240 tilt
sensor tiltPin = 5
```

```
GPIO.setup(tiltPin, GPIO.IN,
pull_up_down=GPIO.PUD_UP) tilt = 0
```

```
GPIO.add_event_detect(tiltPin, GPIO.FALLING, callback=alert, bouncetime=100)
```

```
def alert(ev=None):
global tilt
```

```
tilt = 1
```

Считайте данные после чего обнулите переменную вибрации.

```
print tilt
tilt = 0
```

Выведите состояние датчиков на экран в бесконечном цикле с частотой 2 Гц. При необходимости чувствительность модуля регулируется с помощью потенциометра (рисунок 17).

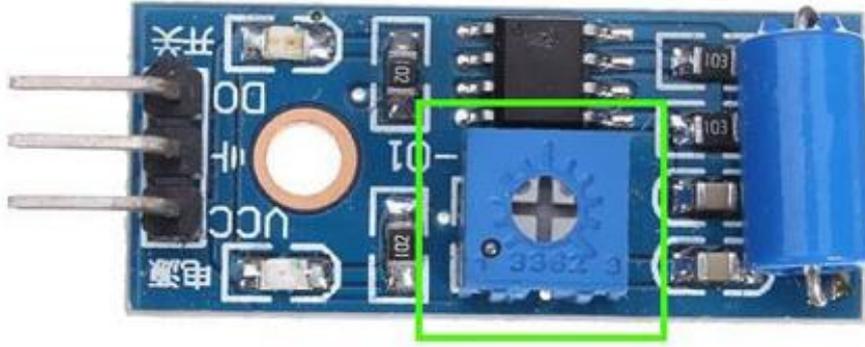


Рисунок 17- Регулировочный потенциометр

4. Объедините скрипты в один и выведите полученную информацию с датчиков в удобном для пользователя виде

Контрольные вопросы

1. Какие основные шаги можно выделить при подключении датчиков?
2. Измерьте примерную погрешность датчика дистанции и значений крена и тангажа. Чем они могут быть обусловлены? Как их можно ликвидировать?
3. Какие датчики вы бы добавили в систему и для каких целей?

ЛАБОРАТОРНАЯ РАБОТА №5

Тема: «Получение снимков и видеопотока с камеры при помощи PICAMERA и OPENCV. Работа с терминалом»

Цель работы: получить снимки и видеопоток с камеры при помощи Pi Camera Open CV. Научиться скачивать и устанавливать модули при помощи терминала.

При выполнении лабораторной работы обучающийся должен:

Знать: методы получения видеопотоков с камеры; методы получения снимков с камеры при помощи PICAMERA И OPENCV.

Уметь: работать с терминалом; получать снимки и видеопоток с камеры.

Формируемые компетенции:

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами;

ПК 2.1. Осуществлять техническое обслуживание компонентов и модулей мехатронных систем и мобильных робототехнических комплексов в соответствии с технической документацией.

Время выполнения работы: 1 час 30 минут.

Ход работы:

1. Проведение преподавателем инструктажа по технике безопасности.
2. Последовательно реализовать порядок выполнения работы с занесением данных в технический отчет.
3. Письменно ответить на контрольные вопросы.

Порядок выполнения работы

1. Подключение камеры к Raspberry Pi.

