



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)

Институт опережающих технологий «Школа Икс»

Директор ИОТ «Школа Икс»
П.В. Герасин
подпись И.О. Фамилия
«_____» _____ июля 2024г.

ОТЧЁТ

По _____ Учебная практика
вид практики

В _____ ДГТУ ИОТ «Школа Икс»
наименование базы практики

Обучающийся _____ «03» июля 2024г. Р. А. Лемешкин
подпись дата И.О.Фамилия

Обозначение отчёта _____ 15.03.06.930000.000 Группа _____ ХР21

Направление подготовки _____ 15.03.06 Мехатроника и робототехника
код и наименования направления подготовки

Направленность (профиль) _____

Руководитель практики:

от предприятия _____ директор _____ П.В. Герасин
должность подпись, дата И.О.Фамилия

от ДГТУ ИОТ «Школа Икс» _____ М.П. _____ Д.Р. Хашев
ст. преп. _____ И.О.Фамилия
должность подпись, дата

Оценка _____ _____ дата _____ подпись преподавателя

Ростов-на-Дону
2024



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)

Институт опережающих технологий «Школа Икс»

ЗАДАНИЕ

По _____ Учебная практика
вид практики

В _____ ДГТУ ИОТ «Школа Икс»
наименование базы практики

В период с «03» июня 2024г. по «03» июля 2024г.

Обучающийся _____ Лемешкин Руслан Андреевич
Фамилия Имя Отчество полностью

Обозначение отчёта _____ 15.03.06.930000.000 _____ Группа _____ ХР21

Срок предоставления отчёта на кафедру «03» июля 2024г.

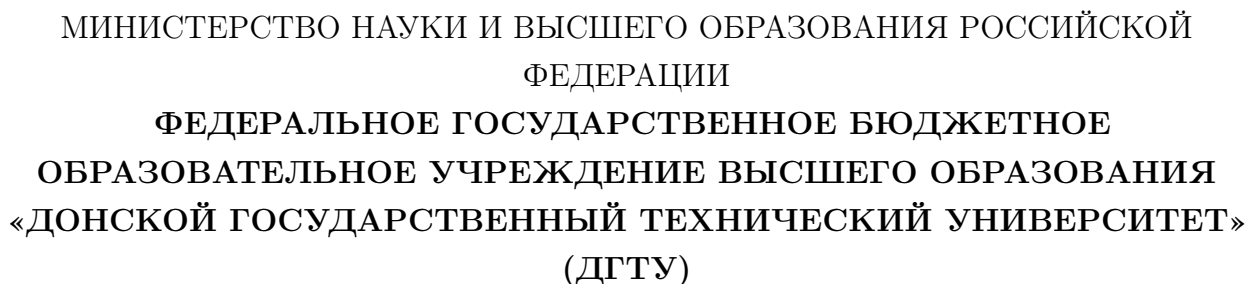
Содержание индивидуального задания:

Создать программное обеспечения для дифференциальной платформы "Eurobot Жук".

Функции: езда по энкодерам; езда по ультразвуковым датчикам; езда по гироскопу;
управление сервоприводами и комбинация этих функций.

Руководитель практики от
ДГТУ ИОТ «Школа Икс» _____ «03» июля 2024г. _____ Д.Р. Хашев
подпись дата И.О.Фамилия

Обучающийся _____ «03» июля 2024г. _____ Р. А. Лемешкин
подпись дата И.О.Фамилия



Директор	ИОТ «Школа Икс»
	П.В. Герасин
подпись	И.О. Фамилия
« »	июля 2024г.

№	Мероприятие	Срок выполнения
1	Получение доступов к материалам	03.06.24
2	Ознакомление с имеющейся документацией	04.06.24-06.08.24
3	Изучение фактической реализации проекта	09.06.24-11.06.24
4	Реализация собственных доработок	14.06.24-22.06.24
5	Написание документации к роботам	25.06.24-29.06.24

от предприятия	<div style="border-bottom: 1px solid black; display: inline-block; width: 100%;">директор</div> <div style="display: inline-block; width: 100%;">подпись, дата</div>	<div style="border-bottom: 1px solid black; display: inline-block; width: 100%;">П.В. Герасин</div> <div style="display: inline-block; width: 100%;">И.О.Фамилия</div>
от ДГТУ ИОТ «Школа Икс»	<div style="border-bottom: 1px solid black; display: inline-block; width: 100%;">М.П. ст. преп.</div> <div style="display: inline-block; width: 100%;">подпись, дата</div>	<div style="border-bottom: 1px solid black; display: inline-block; width: 100%;">Д.Р. Хашев</div> <div style="display: inline-block; width: 100%;">И.О.Фамилия</div>

