



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

Факултет Приложна математика и информатика



ДИПЛОМНА РАБОТА

Тема: Автоматизирана купичка за хранене на домашен любимец

Дипломант: Виктория Радославова Радкова

Фак. №: 471220010

Специалност: Информатика и софтуерни науки

Образователно-квалификационна степен: бакалавър

Дипломен ръководител: доц. д-р Златко Захариев

София 2024



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

Факултет Приложна математика и информатика



Утвърждавам:

Декан на ФПМИ:

(проф. д-р инж. Д. Иванова)

Дата на задаване: **10.04.2024**

ЗАДАНИЕ

за разработка на дипломна работа

Дипломант: Виктория Радославова Радкова

Фак. №: 471220010

Специалност: Информатика и софтуерни науки

Образователно- квалификационна степен: **бакалавър**

1. Тема на дипломната работа:

Автоматизирана купичка за хранене на домашен любимец

2. Заявител на темата: Катедра Информатика - ФПМИ

3. Обяснителна записка :

Автоматизирана IoT система за хранене на домашен любимец от разстояние, разработена с помощта на Raspberry Pi. Потребителят ще може да определя по кое време да бъде пусната храна в купичката на неговия домашен любимец, чрез мобилно приложение.

Дипломен ръководител:

Ръководител катедра:

доц. д-р Златко Захариев - ФПМИ

доц. д-р Златко Захариев

Съдържание

Увод	1
Глава 1	2
Проучване	2
1.1 Интернет на нещата (IoT)	2
1.1.1 Как работи IoT	2
1.1.2 Сигурност на IoT	3
1.1.3 Приложения на IoT	4
1.1.4 Предимства и недостатъци на IoT	6
1.2 Автоматизирани системи за хранене на домашни любимци	8
1.2.1 Примери за автоматизирани системи за хранене на домашни любимци	9
1.2.2 Предимства и недостатъци на автоматизираните системи за хранене на домашни любимци	10
Глава 2	13
Планиране	13
2.1 Софтуерни технологии	13
2.1.1 IoT платформи	13
2.1.2 Програмни езици	15
2.2 Хардуерни компоненти	17
2.2.1 Микроконтролери	17
2.2.2 Мотори	19
2.3 Избрани технологии	21
2.3.1 Софтуерни технологии	21
2.3.2 Хардуерни компоненти	22
2.3.3 Допълнителни компоненти	23
Глава 3	24
Проектиране	24
3.1 Основни изисквания	24

3.1.1 Функционални изисквания	24
3.1.2 Нефункционални изисквания	24
3.2 Схема на системата за хранене на домашни любимци	25
Глава 4	27
Реализация.....	27
4.1 Свързване на хардуерните компоненти	27
4.2 Изработка на контейнера за храна.....	28
4.3 Конфигурация на IoT платформата Blynk	30
4.4 Реализация на мобилно приложение чрез Blynk	34
4.5 Реализация на функционалността в Raspberry Pi	34
4.5.1 Първоначална инициализация на Raspberry Pi	34
4.5.2 Връзка с микроконтролера	34
4.5.3 Програмна реализация	35
4.6 Тестване	41
Глава 5	43
Представяне на готовия продукт	43
5.1 Мобилно приложение	43
5.2 Автоматизирана система за хранене на домашни любимци	46
Заключение.....	49
Декларация за авторство	50
Литературни източници	51
Използвани съкращения	52
Приложения.....	53

Увод

С постоянно развиващите се технологии в днешно време, хората търсят начини да оптимизират и намалят ежедневните си задачи и задължения. В този аспект, Интернет на нещата (IoT) играе ключова роля, предоставяйки възможност за управление на множество смарт устройства от разстояние.

Целта на настоящата дипломна работа е да се реализира автоматизирана IoT система за хранене на домашни любимци от разстояние, която да се управлява дистанционно, посредством мобилно приложение.

Процесът на разработка ще се състои от няколко основни етапа. Първоначално ще бъде проведено задълбочено проучване на подобни съществуващи продукти, като за целта ще бъдат разгледани статии и видеа. Това ще подпомогне за идентифицирането на текущите тенденции и предизвикателства в областта на IoT системите за хранене на домашни любимци.

Следващият етап ще се фокусира върху проектирането на системата. Ще бъдат избрани подходящи хардуерни компоненти, ще се определи архитектурата на софтуера и начина по който той ще комуникира с хардуера. Ще се изберат подходящи технологии за разработка, включващи IoT платформата.

В третият етап ще бъде извършена реалната разработка на системата. Хардуерните компоненти ще бъдат сглобени и конфигурирани, а избрания микроконтролер ще бъде програмиран така, че да комуникира с тях. Паралелно с това ще се разработи софтуерът, включително и мобилното приложение, чрез което потребителите ще контролират храненето на домашния си любимец от разстояние.

След завършването на хардуерната и софтуерната разработка, системата за хранене на домашни любимци от разстояние ще бъде тествана, за да се подсигури нейната функционалност, надеждност и ефективност в реални условия.