Для начала работы необходимо правильно подключить камеру к Raspberry Pi. Для этого один конец AWM ленты подключается к камере, другой конец в специально отведенный порт на плате (рисунок 18). Рядом с данным портом есть опознавательная надпись «CAMERA» (не спутать с «DISPLAY» портом).



Рисунок 18- Подключение камеры к плате

2. Активация модуля камеры.

Для дальнейшей работы необходимо активировать модуль камеры на плате. Для этого нужно напечатать в терминале «`sudo raspi-config`» и нажать Enter. После этого появится окно настроек конфигураций (рисунок 19). Выбираем 5 пункт Interfacing Options.

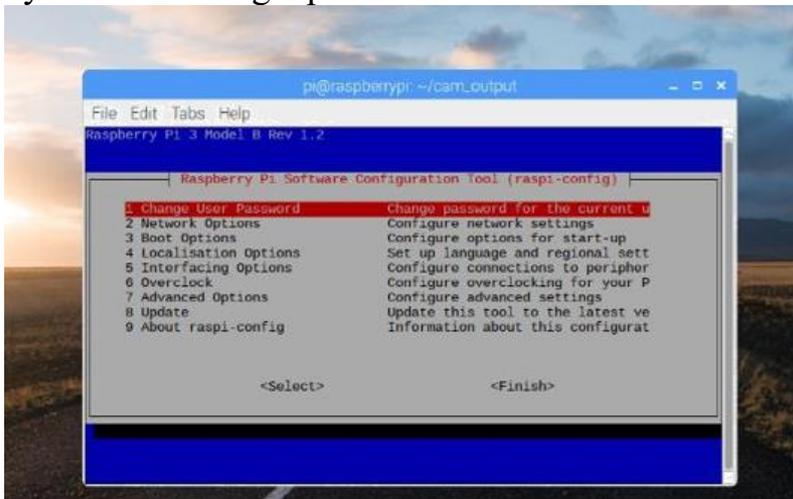


Рисунок 19- Настройка конфигураций

Далее необходимо активировать камеру (рисунок 20). Для этого нужно выбрать первый пункт с названием «Camera» и активировать его, после чего перезагрузить систему (скорее всего перезагрузка будет предложена автоматически).

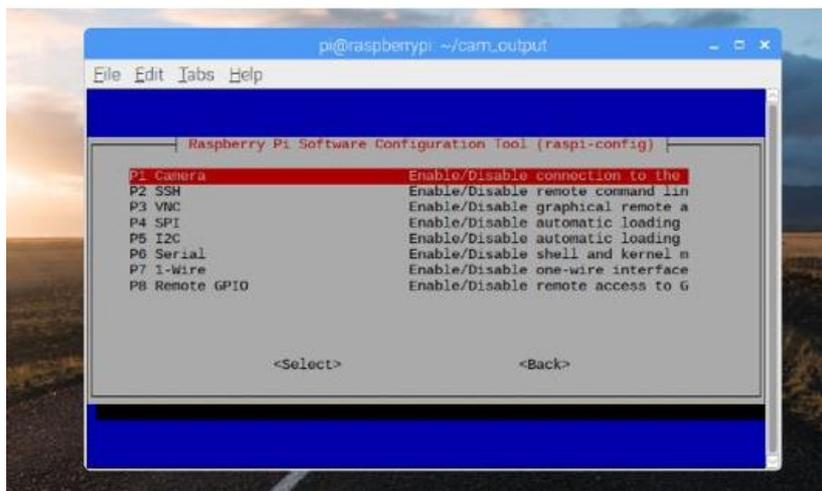


Рисунок 20- Меню для активации камеры

3. Тест модуля камеры

Следующим шагом необходимо протестировать взаимодействие платы с камерой, что станет подтверждением того, что все было подключено верно. Для этого в терминале необходимо прописать команду «`raspistill -o output.jpg`». Данная команда активирует камеру, делает фотографию и сохраняет полученный результат с расширением `jpg` в директории, которая является рабочей в терминале в данный момент.

4. Установка `picamera`.

Для взаимодействия камеры через Python сначала необходимо провести подготовительную работу и установить все необходимые пакеты. Один из них - `PiCamera`, который делает возможным взаимодействие с интерфейсом камеры. Для этого вводим в терминал команду «`pip install `picamera[array]``».

Важно заметить, что необходимо писать не просто `picamera`, а именно с добавлением `array` в квадратных скобках. Все получаемые изображения являются массивами `NumPy`.

Далее скачайте `OpenCV`. Для этого в терминале необходимо прописать «`sudo apt-get install python-opencv`».

5. Получение снимка, используя Python и `OpenCV`.

Создаем новый файл с расширением `.py`. К примеру, `image.py` и вставляем туда следующий код:

```
# импортирует необходимые
пакеты
from picamera.array import
PiRGBArray
from picamera import
PiCamera
import time

import cv2

# инициализируем камеру и ссылку на захват необработанной
камеры
camera = PiCamera()
rawCapture = PiRGBArray(camera)

# выделяем время для нагрева камеры
```

```
time.sleep(0.1)
```

```
# захватываем изображение через
камеру camera.capture(rawCapture,
format="bgr") image = rawCapture.array
```

```
# выводим полученное изображение на
экран cv2.imshow("Image", image)
cv2.waitKey(0)
```

После чего сохраняем полученный код и запускаем с помощью команды «python image.py».

6. Получение потокового видео с камеры (стрима), используя Python и OpenCV.

Следующим шагом является получение стрима видео с камеры и вывод его на экран. Для этого создаем новый файл с расширением py. К примеру, video.py и вставляем в него следующий код:

```
# импортируем необходимые
пакеты from picamera.array import
PiRGBArray from picamera import
PiCamera import time
```

```
import cv2
```

```
# инициализируем камеру и ссылку на захват необработанной
камеры camera = PiCamera()
```

```
camera.resolution = (640, 480)
camera.framerate = 32
```

```
rawCapture = PiRGBArray(camera, size=(640, 480))
```

```
# выделяем время для нагрева камеры
```

```
time.sleep(0.1)
```

```
# захватываем кадры, получаемые с камеры
```

```
for frame in camera.capture_continuous(rawCapture, format="bgr",
use_video_port=True):
```

```
# grab the raw NumPy array representing the image, then initialize the
timestamp
```

```
# and occupied/unoccupied text
image = frame.array
```

```
# выводим кадры
cv2.imshow("Frame", image)
key = cv2.waitKey(1) & 0xFF

# отчищаем поток для подготовки следующего
кадра rawCapture.truncate(0)

# при нажатии клавиши 'q', цикл и работа камеры будут
остановлены if key == ord("q"):

break
```

Сохраняем все изменения и запускаем написанный скрипт.

Контрольные вопросы

1. Опишите основные шаги подключения камеры.
2. Какие пакеты необходимо импортировать для работы с камерой и получения потокового видео?
3. Для чего нужен аргумент в команде «`pip install `picamera[array]``».
4. Назовите 5 задач, в которых необходимо использования камеру МРП.

ЛАБОРАТОРНАЯ РАБОТА №6

Тема: «Дистанционное управление платформой с мобильного устройства на платформе ANDROID. Дистанционное управление платформой при помощи ИК пульта»

Цель работы: Научить управлять МРП и получать данные с сенсоров при помощи мобильного устройства на основе Android. Научить управлять МРП при помощи ИК пульта.

При выполнении лабораторной работы обучающийся должен:

знать: основы дистанционного управления платформой с мобильного устройства на платформе ANDROID; основы дистанционного управления платформой при помощи ИК-пульта.

уметь: дистанционно управлять платформой с мобильного устройства на платформе ANDROID; дистанционно управлять платформой при помощи ИК-пульта.

Формируемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам;

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

ПК 2.1. Осуществлять техническое обслуживание компонентов и модулей мехатронных систем и мобильных робототехнических комплексов в соответствии с технической документацией.

Время выполнения работы: 1 час 30 минут.

Ход работы:

1. Проведение преподавателем инструктажа по технике безопасности.
2. Последовательно реализовать порядок выполнения работы с занесением данных в технический отчет.
3. Письменно ответить на контрольные вопросы.

Порядок выполнения работы

1. Установить и запустить приложение из папки Android на мобильное устройство семейства android.
2. Запустить на Raspberry скрипт main.py, который находится в директории /home/pi/control_sketch (при каждой загрузке ОС этот скрипт запускается автоматически).
3. Нажать кнопку «Поиск устройств». Выбрать из списка «Arduino-Bluetooth» (пароль по умолчанию 0000 или 1234).
4. При подключении появится статус «Связь установлена». При помощи оранжевого колеса можно управлять движением тележки. В нижней части экрана пишется информация, полученная с датчиков.

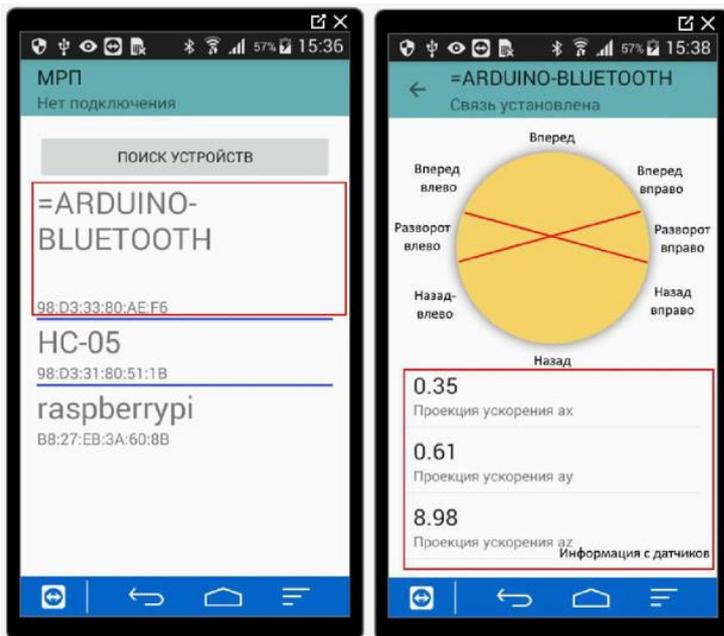


Рисунок 21-Экраны android приложения

5. Внимательно изучите код в main.py и выделите команды, отвечающие за прием и передачу данных по Bluetooth.
6. Скачайте бесплатное приложение на Android “Bluetooth Terminal” <https://play.google.com/store/apps/details?id=Qwerty.BluetoothTerminal&hl=ru>
7. Напишите свой скрипт, который будет отсылать произвольную строку на ваш мобильный телефон и ждать ответа от телефона. Ответом будет служить строка, полученная с МРП. Если МРП получило такую же строку, что и отсылала, то вышлите обратно на телефон слово “True”, иначе “False”. Осуществим дистанционное управление МРП при помощи ИК пульта. ИК приемник VS1838B должен быть подключен как указано на рисунке 22.



Рисунок 22- Выводы VS1838B 9. Установка необходимых пакетов и библиотек

Обновите apt-get до последней версии. Для этого в терминале необходимо указать команду «\$ sudo apt-get update». После чего установите пакет LIRC (Linux Infrared Remote Control), который отвечает за работу ИК приемника «\$ sudo apt-get install lirc».

10. Настройка конфигураций для начала работы. Добавьте в файл «/etc/modules» строки: lirc_dev