Ростов-на-Дону
2024

ДНЕВНИК ПРОХОЖДЕНИЯ ПРАКТИКИ

В данном разделе ежедневно, кратко и четко записываются выполняемые работы, и в конце каждой недели журнал представляется для проверки руководителю (от предприятия и университета) практики. При выполнении одной и той же работы несколько дней, в графе «дата» сделать запись «с ____ по ____».

Дата	Место работы	Выполняемые работы	Оценка руководителя
03.06.24	ДГТУ ИОТ «Школа Икс»	Получение доступов к материалам	
с 04.06.24 по 06.08.24	Брейк 5	Ознакомление с имеющейся документацией	
с 09.06.24 по 11.06.24	Брейк 4	Изучение фактической реализации проекта	
с 14.06.24 по 22.06.24	Брейк 2	Реализация собственных доработок	
с 25.06.24 по 29.06.24	Брейк 3	Написание документации к роботам	

ОТЗЫВ-ХАРАКТЕРИСТИКА

Обучающийся _____ Лемешкин Руслан Андреевич _____
Фамилия Имя Отчество

2 курса группы ХР21 Институт опережающих технологий «Школа Икс»

Вид практики _____ Учебная практика _____

Наименование базы практики _____ ДГТУ ИОТ «Школа Икс» _____

Обучающийся выполнил задания программы практики

Обучающийся дополнительно ознакомился/изучил

Заслуживает оценки _____

Руководитель практики

от предприятия

П.В. Герасин

« ____ » _____ июля 2024г.

М.П.

Содержание

Содержание	1
Введение	2
1 Обзор работа	3
2 MicroPython	6
2.1 Окружение	6
2.2 Библиотеки	7
3 Энкодеры	9
4 Ультразвуковые датчики	11
5 Гироскоп	12
6 Подготовка отчёта	14
Заключение	15

Введение

Данный отчёт по практике можно считать документацией к роботам Жук проекта Eurobot на конец июля 2024 года и описанием технических особенностей этих роботов.

Изначальная задача, создание прошивки для робота, сильно зависит от понимания различных особенностей устройства, поэтому создание документации является важным этапом для достижения цели.

Краткий обзор отчёта:

В первой части описан язык программирования (ЯП) MicroPython и среда для работы с ним;

Во второй приведены общие положения, касающиеся робота, рисунок-схема подключений и некоторые параметры корпуса;

В третьей описаны особенности использованных в роботах энкодеров и мои наработки, связанные с ними;

В четвёртой затронуты особенности УЗ датчиков;

В пятой описана работа, проделанная с гироскопом.

1 Обзор работа

Робот (Жук) предназначен для ознакомления студентов первого и второго курса с особенностями программирования и конструирования роботов. Он представляет собой красный пластиковый цилиндр с двумя колёсами по бокам, а также упорами спереди и сзади в виде железных шариков в пластиковых креплениях. При изучении робота стало понятно, что упорам требуется доработка: надо уменьшить их высоту, чтобы опираясь на них робот не отрывал колёса от земли. Я временно решал проблему, вытаскивая один из шариков. Впоследствии было предложено разобрать робота, и уменьшить высоту внешней части крепления, добавив шайбы на крепёжные винты.

Жук управляется микроконтроллером (МК) ESP-WROOM-32 (esp32, еспшка), изображение есть на 1.1 и в открытых источниках. Также в нём есть активная периферия:

- 6 УЗ-датчиков HRC SR04;
- 2 электромотора с энкодерами GA25-370;
- 1 MPU 6050 (гироскоп)(GY-521).

Важно заметить, что её можно использовать только физически отключая не используемые в данный момент элементы, при этом использовать все периферийные устройства одновременно нельзя.

Кроме перечисленного выше, 2 ультразвуковых датчика подключены через **расширитель пинов i2c pcf8574**, моторы управляются через **драйвер L9110S**, а питание идёт от аккумуляторов через **DC-DC преобразователь HW-411**. Информацию о модели аккумуляторов предстоит найти в будущем и включить в документацию. Подробнее каждый из этих элементов будет освещён в соответствующих разделах.