Последният етап ще включва подготовка на подробна техническа документация, описваща всички аспекти на разработката, включително хардуерни схеми, софтуерен код и инструкции за инсталация и употреба.

Тази дипломна работа е насочена към създаването на иновативно решение за автоматизирано хранене на домашни любимци от разстояние, чрез използване на IoT технологии. Изследването и разработката обхващат както хардуерни, така и софтуерни компоненти, които заедно формират една интегрирана система, управлявана дистанционно чрез мобилно приложение.

Глава 1

Проучване

1.1 Интернет на нещата (IoT)

Интернет на нещата (IoT, съкратено от Internet of Things) представлява система от взаимосвързани физически устройства, обменящи данни по между си чрез Интернет.[1] IoT предоставя възможност за опростяване и автоматизиране на задачи, които са сложни и понякога извън обхвата на човешките възможности.[2] IoT устройствата, известни още като „умни обекти“, могат да варират от устройства за „умен дом“, като например автоматизирани термостати, крушки, климатици, до сложни индустриални машини и транспортни системи.[3]

1.1.1 Как работи IoT

Екосистемата на Интернет на нещата е изградена от умни устройства, свързани към Интернет, които използват процесори и сензори, за да събират и обработват информация и данни от околната среда. IoT устройствата обменят събраните сензорни данни посредством връзка с централен хъб, където информацията се изпраща към облачно хранилище или се обработва и анализира локално. След първоначалната си конфигурация, умните устройства изпълняват повечето си функции автоматично, без нуждата от пряка човешка намеса. Въпреки това, потребителите имат възможност да достъпват събраната информация и статистическите данни и да правят промени по настройките. [1]

Интернетът на нещата използва облачни технологии, изкуствен интелект (AI, съкратено от Artificial Intelligence) и машинно обучение, за да улесни и ускори процеса на събиране на данни, както и да автоматизира и оптимизира техния анализ.

Типът на мрежата, използваните комуникационни протоколи и начинът на свързване на различните IoT устройства, зависят от специфичната Интернет на нещата платформа, която се използва, както и от технологичните и функционални възможности, които тя предлага.

Интернетът на нещата се базира на стандартите за Интернет протоколи (IP) и Протоколи за управление на предаванията (TCP), които осигуряват свързаността между умните устройства и Интернет.[2] Чрез тази свързаност, устройствата могат да събират, анализират и обменят данни, което улеснява автоматизацията и управлението на различни процеси, без нуждата от човешка намеса.

Съществуват два основни типа свързани устройства: проектирани като дигитални от самото начало и такива, които са физически предмети с добавена свързаност. „Дигитално пъrvите“ устройства, включват машини и устройства, които са специално проектирани с вградена свързаност, като например смартфони, медийни плейъри, терминали за мобилни плащания, реактивни двигатели и други. Тези устройства имат вградени възможности за генериране и обмен на данни и комуникират с други машини посредством комуникация тип „машина-към-машина“ (M2M, съкратено от Machine-to-Machine). От друга страна, „Физическо пъrvите“ устройства, като например ключодържатели, медицински устройства и превозни средства съдържат чипове и сензори, които са добавени след тяхното производство, което им позволява да изпълняват нови функции и да бъдат проследявани. Някои експерти разделят тези продукти в по-подробна класификация, включваща пет категории на интерактивност от чисто дигитални устройства, до чисто физически устройства без никакви дигитални функции.[2]

1.1.2 Сигурност на IoT

Един от основните проблеми на Интернет на нещата (IoT) е осигуряването на достатъчна защита срещу хакери. Историята познава многобройни случаи на хакери, които са успели да проникнат в различни умни устройства, като например видеокамери, бебешки монитори и медицински устройства. Липсата на подходяща сигурност може да доведе до загуба или неправилна употреба на лични данни, като например здравна или финансова информация. Свързаните устройства и системи, заедно с данните, съхранявани в облака, увеличават броя на уязвимите точки.

Например, през 2017-та година почти половин милион пейсмейкъри са били изтеглени от употреба, поради уязвимости към хакерски атаки и злоупотреба с устройствата. През същата година, престъпници са успели да пробият в системата на казино през умен аквариум, който не е бил добре защищен, в следствие на което са били откраднати големи количества данни.[2]

Друг известен проблем съществува при камерите за видеонаблюдение и за проследяване на движенията на хората. Когато едни видео данни се комбинират с други типове данни, генериирани от сензори, камери, клетъчни записи, компютърни логове и други системи, е възможно да се идентифицира къде е бил човек или какво е правил във всеки един момент. Тази информация може потенциално да застраши човешки животи, ако попадне в грешните ръце и бъде използвана за злоупотреба.

Въпреки огромните предимства и възможности, които предлага Интернетът на нещата, той също така поставя сериозни предизвикателства в областта на

сигурността и защитата на личните данни. Затова е от значителна важност е да се обърне достатъчно внимание на сигурността и защитата на данните в една IoT система.

1.1.3 Приложения на IoT

Бизнес

Много компании от различни сектори внедряват IoT технологии, за да повишат ефективността и производителността си, да разберат по-добре нуждите на клиентите си и да подобрят цялостните си бизнес резултати. Решенията, базирани на Интернет на нещата, предлагат предимства и ползи, които могат да бъдат както специфични за дадена индустрия, така и приложими в различни сектори.