```
lirc_rpi gpio_in_pin=15
```

Возможно, потребуется права администратора и работа черед root. Для этого введите в терминал «sudo -i».

После чего добавьте следующие строки в файл «/etc/lirc/hardware.conf»:

```
LIRCD_ARGS="--uinput --listen"
```

```
LOAD_MODULES=true
```

```
DRIVER="default"
```

```
DEVICE="/dev/lirc0"
```

```
MODULES="lirc_rpi"
```

Если файла hardware.conf нет в директории, создайте его самостоятельно.

Добавьте следующие строки в «/boot/config.txt»: dtoverlay=lirc-rpi,gpio_in_pin=15

Обновите следующие строки в «/etc/lirc/lirc_options.conf»:

```
driver = default
```

```
device = /dev/lirc0
```

После чего перезапустите lircd с помощью команд:

```
$ sudo /etc/init.d/lircd stop
```

```
$ sudo /etc/init.d/lircd start
```

Проверьте статус:

```
$ sudo /etc/init.d/lircd status
```

Перезапустите Raspberry:

```
$ reboot
```

10. Проверка работы:

Напишите в терминале: \$

```
sudo /etc/init.d/lircd stop
```

```
$ mode2 -d /dev/lirc0
```

Нажмите на пульте любые кнопки. В ответ вы должны получить строки следующего формата:

```
pulse 560
```

```
space 1706
```

```
pulse 535
```

```
...
```

11. Запись файла конфигураций для отдельных кнопок на пульте. Напишите в терминале команды:

```
$ sudo /etc/init.d/lircd stop
$ sudo irrecord -d /dev/lirc0 ~/lircd.conf
```

Следуйте инструкциям в терминале. В результате будет создан файл с расширением «.lircd.conf».

Сохраните копию оригинального lircd.conf файла и замените его:

```
$ sudo mv /etc/lirc/lircd.conf /etc/lirc/lircd_original.conf
```

```
$ sudo cp ~/lircd.conf /etc/lirc/lircd.conf
$ sudo /etc/init.d/lircd start
```

В созданном .lircd.conf файле необходимо убрать второй столбец у каждой записанной кнопки. К примеру, в случае как на рисунке 23 необходимо у каждой кнопки убрать «0x7E825B6C».

```
begin codes
KEY_POWER          0x219E48B7 0x7E825B6C
KEY_HOME           0x219E609F 0x7E825B6C
KEY_UP             0x219EA05F 0x7E825B6C
KEY_LEFT           0x219EE01F 0x7E825B6C
KEY_ENTER          0x219E10EF 0x7E825B6C
KEY_RIGHT          0x219E906F 0x7E825B6C
KEY_DOWN           0x219E00FF 0x7E825B6C
KEY_BACK           0x219ED827 0x7E825B6C
KEY_STOP           0x219E20DF 0x7E825B6C
KEY_OPTION         0x219E58A7 0x7E825B6C
KEY_REWIND         0x219EF807 0x7E825B6C
KEY_PLAYPAUSE     0x219E50AF 0x7E825B6C
KEY_FASTFORWARD   0x219E7887 0x7E825B6C
KEY_PREVIOUS       0x219E40BF 0x7E825B6C
KEY_NEXT           0x219E807F 0x7E825B6C
KEY_SEARCH         0x219EF00F 0x7E825B6C
KEY_EJECTCD       0x219E08F7 0x7E825B6C
end codes
```

Рисунок 23- Пример конфигурационного файла

После чего введите в терминал команду «irw» и нажмите на записанные кнопки пульта. В результате должен прийти ответ, с именем записанной кнопки.

13. Работа с /etc/lirc/irhex.lircrc файлом.

В файле /etc/lirc/irhex.lircrc записана обработка сигналов, которая происходит после каждого нажатия кнопки, например:

```
begin

prog = irhex
button =
KEY_1

config = echo "You pressed one"
repeat = 0
```

end

Чтобы запустить данный файл, необходимо ввести команду «irexec» и имя файла:

```
$ irexec irexec.lircrc
```

Нажмите на записанные кнопки пульта, в ответ должно прийти поле «config», соответствующей кнопки. Например при нажатии на клавишу KEY_1 придет ответ

«You pressed one».

14. Использование Python для приема данных с ИК пульта.

В начале, потребуется изменить /etc/lirc/irexec.lircrc файл следующий образом (если использовать предыдущий пример):

```
begin
```

```
prog = irexec
button =
KEY_1
config = echo "You pressed one"
repeat = 0
end
```

```
begin
```

```
prog = myprogram
```

```
button = KEY_1
```

```
config = one
```

```
end
```

После чего установите пакет python-lirc:

```
$ sudo apt-get install python-lirc
```

```
$ git clone https://github.com/tompreston/python-lirc.git $
```

```
cd python-lirc/
```

```
$ make py3 && sudo python3 setup.py install
```

```
$ make py2 && sudo python setup.py install
```

Создайте python скрипт, вставьте следующий код:

```
import lirc
```

```
while True:
```

```
sockid = lirc.init("myprogram", "/etc/lirc/irexec.lircrc")
```

```
code = lirc.nextcode()  
print (code)
```

```
lirc.deinit()
```

Запустите скрипт, нажмите на кнопки пульта и, если все было выполнено верно, в ответ придет код нажатой кнопки.

15. Используя функции для движения МРП из лабораторной работы №3, напишите скрипт, который будет управлять движением МРП согласно нажатым клавишам на пульте.

Контрольные вопросы

1. Как управлять МРП при помощи телефона? Какой функционал предоставляет android приложение?
2. Как происходит обмен данными между МРП и мобильным телефоном?
3. Иногда данные в ходе обмена могут искажаться, как узнать, что данные были не точными? Какие алгоритмы могут быть использованы для обеспечения надежности передачи данных?
4. Опишите процесс подключения ИК датчика и его настройки для приема данных.

ЛАБОРАТОРНАЯ РАБОТА №7

Тема: «Программирование алгоритмов движения платформы по заданной траектории»

Цель работы: Написать скрипт, который позволит МРП двигаться вдоль черной линии.

При выполнении лабораторной работы обучающийся должен:

знать: простые операции по программированию алгоритмов движения платформы по траектории.

уметь: писать программный код для движения платформы по заданной траектории.

Формируемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие;

ПК 2.2. Диагностировать неисправности мехатронных систем и мобильных робототехнических комплексов с использованием алгоритмов поиска и устранения неисправностей.

Время выполнения работы: 1 час 30 минут.

Ход работы:

1. Проведение преподавателем инструктажа по технике безопасности.
2. Последовательно реализовать порядок выполнения работы с занесением данных в технический отчет.
3. Письменно ответить на контрольные вопросы.

Порядок выполнения работы

1. В любом графическом редакторе нарисуйте извилистую траекторию, по которой будет ездить робот, либо распечатайте готовую из файла trajectory.pdf (рисунок 24).

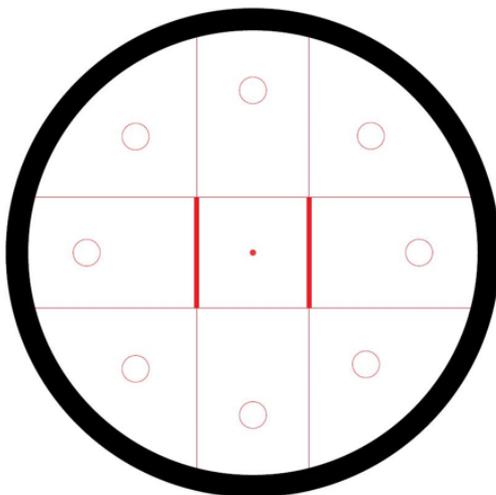


Рисунок 24- Пример трассы для МРП

Рассмотрим движение по линии с использованием двух датчиков цвета. В начальном состоянии МРП нужно установить таким образом на трек, чтобы черная линия проходила между датчиками цвета.

Алгоритм движения МРП будет следующий:

- Если оба датчика видят белый цвет – двигаемся вперед;
- Если один из датчиков видит белый, а другой черный – поворачиваем в сторону черного;
- Если оба датчика видят черный цвет – мы на перекрестке (например, остановимся).