Ниже, на рисунке 1.1 отражена схема подключения элементов друг к другу. Рисунок был отредактирован и актуализирован мной в процессе подготовки к дальнейшей работе. Ознакомиться с оригиналом рисунка можно на доске в мире. При работе над рисунком были использованы

разнообразные инструменты, предлагаемые упомянутой электронной доской и я научился более тонко работать с соединительными линиями и местами их соединения, что поможет в будущем делать аналогичные схематические рисунки других устройств.

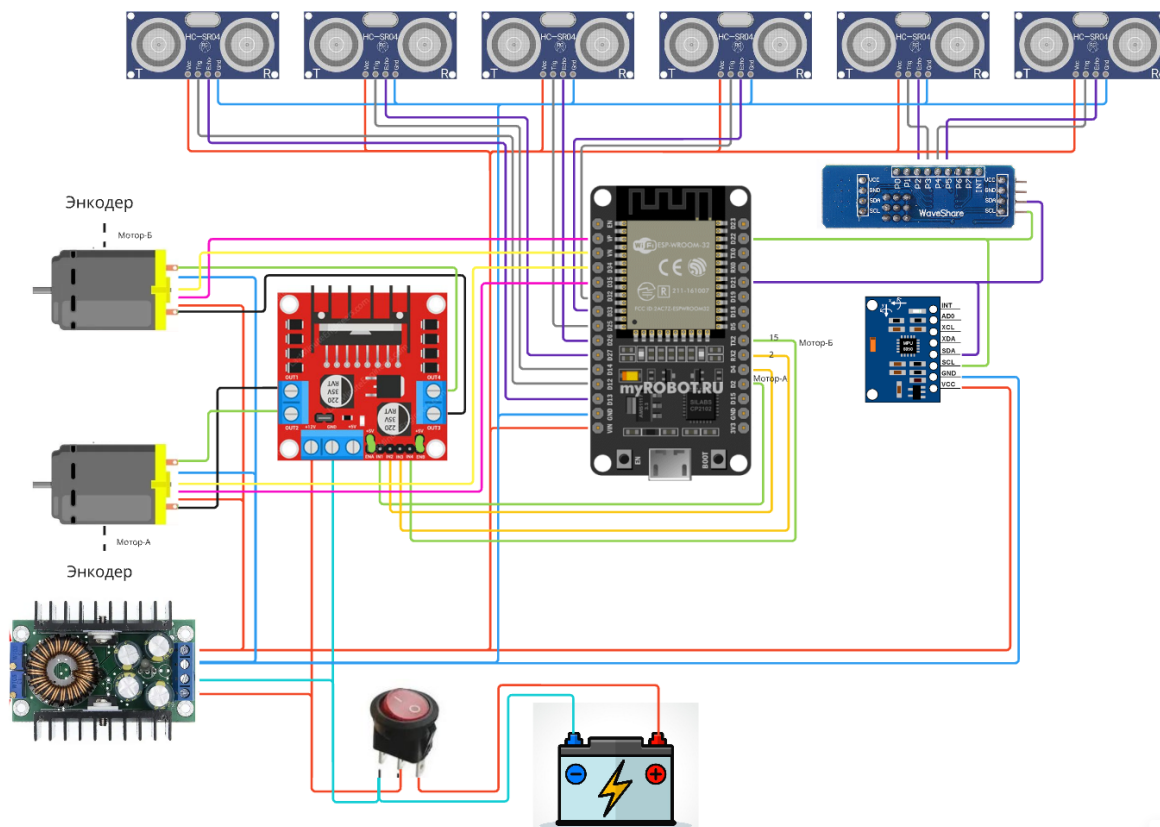


Рисунок 1.1 — Схематический рисунок подключений

При доработке рисунка я научился:

- прозванивать электрические цепи
- изучил сопутствующую роботу документацию
- выявил несоответствия реального и описанного в проекте
- рассмотрел чужие наработки в области создания мобильных роботов

Анализируя всё вышесказанное, можно сделать вывод, что:

- а) Жук обладает разнообразными датчиками для отредактирования в пространстве

- б) Жук имеет аккумуляторы для автономной работы
- в) Документация часто не соответствует действительности, и её актуализация является необходимой частью рабочего процесса
- г) При дальнейшей работе необходимо добавить к документации модель аккумуляторов

2 MicroPython

MicroPython — это интерпретируемый язык программирования, основанный на Python. Отличается наличием библиотек для управления микроконтроллерами и некоторыми оптимизациями для большей производительности. Был выбран по историческим причинам и из-за личных предпочтений.