Технологиите на Интернет на нещата спомагат за решаването на множество различни бизнес проблеми, като например:

- автоматизация на производствените процеси
- проследяване на прогреса на различни бизнес проекти
- наблюдение и анализ на клиентското поведение с цел подобряване на качеството на услугите
- проследяване на процеса на работа на служителите и контрол върху ключовите показатели за ефективност
- интеграция и адаптация на бизнес модели
- оптимално управление и използване на ресурсите

Селско стопанство

Интелигентните системи, базирани на технологиите на Интернет на нещата (IoT), намират широко приложение в съвременното стопанство. Те предоставят възможност за мониторинг и поддържане на оптимални нива на различни параметри като осветление, температура, влажност на въздуха и почвата. Чрез мрежа от свързани сензори, разположени стратегически в стопанството, се събират и анализират данни в реално време. Това позволява на фермерите да контролират условията в своите оранжерии, животновъдни ферми и складови помещения, като осигуряват подходяща среда за растенията и животните.

Интелигентни градове

Интернетът на нещата може да се приложи и за изграждане на интелигентни градове. В един интелигентен град могат да бъдат внедрени системи за интелигентно улично осветление, интелигентни измервателни уреди за комунални услуги, системи за контрол на трафика и наблюдение на замърсяването на въздуха. Това позволява намаляване на трафика, пестене на енергия и следене и поддържане на екологичните условия в норма.

В умните градове, също така могат да бъдат използвани интелигентни сензори и видеонаблюдение, с цел контрол на градската инфраструктура. Това позволява оптимизация на разходите и подобряване качеството на живот на населението.

Транспорт

Технологиите на Интернет на нещата позволяват оптимизиране на логистиката и ефективността на транспортната система. С тяхното внедряване става възможно проследяването на работните параметри на оборудването, маршрутите и местоположението на транспортните средства.

IoT базираните инструменти следят движението на превозни средства, като автомобили и камиони, в реално време и подпомагат навигацията. Също така, самите превозни средства могат да бъдат оборудвани със специални сензори, които да докладват за налягането в гумите и да съобщават за възможни повреди в тях, позволявайки на водача бързо да предприеме необходимите действия.

Тези технологии не само подобряват безопасността и ефективността на транспортните системи, но и значително намаляват разходите за поддръжка.

Медицина

Интернет на медицинските неща (IoMT, съкратено от Internet of Medical Things) представлява инфраструктура, състояща се от интелигентни устройства, специализирани програми, приложения и услуги. IoMT разширява възможностите на традиционната медицина, като позволява на лекарите да предоставят дистанционна грижа и същевременно да събират подробни данни за здравословното състояние на пациента.

Интелигентните устройства и IoT приложенията в медицината се разделят на два основни типа:

- професионални устройства за лекари и клиники
- потребителски устройства

Интелигентните устройства за следене на здравето включват разнообразие от преносими устройства и сензори, като например фитнес тракери и смарт

часовници. Друг пример за IoT устройства са апаратите за измерване на кръвното налягане и сърдечния ритъм. Тези устройства събират и анализират данни, което позволява на лекарите да поставят диагнози и да предлагат лечебни планове дори дистанционно.

Ителигентни домове

За обикновения потребител, приложението на Интернет на нещата може би е най-често срещано в домакинството. Тази област включва интелигентни уреди, които могат да бъдат управлявани чрез смартфон, като роботизирани прахосмукачки, кухненски роботи, хладилници, чайници и така нататък. Освен вече изброените умни уреди, съществуват и комплексни интелигентни домашни системи, които включват сензори за влажност на въздуха, сензори за дим, както и свързани отоплителни, осветителни и други устройства.

Ителигентните уреди в дома могат да обменят информация помежду си и да функционират според предварително зададени сценарии. Например, тези системи могат автоматично да включват осветлението в коридора, когато се отвори входната врата или когато сензорът за движение засече присъствие.

Умните уреди облекчават ежедневните задължения на потребителите и правят живота им по-комфортен. С нарастващото използване на тези технологии, интелигентните уреди стават все по-достъпни и популярни, подобрявайки качеството на живот на потребителите.

1.1.4 Предимства и недостатъци на IoT

Предимства

С развитието на технологиите за Интернет на нещата и тяхното навлизане в ежедневието ни, те предоставят разнообразни предимства и възможности в различни области. Тези технологии значително усъвършенстват начина, по който управляваме и контролираме различни аспекти на живота ни. Някои от основните предимства на Интернет на нещата са:

- лесен достъп до информация по всяко време, без значение от местоположението на потребителя, посредством потребителско умно устройство (смартфон, таблет, компютър)
- организирана работа на свързани интелигентни устройства и възможност за работа по предварително зададени параметри, без да е необходима пряка човешка намеса (автоматично осветление, терmostати и така нататък)