2. Запрограммируйте алгоритм выше с использованием функций из лабораторной работы №3

3. Модифицируйте алгоритм так, чтобы уменьшить время прохождения трека.

4. Сравните свои результаты с коллегами.

Контрольные вопросы

1. Расскажите алгоритм движения по линии с двумя датчиками цвета.
2. Можно ли использовать 1 датчик для решения этой же задачи?
3. Как вы думаете, можно ли использовать П, ПИ, ПД, ПИД регуляторы при решении данной задачи? А при использовании датчиков освещенности? Какие модификации в алгоритме пришлось бы произвести?

ЛАБОРАТОРНАЯ РАБОТА №8

Тема: «Реализация алгоритмов самоориентации платформы с помощью вспомогательных датчиков»

Цель работы: Научиться создавать системы принятия решений для выполнения различных задач по перемещению МРП.

При выполнении лабораторной работы обучающийся должен:

знать: последовательность написания алгоритмов самоориентации платформы с помощью вспомогательных датчиков; технические характеристики вспомогательных датчиков.

уметь: реализовывать программные коды по управлению платформы.

Формируемые компетенции:

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие;

ПК 2.1. Осуществлять техническое обслуживание компонентов и модулей мехатронных систем и мобильных робототехнических комплексов в соответствии с технической документацией.

Время выполнения работы: 1 час 30 минут.

Ход работы:

1. Проведение преподавателем инструктажа по технике безопасности.
2. Последовательно реализовать порядок выполнения работы с занесением данных в технический отчет.
3. Письменно ответить на контрольные вопросы.

Порядок выполнения работы

1. Постройте простой лабиринт из книг или подручных материалов для МРП (рисунок 25).

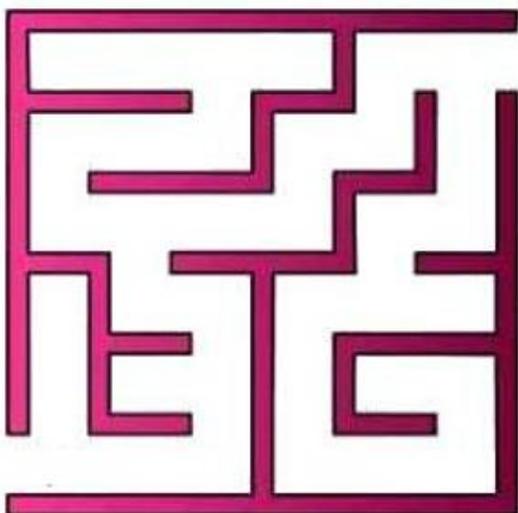


Рисунок 25- Пример лабиринта для МРП

2. На основании предыдущих лабораторных работ добавьте в новый скрипт функции передвижения вперед, разворота влево / вправо на 90° , а также чтения значений УЗ датчика дистанции.

3. Напишите алгоритм, который по правилу правой руки (в каждой узловой точке лабиринта выбрать одно и то же направление поворота, например, вправо) будет проходить ваш лабиринт.

Для реализации данного алгоритма используйте УЗ датчик дистанции, который поможет вам искать стены и тупики. Если расстояние до стены меньше $d_{\text{порогового}}$, то повернуть на 90° .

4. Перепишите алгоритм с использованием правила левой руки и сравните время прохождения лабиринта.

5. Найдите наклонную поверхность и поставьте на нее МРП.

6. При помощи GY-9250 получите значения крена и тангажа (рисунок 26).

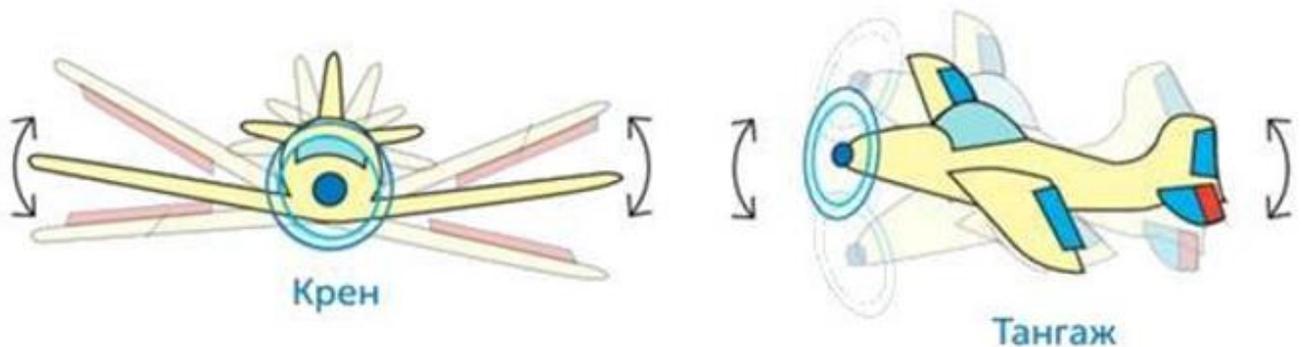


Рисунок 26- Крен и тангаж

7. Реализуйте регулятор на ваш выбор (П, ПИ, ПД, ПИД), который будет из состояния покоя по значению тангажа регулировать скорость тележки, так, чтобы при минимальной скорости она могла заехать наверх.

8*. Немного измените крен наклонной поверхности. Добавьте в регулятор значение крена и регулируйте скорость тележки, так, чтобы при минимальной скорости она могла заехать наверх.

Контрольные вопросы:

1. Как при помощи УЗ датчика дистанции выбраться из лабиринта?
2. Можно ли при помощи датчика GY-9250 и УЗ датчика дистанции построить карту глубины помещения?
3. Как влияют П, И, Д составляющие регулятора на выходную величину?

ПРАВИЛА ТЕХНИКИ БЕЗОПАСНОСТИ ПРИ РАБОТЕ ОБУЧАЮЩИХСЯ В ЛАБОРАТОРИЯХ

1. В целях предупреждения несчастных случаев напряжение выше 36 В следует считать опасным для жизни.
2. Начиная работу, обучающиеся должны убедиться в том, что на лабораторном столе (вводные рубильники, пакетные выключатели) нет напряжения.
3. Составление, разборку или изменение схемы производят только с разрешения преподавателя.