Альтернативой является язык Arduino, но здесь он рассматриваться не будет, так как в моей работе не использовался.

Для правильной работы необходимо прошить МК. В рамках практики я использовал встроенные средства Thonny. При этом есть альтернативные способы, описание которых можно найти в открытых источниках.

2.1 Окружение

Для работы с кодом использовалась IDE Thonny, установленная в виртуальное окружение для Python. Создаётся виртуальное окружение последовательностью команд в терминале:

```
mkdir /path/to/project/directory/ # здесь создаётся директория  
    (папка) в которой будут храниться файлы проекта. Команда зависит  
    от ОС (приведён пример для linux) и имеет аналоги в граф.
```

интерфейсе

```
cd /path/to/project/directory/ # переход в рабочую директорию.
```

Команда зависит от ОС (приведён пример для linux) и обязана исполняться в командной строке

```
venv venv # команда, которая создаёт в текущей директории папку
```

venv и автоматически генерирует в ней необходимые файлы

Для активации venv надо запустить скрипт activate. Например:

```
source /path/to/project/directory/venv/bin/activate
```

Сама IDE устанавливается командой `pip install thonny` внутри активированного venv.

Все библиотеки рекомендуется устанавливать локально в окружение текущего проекта, что я и сделал с помощью той же команды `pip install`.

В моём случае папкой проекта является `Practice/`. В ней создана папка `venv`, в которой собраны все автоматически генерируемые файлы. Рядом стоит создать папки для файлов с собственным кодом и файлов библиотек, либо же (для оптимизации) собрать в один файл все необходимые Функции, но при таком подходе дальнейшая разработка будет несколько затруднена.

2.2 Библиотеки

Как было сказано ранее, важной особенностью выбранного ЯП является сильное разбиение на библиотеки, поэтому работу следует начинать как раз с выбора этих пакетов.

Для получения общей информации я задал вопрос нейросети Perplexity и она предложила мне использовать следующие библиотеки:

machine - встроенная библиотека для работы с микроконтроллером и его периферийными устройствами.

Документация: <https://docs.micropython.org/en/latest/library/machine.html>

ustruct - встроенная библиотека для работы с байтовыми структурами.

Документация: <https://docs.micropython.org/en/latest/library/ustruct.html>

utime - встроенная библиотека для работы с таймерами и датой.

Документация: <https://docs.micropython.org/en/latest/library/utime.html>

uos - встроенная библиотека для работы с файловой системой.

Документация: <https://docs.micropython.org/en/latest/library/uos.html>

machine.I2C - встроенный класс для работы с интерфейсом I2C.

Документация: <https://docs.micropython.org/en/latest/library/machine.I2C.htm>

pcf8574 - сторонняя библиотека для работы с расширителем пинов PCF8574.

Документация: <https://github.com/mcauser/micropython-pcf8574>

hw411 - сторонняя библиотека для работы с DC-DC преобразователем HW-411.

Документация: <https://github.com/mcauser/micropython-hw411>

mpu6050 - сторонняя библиотека для работы с гироскопом MPU-6050.

Документация: <https://github.com/adamjezek98/MPU6050-ESP8266-MicroPython>

sr04 - сторонняя библиотека для работы с УЗ-датчиками HRC SR04.

Документация: <https://github.com/rsc1975/micropython-hcsr04>

В этом ответе оказалась несуществующей библиотека для работы с драйвером, а судя по информации в поисковике для него не нужно дополнительное ПО. К указанной библиотеке для mpu6050 документация крайне скудная, в дальнейшем следует глубже изучить вопрос и поискать альтернативы. С большой вероятностью в рамках проекта окажутся бесполезны библиотеки для управления файловой и операционной системами.

Таким образом в дальнейшей работе надо обратить пристальное внимание на упомянутые библиотеки для работы с sr04, rsc8574, временем, а также machine для работы с самим МК.