- прехвърляне на големи количества данни и информация между всички участващи устройства в една мрежа
- възможност за автоматизация на множество различни процеси, спомагащи за подобряване на комфорта в дома и в града или качеството на разнообразни бизнес услуги (автоматизирани производствени процеси, наблюдение на машини, предвиждане на повреди и така нататък)
- възможност за повече свободно време за потребителя (например с помощта на автоматизирани прахосмукачки и кухненски уреди)
- подобрена безопасност и сигурност (например чрез интеграция на сензори за дим и системи за видеонаблюдение)
- подобряване на здравето (чрез устройства за следене на човешките жизнени показатели)

Внедряването на IoT технологии трансформира нашето ежедневие, като предлага нови възможности за автоматизация, подобрява комфорта и ефективността и предоставя възможност за повече свободно време. Интернет на нещата не само улеснява ежедневието ни, но и предоставя по-голям контрол и гъвкавост, оптимизирайки ресурсите и процесите в дома и бизнеса.

Недостатъци

Въпреки че IoT технологиите предлагат значителни предимства, тяхното внедряване е свързано с редица трудности и рискове, които трябва да се вземат предвид. Те могат да варират в зависимост от конкретните приложения и индустрии, в които се използват. Някои от основните трудности с които потребителят може да се сблъска са:

- хакерски атаки и риск от изтичане на поверителна информация (хакерите могат да получат достъп до поверителна информация или да поемат контрол над самите умни устройства)
- вероятност за срив на цялата система при неизправност на едно от устройствата (например, ако централният хъб на интелигентен дом се повреди, всички свързани устройства могат да спрат да функционират правилно)
- липса на универсален международен стандарт за съвместимост на IoT устройствата (може да съществува несъвместимост между продукти от различни производители)

- нараствање на броја на умните устройства (може да доведе до значително нараствање на обема на генерираните данни, което може да затрудни тяхното събиране, съхранение и управление)

Въпреки съществуващите рискове и предизвикателства, развитието на IoT технологиите и тяхното приложение в различни индустрии продължава да се увеличава. С развитието на облачните технологии и внедряването на по-добри мерки за сигурност, тези технологии ще продължат да трансформират начина, по който живеем и работим. Разбирането и управлението на рисковете е ключово за успешното използване на IoT решенията, които предлагат значителни предимства и подобрения в ефективността и качеството на услугите.

1.2 Автоматизирани системи за хранене на домашни любимици

В съвременния свят, където забързаният начин на живот често ограничава времето, което можем да отделим на нашите домашни любимици, автоматизираните системи за хранене играят все по-важна роля. Тези технологии предоставят удобство и сигурност, като гарантират, че нашите любимици получават необходимата храна в точното време, дори когато ние не сме у дома. Създадени да отговорят на нуждите на собствениците на домашни любимици, автоматизираните хранилки предлагат редица предимства, които значително подобряват качеството на живот както за животните, така и за техните стопани.

Автоматизираните системи за хранене на домашни любимици не само осигуряват редовно и прецизно дозиране на храната, но също така помагат за поддържането на здравословни навици на хранене. Те могат да бъдат програмирани да разпределят определени количества храна в определени интервали, което е особено полезно за животни с хранителни нужди или за тези, които са склонни към преяждане. Също така, тези системи предоставят спокойствие на собствениците, които могат да бъдат сигурни, че техните домашни любимици ще бъдат нахранени дори при тяхно отсъствие.

През последните години, технологиите в областта на автоматизираните системи за хранене се развиха значително. Новите модели включват различни инновации като Wi-Fi свързаност, мобилни приложения за управление и програмиране на храненето, както и възможности за мониторинг на консумацията на храна в реално време. Някои усъвършенствани модели дори разполагат с камери, които позволяват на собствениците да наблюдават своите домашни любимици докато се хранят.

1.2.1 Примери за автоматизирани системи за хранене на домашни любимци

PETLIBRO Automatic Pet Feeder

Характеристики:

- PETLIBRO предлага няколко модела, включително такива с Wi-Fi свързаност и възможност за настройка на хранения чрез мобилно приложение. Един от популярните модели има капацитет от 8.5 чаши и може да работи на батерия.[4]

Предимства:

- Лесен за настройка и използване, възможност за настройка на множество хранения на ден, резервна батерия за продължаване на работата при прекъсване на захранването.

Недостатъци:

- Малък капацитет за храна, отнема 8 часа за зареждане на батерията.

Whisker Feeder Robot

Характеристики:

- Предлага удобно и интуитивно приложение за управление и настройка на различни хранения. Снабден е с резервна батерия, която осигурява до 24 часа работа при прекъсване на захранването, както и с устойчив на дъвчене кабел за захранване. Това го прави идеален за собственици, които търсят надеждност и удобство.[4]

Предимства:

- Прецизни настройки за хранене, устойчив на опити за преобръщане, лесен за настройка и синхронизация с приложение.

Недостатъци:

- Пластмасова купа вместо метална или стъклена, големи размери, висока цена.