4. Запрещается включать вновь составленную или измененную схему без предварительной проверки ее преподавателем.
5. Запрещается прикасаться к токоведущим частям и металлическим частям незаземленных электрических аппаратов, если на щите имеется напряжение.
6. Все операции производить только одной рукой. При этом следует остерегаться прикосновений какой-либо частью тела к окружающим металлическим либо влажным предметам. Опасно, например, прикасаться одновременно к электрическим машинам, корпусу щита, водопроводным трубам, трубам центрального отопления или находиться на мокром либо цементном полу.
7. Перед включением напряжения следует убедиться в том, что все регулирующие аппараты находятся в исходном положении. После отключения напряжения необходимо немедленно восстановить на всех регулировочных аппаратах исходное положение.
8. Перед включением напряжения следует предупредить об этом всех участников работы. Необходимо убедиться, что никому из них не угрожает опасность попасть под напряжение.
9. Если при прикосновении к какой-либо части оборудования ощущается напряжение, то необходимо прекратить работу, выключить ток и вызвать преподавателя.
10. Если до начала работы или в ходе работы обнаружена неисправность оборудования, следует прекратить работу, отключить напряжение и сообщить преподавателю или инженеру о неполадках в работе. Устранять неполадки собственными силами запрещается.
11. При работе с цепями переменного тока, содержащими конденсаторы, следует соблюдать особую осторожность, имея в виду возможность значительного возрастания напряжения на отдельных участках по сравнению с напряжением источника тока вследствие возможного явления резонанса напряжений.
12. Запрещается прикасаться к приборам, находящимся на задней стенке щитов.
13. Запрещается обучающимся заходить за лабораторный стенд.

При необходимости пользоваться кнопкой аварийного срочного отключения.

Запрещается приступать к выполнению работы до тех пор, пока преподавателем не будет установлено, что студенту известны цель работы, метод

Ее выполнения, способ обращения с оборудованием, диапазон переменных величин и предполагаемые результаты.

Запрещается покидать лабораторию без разрешения преподавателя.

Запрещается оставлять без надзора установки, приведенные в рабочее состояние.

Перед началом работы следует распределить между членами бригады обязанности с таким расчетом, чтобы обеспечить соблюдение правил техники безопасности. Необходимо, например, обеспечить возможность быстрого аварийного отключения установки студентом, постоянно находящимся вблизи главного рубильника.

Рекомендуется останавливать агрегат всякий раз, когда возникает необходимость обсудить дальнейший план работы.

Запрещается переносить приборы с одного места на другое.

Запрещается трогать оборудование, неиспользуемое в данной работе.

После проработки настоящей инструкции на первом лабораторном занятии все обучающиеся обязаны расписаться в лабораторном журнале по технике безопасности.

Без инструктажа и отметки в журнале преподавателю категорически запрещается допускать обучающихся к лабораторным работам.

КРИТЕРИИ ОЦЕНКИ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Оценка *«отлично»* ставится в том случае, если обучающийся:

- а) выполнил работу в полном объеме с соблюдением необходимой последовательности проведения опытов и измерений;
- б) самостоятельно и рационально выбрал и подготовил для опыта необходимое оборудование, все опыты провел в условиях и режимах, обеспечивающих получение результатов и выводов с наибольшей точностью;