3 Энкодеры

Двигатели с энкодерами подключены по шести проводам, значение которых показано на картинке ниже:



Рисунок 3.1 — Распиновка энкодеров

При подключении двигателя нет разницы, где плюс, а где минус, поэтому предлагаю подбирать нужный результат в процессе написания кода.

Также важно сказать, что двигатели и энкодеры подключаются через аналоговые пины и управление происходит просто подачей напряжения на соответствующие выходы без всяких библиотек. Для снижения нагрузки на линии питания микроконтроллера используется драйвер.

Ниже приведён отрывок кода, который выключает моторы. Судя по моим наблюдениям, у моторов в данной конструкции робота есть всего 3 состояния: вперёд, назад и выключены. Чтобы поменять направление движения, надо поменять у всех пинов IN на OUT и наоборот.

```
from machine import Pin

pin_l1 = Pin(2, Pin.IN)
pin_l2 = Pin(15, Pin.OUT)
pin_r1 = Pin(4, Pin.IN)
pin_r2 = Pin(0, Pin.OUT)

pin_l2.value(0)
pin_r2.value(0)
```

4 Ультразвуковые датчики

Для ориентирования в пространстве в роботе "Жук" установлены шесть ультразвуковых датчиков hrc sr04. Каждый из них рассчитан на напряжение 5v, но способен работать от линии 3.3v. Питание идёт через МК, поэтому одновременное подключение всех датчиков перегружает esp32, вызывают перегрев линий питания и мешают прошивке.

В будущем необходимо расширить данный раздел информацией о библиотеке для данных датчиков, их распиновке и максимальном количестве, которое может работать от еспшки одновременно.

5 Гироскоп

Из-за проблемы с УЗ датчиками, я стал работать с гироскопом. Результатом моей работы стал приведённый ниже код.

```
from machine import Pin
import utime
from mpu6050 import mpu6050

mpu = mpu6050.MPU6050(Pin(21, Pin.IN),
                      Pin(22, Pin.IN))

# Функция для чтения данных с модуля MPU-6050
def gyro_read():
    # Чтение данных акселерометра
    acc_x = mpu.get_accel_x()
    acc_y = mpu.get_accel_y()
    acc_z = mpu.get_accel_z()

    # Чтение данных гироскопа
    gyro_x = mpu.get_gyro_x()
    gyro_y = mpu.get_gyro_y()
    gyro_z = mpu.get_gyro_z()

    return acc_x, acc_y, acc_z, gyro_x, gyro_y, gyro_z

# Основная программа
while True:
    # Чтение данных с модуля MPU-6050
    acc_x, acc_y, acc_z, gyro_x, gyro_y, gyro_z = read_data()

    # Расчет положения и ориентации
```

```
roll = (acc_x + acc_y + acc_z) / 3
pitch = (gyro_x + gyro_y + gyro_z) / 3

# Вывод результатов
print(f"Roll: {roll:.2f}°, Pitch: {pitch:.2f}°")

# Ожидание следующего цикла
utime.sleep(0.1)
```

В теории он должен рассчитывать вращение по вертикальной оси (roll) и горизонтальной (pitch), но на практике я не смог его проверить из-за особенностей программного обеспечения. По неизвестным причинам Thonny не видел библиотеку ctypes. Как один из вариантов решения этой проблемы — включение в исполняемый файл библиотеки три6050 необходимых функций из зависимых пакетов, но данный подход требует глубокого знания использованных библиотек и осуществить его в рамках практики не представилось возможным.

6 Подготовка отчёта

Важной частью практики я считаю подготовку отчёта, в процессе которой я расширил свои знания о языке программирования для автоматической вёрстки документов \LaTeX и подготовил шаблоны для быстрого оформления будущих отчётов. В процессе я прочитал большой объём литературы, описывающей различные нюансы написания программ.

Также для оформления были изучены и учтены ГОСТы 2.105 и 7.32.

Заключение

В процессе практики была проделана работа по изучению роботов Eurobot Жук, создан структурированный документ с их описанием, предложены решения некоторых проблем и намечены следующие цели при работе над данным проектом.

p.s. Для дальнейшей работы над документацией к Жуку можно запросить исходники этого отчёта у меня в телеграме или дискорде (@LemeRus). Разве что свёрстан он на L^AT_EX.