PetSafe Smart Pet Feeder

Характеристики:

- Може да се програмират до 12 хранения на ден, с възможност за настройка на порции от 1/8 до 4 чаши. Осигурява се и бавно хранене за домашни любимци, които ядат твърде бързо.[5]

Предимства:

- Лесен за програмиране, прозрачен контейнер за лесно наблюдение на количеството храна, съвместимост както с мокра, така и със суха храна.

Недостатъци:

- Липса на Wi-Fi свързаност при базовия модел.

Cat Mate C300 Automatic Digital Pet Feeder

Характеристики:

- Подходящ за мокра храна, този модел има вградени ледени пакети, които поддържат храната свежа. Може да разпределя до три порции храна и разполага с лесен за почистване съд.[6]

Предимства:

- Идеален за мокра храна, съд за миене в миялна машина, вграден индикатор за състоянието на батерията.

Недостатъци:

- Малък капацитет за храна, изискава отделно закупуване на купички от неръждаема стомана.

1.2.2 Предимства и недостатъци на автоматизираните системи за хранене на домашни любимци

Предимства

Автоматизираните системи за хранене на домашни любимци предоставят множество предимства, които улесняват живота както на домашните любимци, така и на техните собственици. Тези предимства включват удобство, прецизност, безопасност и много други, които значително подобряват качеството на живот.

Удобство:

- Автоматизираните системи за хранене осигуряват редовно хранене на домашните любимци, дори когато собствениците са заети или извън дома. Това е особено полезно за хора с натоварени графици или тези, които често пътуват. Системите могат да бъдат програмирани да разпределят храна в точно определени часове, което гарантира, че домашните любимци получават необходимото хранене без прекъсвания.

Прецизност:

- Точността на дозиране на порциите е едно от най-големите предимства на автоматизираните системи за хранене. Те позволяват настройка на точни

количества храна, което е особено важно за животни с конкретни хранителни нужди или здравословни проблеми.

Безопасност:

- Много модели автоматизирани системи за хранене имат вградени механизми за предотвратяване на преяждане и защита на храната. Това е особено важно за домашни любимци, които са склонни към преяждане. Например, някои устройства имат функции за бавно хранене, които разпределят храната в малки порции за определен период от време, предотвратявайки бързото погъщане на големи количества храна. Освен това, хранилките са проектирани така, че да бъдат устойчиви на опити за преобръщане или разбиване от животните, осигурявайки безопасност и запазване на храната.

Автоматизираните системи за хранене на домашни любимци предоставят спокойствие на собствениците, които могат да бъдат сигурни, че техните любимци ще бъдат нахранени точно по график, дори когато те не са у дома. Много от съвременните модели могат да се свързват със смарт технологии, което позволява дистанционно управление и наблюдение чрез мобилни приложения, предоставяйки допълнителен контрол и гъвкавост.

Недостатъци

Автоматизираните системи за хранене на домашни любимци предоставят значителни предимства, но също така идват с някои недостатъци, които трябва да се вземат предвид. Ето някои от основните недостатъци на тези устройства:

Цена:

- Един от основните недостатъци на висококачествените автоматизирани системи за хранене е тяхната висока цена. Моделите с множество функции като Wi-Fi свързаност, камери и прецизни настройки за хранене могат да бъдат значително по-скъпи в сравнение с обикновените хранилки. Това може да бъде пречка за много собственици на домашни любимци, особено за тези с по-ограничен бюджет.

Сложност:

- Някои автоматизирани системи за хранене изискват значително време за настройка и програмиране. Устройствата с много функционалности често идват с подробни инструкции и множество настройки, които могат да бъдат сложни за разбиране и използване от по-малко технологично напреднали

потребители. Това може да доведе до разочарование и грешки при програмирането, което от своя страна може да повлияе на правилното хранене на домашните любимци.

Електрозависимост:

- За да функционират, автоматизираните системи за хранене се нуждаят от електрозахранване. При прекъсване на захранването, устройството може да спре да работи, което може да доведе до пропускане на хранения и стрес за домашните любимци. Въпреки че някои модели са оборудвани с резервни батерии, те не винаги гарантират продължителна работа при дълготрайни прекъсвания на захранването. Зависимостта от електрозахранване също означава, че потребителите трябва да следят състоянието на батерийте и да ги сменят редовно, за да избегнат неочеквани проблеми.

Ограничения на храната:

- Много автоматизирани системи за хранене са предназначени само за суха храна и не могат да работят с мокра или полу-влажна храна. Това ограничава възможностите за хранене на домашните любимци, особено за тези, които изискват специални диети или предпочитат мокра храна. Освен това, някои модели имат по-малък капацитет за съхранение на храна, което означава, че собствениците трябва често да зареждат хранилките, особено ако имат по-големи домашни любимци или няколко животни.

Поддръжка и почистване:

- Поддръжката и почистването на автоматизираните системи за хранене могат да бъдат допълнителни задачи за собствениците. Някои устройства имат сложни конструкции, които изискват разглобяване за почистване, което може да бъде неудобно и времеемко. Натрупването на храна и остатъци може да доведе до разваляне на храната и проблеми със здравето на домашните любимци, ако не се почистват редовно.