- в) в представленном отчете правильно и аккуратно выполнил все записи, таблицы, рисунки, чертежи, графики, вычисления и сделал выводы;
- г) правильно выполнил анализ погрешностей;
- д) соблюдал требования безопасности труда.

Оценка *«хорошо»* ставится в том случае, если выполнены требования к оценке *«отлично»*, но:

- а) опыт проводился в условиях, не обеспечивающих достаточной точности измерения,
- б) было допущено два-три недочета, или не более одной негрубой ошибки и одного недочета.

Оценка *«удовлетворительно»* ставится, если:

работа выполнена не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы, или если в ходе проведения опыта и измерений были допущены следующие ошибки:

- а) опыт проводился в нерациональных условиях, что привело к получению результатов с большей погрешностью,
- б), или в отчете были допущены в общей сложности не более двух ошибок (в записях единиц, измерениях, в вычислениях, графиках, таблицах, схемах, анализе погрешностей и т. д.), не принципиального для данной работы характера, но повлиявших на результат выполнения,
- в) или работа выполнена не полностью, однако объем выполненной части таков, что позволяет получить правильные результаты и выводы по основным, принципиально важным задачам работы.

Оценка *«неудовлетворительно»* ставится в том случае, если:

- а) работа выполнена не полностью, и объем выполненной части работы не позволяет сделать правильных выводов,
- б) или опыты, измерения, вычисления, наблюдения производились неправильно,
- в) или в ходе работы и в отчете обнаружилось в совокупности все недостатки, отмеченные в требованиях к оценке *«3 - удовлетворительно»*.

Библиографический список

1. Васильков, А.В. Источники электропитания: Учебное пособие / А.В.Васильков, И. А. Васильков. – М.: Форум, 2016. – 400 с.
2. Ключев, А.С. Монтаж средств измерений и автоматизации: справочник / А.С. Ключев. – М: Энергоатомиздат, 2012. – 114 с.
3. Лоторейчук, Е.А. Теоретические основы электротехники: учебник / Е.А. Лоторейчук. - М.: ИД «ФОРУМ»: ИНФРА-М, 2017. – 317 с.
4. Магда, Ю. Raspberry Pi. Руководство по настройке и применению / Ю. Магда. – Издательство: ДМК-Пресс, 2014. – 147 с.
5. Монк, С. Мейкерство. Arduino и Raspberry Pi. Управление движением, светом и звуком / С. Монк. – Издательство: СПб.: БХВ-Петербург, 2017. – 336 с.
6. Петин, В.А. Микропроцессоры RASPBERRY PI. Практическое руководство / В.А. Петин. – Издательство.: ВHV», 2015. –113с.
7. Петин, В.А. Проекты с использованием контроллера Arduino / В.А. Петин. – СПб.: БХВ-Петербург, 2014. – 400 с.
8. Приказ Минобрнауки России от 09.12.2016 № 1550 «Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 15.02.10 Мехатроника и мобильная робототехника (по отраслям)».
9. Ричардсон, М., Уоллес, Ш. Заводим Raspberry / М. Ричардсон, Ш.Уоллес. – Издательство: Амперка, 2015. – 230с.
- 10.Р.Дэвис,С. С++ для чайников / С. Р.Дэвис. – Издательство: Диалектика, 2018. – 400 с.
- 11.Славинский, А.К. Электротехника с основами электроники: Учебное пособие / А.К. Славинский, И.С. Туревский. – М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2016. – 448 с.
- 12.Федотов, А.В. Основы теории автоматического управления. Учебное пособие / А.В. Федотов - Омск: Изд-во ОмГТУ, 2012. - 279 с.
- 13.Шварц, М. Интернет вещей с ESP8266 / М. Шварц. – Издательство: СПб.: БХВ-Петербург, 2018. –192 с.
- 14.Шеховцов, В.П. Электрическое и электромеханическое оборудование: учебник / В.П. Шеховцов. – 3-е издание. – М.: ФОРУМ: ИНФРА-М, 2017. – 416 с.
- 15.Шишмарев, В.Ю. Автоматика. Учебник для среднего профессионального образования / В.Ю. Шишмарев. М.: Издательский центр «Академия», 2016. - 288 с.