Въпреки множеството предимства, автоматизираните системи за хранене на домашни любимци не са лишени от недостатъци. Високата цена, сложността на настройка и програмиране, зависимостта от батерията и захранването, ограниченията в използваната храна и необходимостта от редовна поддръжка и почистване са фактори, които трябва да се вземат предвид при избора на такава система. Внимателното разглеждане на тези аспекти ще помогне на собствениците на домашни любимци да направят информиран избор, който най-добре отговаря на техните нужди и възможности.

Глава 2

Планиране

2.1 Софтуерни технологии

В съвременния свят на свързани устройства и интелигентни решения, Интернет на нещата (IoT) играе ключова роля в автоматизацията и управлението на различни процеси. Създаването на автоматизирана IoT система за хранене на домашни любимци изисква избор на подходящи софтуерни технологии и хардуерни компоненти, които да осигурят надеждна и ефективна работа на системата.

2.1.1 IoT платформи

Blynk

Blynk е една от най-популярните IoT платформи, която предоставя лесен начин за свързване на хардуерни устройства и управление на техните функции с опция за мобилно приложение. Blynk предлага готови библиотеки за различни микроконтролери като Arduino и Raspberry Pi, което улеснява разработката на IoT проекти.

Предимства:

- Лесна настройка и употреба. Платформата Blynk е лесна за настройка и ползване, дори за потребители без задълбочени технически умения.
- Множество поддържани устройства. Платформата поддържа над 400 хардуерни модели, което я прави гъвкава и подходяща за различни проекти.
- Голямо общество и поддръжка. Blynk има активно общество от разработчици и предоставя бърза и ефективна поддръжка на клиентите си.
- Богат набор от функционалности. Включва възможности като визуализация на данни, управление на устройства, мониторинг и актуализации.

Недостатъци:

- Цена. Въпреки че платформата предлага бесплатна версия, много от напредналите функции изискват абонамент, което може да се окаже скъпо за някои потребители.
- Проблеми с връзката. Някои потребители съобщават за проблеми при свързване на приложението към мрежата.

- Ограничени функции за Bluetooth. Някои потребители са забелязали липсата на определени функционалности за Bluetooth, което може да бъде ограничение за някои проекти.[7]

ThingsBoard

ThingsBoard е мащабируема IoT платформа с отворен код, която предоставя възможности за събиране, обработка и визуализация на данни от различни устройства. Тя поддържа множество протоколи като MQTT, CoAP и HTTP, което я прави гъвкава и лесна за интеграция с различни хардуерни компоненти. ThingsBoard предлага възможности за управление на устройства, анализ на данни в реално време и създаване на интерактивни дашборди, което я прави подходяща за сложни IoT проекти.

Предимства:

- Отворен код. ThingsBoard е платформа с отворен код, която позволява персонализация и адаптиране към специфичните нужди на проекта.
- Мащабируемост. Поддържа множество протоколи, което я прави лесна за интеграция и подходяща за големи проекти.
- Добри аналитични възможности. Платформата предоставя възможности за анализ на данни в реално време и създаване на интерактивни дашборди, което я прави идеална за сложни IoT решения.

Недостатъци:

- Ограничена поддръжка за мобилни приложения. Въпреки че ThingsBoard предлага добри функции за десктоп и уеб приложения, липсва мобилно приложение, което да улесни управлението и мониторинга на устройствата през смартфони и таблети.
- Високи изисквания към хардуера. За да работи ефективно, ThingsBoard изисква значителни изчислителни ресурси и памет. Това може да увеличи разходите за хардуер, особено при мащабни инсталации и сложни проекти.
- Сложност на конфигурацията. Платформата може да бъде трудна за конфигуриране и настройка, особено за потребители без задълбочени технически знания.[8]

AWS IoT Core

AWS IoT Core е облачна платформа, предоставяна от Amazon Web Services (AWS), която позволява сигурно свързване на устройства с облака и други

устройства. Тя поддържа събиране на данни, управление на устройства и обработка на данни в реално време. AWS IoT Core е подходяща за мащабни проекти, които изискват висока надеждност и сигурност.

Предимства:

- Висока надеждност и сигурност. AWS IoT Core предлага висока надеждност и сигурност, подходяща за мащабни проекти, които изискват защита на данните и непрекъснатост на услугите.
- Множество интеграции. Поддържа интеграция с други AWS услуги, което улеснява разширяването на функционалността на IoT проектите.
- Добри възможности за анализ. Платформата предоставя добри аналитични инструменти за обработка на данни в реално време.[9]

Недостатъци:

- Висока цена. Използването на AWS IoT Core може да бъде скъпо, особено за малки и средни предприятия.
- Сложност. Настройката и управлението на платформата изискват задълбочени технически знания и опит.

2.1.2 Програмни езици

C++

Програмният език C++ е широко използван за програмиране на микроконтролери, тъй като осигурява висока производителност и ефективност, позволявайки директен контрол върху хардуера. C++ предлага обширни библиотеки, които улесняват интеграцията с различни периферни устройства, правейки го предпочитан избор за сложни вградени системи и IoT проекти.

Предимства:

- Висока производителност. C++ е компилиран език, което означава, че кодът се превежда директно в машинен език преди изпълнение. Това води до по-висока ефективност и бързина в сравнение с интерпретирани езици.
- Ниско ниво на хардуерния достъп. C++ предоставя възможност за директен достъп до хардуерните ресурси, което позволява по-добър контрол върху изпълнението на програмата и управлението на ресурсите.
- Обширна библиотека. Съществуват множество библиотеки и готови решения, които улесняват разработката на различни проекти с микроконтролери.

- Портативност. C++ кодът може да бъде пренесен лесно между различни платформи и микроконтролери с минимални промени.[10]

Недостатъци:

- Сложност. Синтаксисът и концепциите на C++ могат да бъдат сложни за начинаещите, което изиска по-дълъг период на обучение и опит.
- Отстраняване на грешки. Отстраняването на грешки и управлението на паметта могат да бъдат трудни и времеемки, особено при по-сложни програми.
- Обем на кода. Кодът на C++ може да бъде по-голям и по-труден за четене и поддръжка, особено за по-малки проекти.

Python

Програмният език Python е лесен за научаване интерпретиран език, широко използван при микроконтролери. Неговият прост синтаксис и обширна библиотека от модули улесняват бързото създаване на прототипи и интеграцията с хардуерни компоненти. Въпреки че е по-бавен от компилирани езици като C++, Python е изключително популярен за IoT и вградени системи поради своята гъвкавост и лесна употреба.

Предимства:

- Лесен за научаване. Python има прост и интуитивен синтаксис, който го прави лесен за научаване и използване, особено за начинаещи.
- Голяма библиотека. Python разполага с обширна стандартна библиотека и множество външни модули, които улесняват разработката на различни приложения.
- Гъвкавост. Python може да бъде използван за широк спектър от задачи, включително уеб разработка, анализ на данни, автоматизация и работа с хардуер.
- Поддръжка. Има голяма и активна общност от потребители и разработчици, които създават и споделят проекти, ръководства и софтуер.

Недостатъци:

- Ниска производителност. Като интерпретиран език, Python е по-бавен в сравнение с компилирани езици като C++. Това може да бъде ограничение при задачи, изискващи висока производителност.

- Изисквания към ресурсите. Python програмите обикновено изискват повече памет и мощност на процесора, което може да бъде проблем при микроконтролери с ограничени ресурси.
- Сигурност. По отношение на сигурността Python се счита за по-малко сигурен от някои други езици за програмиране като Java или C++. Това е така, защото типовете данни се определят по време на изпълнение, а не по време на компилиране. Това може да доведе до уязвимости, включително препълване на буфер или атаки чрез инжектиране.[11]

2.2 Хардуерни компоненти

2.2.1 Микроконтролери

Raspberry Pi

Raspberry Pi е серия от малки, едноплаткови компютри, разработени от Raspberry Pi Foundation с цел насърчаване на обучението по основни компютърни науки и електроника. Тези микроконтролери са проектирани да бъдат достъпни и лесни за използване, като същевременно предоставят гъвкавост за разнообразни проекти – от образователни приложения до сложни IoT системи.[12]

Предимства:

- Достъпност. Raspberry Pi е относително евтин микроконтролер, което го прави идеален за образователни цели и проекти.
- Гъвкавост. Поддържа различни операционни системи, включително специализирани дистрибуции на Linux като Raspbian.
- Компактност. Малкият му размер го прави удобен за интеграция в различни проекти, включително мобилни и преносими устройства.
- Поддръжка. Има голяма и активна общност от потребители и разработчици, които създават и споделят проекти, ръководства и софтуер.
- Съвместимост. Raspberry Pi поддържа множество интерфейси за свързване на различни сензори и периферни устройства, като GPIO, I2C, SPI и други.

Недостатъци:

- Няма вътрешна памет. Raspberry Pi използва микро SD карти за съхранение на операционната система и данни, което може да ограничи скоростта и капацитета в сравнение с вътрешната памет на стандартните компютри.

- Ограничена поддръжка на някои софтуери. Въпреки че поддържа множество операционни системи, не всички софтуерни приложения са съвместими или оптимизирани за работа на Raspberry Pi.
- Нужда от охлаждане. При по-натоварени задачи може да се наложи допълнително охлаждане, за да се предотврати прегряване.
- Ограничена производителност. Въпреки че е достатъчно мощен за много приложения, Raspberry Pi не може да се сравнява с производителността на традиционните настолни компютри. [13]

Arduino

Arduino е популярна серия от отворен код микроконтролери и хардуерни платформи, предназначени за създаване на интерактивни проекти. Разработени от Arduino.cc, тези платформи са широко използвани в образователни среди, както и от хоби ентузиасти и професионални инженери за различни прототипи и проекти. Arduino платките са базирани на лесен за използване хардуер и софтуер, които позволяват на потребителите да пишат и качват код директно в микроконтролера чрез Arduino IDE.[14]

Предимства:

- Лесни за употреба. Arduino е проектиран да бъде достъпен за начинаещи, с интуитивен софтуер и лесен за разбиране хардуер. Това го прави идеален за образователни цели и прототипи.
- Поддръжка. Arduino има голяма и активна общност от потребители и разработчици, които създават и споделят проекти, библиотеки и ръководства.
- Множество библиотеки. Arduino поддържа широк набор от библиотеки и допълнителни модули, които разширяват функционалността на платките и улесняват интеграцията с различни сензори и устройства.
- Отворен код. Хардуерът и софтуерът на Arduino са с отворен код, което позволява на потребителите да персонализират и разширяват проектите си според нуждите си.
- Достъпност. Arduino платките са достъпни и на разумни цени, което ги прави подходящи за широк кръг от потребители.

Недостатъци:

- Ограничена мощност. Въпреки че Arduino е достатъчно мощен за множество проекти, неговите процесорни и паметни ресурси са ограничени в сравнение с други по-мощни микроконтролери.

- Липса на многозадачност. Платките Arduino не поддържат многозадачност, което може да бъде ограничение за по-сложни проекти, изискващи паралелна обработка.
- Зависимост от външни компоненти. За разлика от някои други платформи, Arduino често изисква допълнителни компоненти и модули за реализиране на по-сложни функционалности.
- Ограничена свързаност. В основната си конфигурация, много от Arduino платките не разполагат с вградена Wi-Fi или Bluetooth свързаност, което може да наложи използването на допълнителни модули. [15]

2.2.2 Мотори

Постояннотокови мотори

Постояннотоковите мотори, още наричани DC мотори (DC, съкратено от Direct Current) са електрически мотори, които работят с постоянен ток. Те се използват в широк спектър от приложения поради своята проста конструкция и лесна управляемост. [16]

Предимства:

- лесни за управление и контрол
- висока ефективност при ниски обороти
- достъпни и широко използвани

Недостатъци:

- износване на четките и колектора
- ограничена продължителност на живота в сравнение с безчетковите мотори

Приложения:

- използват се в играчки, вентилатори, малки електроуреди и други

Безчеткови постояннотокови мотори

Безчетковите постояннотокови мотори, още наричани BLDC мотори (BLDC, съкратено от Brushless Direct Current) работят на постоянен ток, но не използват четки за промяна в направлението на електрическия ток. BLDC моторите се въртят с помощта на магнитни полета.[17]

Предимства:

- по-дълъг живот поради липсата на износващи се четки
- висока ефективност и производителност

- тихи и надеждни

Недостатъци:

- висока цена
- сложна интеграцията, управление и поддръжката

Приложения:

- използват се в дронове, електрически велосипеди, вентилатори с променлива скорост и други

Стъпкови мотори

Стъпковите мотори са синхронни мотори, които движат ротора си на малки стъпки, вместо продължително. Те са особено подходящи за прецизен контрол на позицията.[18]

Предимства:

- прецизен контрол на позицията
- лесни за управление с цифрови сигнали
- висок въртящ момент при ниски скорости

Недостатъци:

- загуба на синхронизация при претоварване
- по-ниска ефективност в сравнение с други типове мотори

Приложения:

- използват се в 3D принтери, CNC машини, роботика и други

Серво мотори

Серво моторите са специален тип мотори, оборудвани с обратна връзка за позицията. Предоставят възможност за прецизен контрол на линейни и ъглови позиции, както и на скоростта на завъртане.[19]

Предимства:

- висока прецизност на контрола
- висока скорост и въртящ момент
- надеждни и стабилни

Недостатъци:

- висока цена
- изискват сложна електроника за управление

Приложения:

- използват се в CNC машини, роботика, автоматизация и други

2.3 Избрани технологии

2.3.1 Софтуерни технологии

IoT платформа – Blynk

В следствие на проведеното проучване, за осъществяване на автоматизираната система за хранене на домашни любимци бе избрана IoT платформата Blynk. Основната причина за този избор е лесната настройка и употреба на платформата. Blynk предлага интуитивен интерфейс, който улеснява създаването на персонализирани мобилни приложения за управление на IoT устройства.

Blynk поддържа множество хардуерни платформи, включително популярни микроконтролери като Arduino и Raspberry Pi. Това осигурява гъвкавост и възможност за интеграция с различни хардуерни компоненти. Blynk предоставя богата библиотека от готови модули и функции, като визуализация на данни, управление на устройства и мониторинг в реално време.

Освен това, Blynk има голяма и активна общност от потребители и разработчици, които споделят проекти, идеи и предоставят поддръжка, улеснявайки решаването на проблеми и намирането на ресурси, необходими за успешната реализация на различни проекти.

Blynk предлага надеждна и сигурна платформа за управление на IoT устройства. Сигурността на данните и надеждната работа на системата са критични за проекти като автоматизираната система за хранене на домашни любимци, където непрекъснатото и правилно функциониране е от съществено значение.

Програмен език – Python

На база проведеното проучване, за настоящата дипломна работа бе избран програмният език Python. Той е известен със своя лесен за четене и разбиране синтаксис, което го прави идеален избор както за начинаещи, така и за опитни програмисти.

Програмният език Python предоставя богата библиотека от готови модули и инструменти, които улесняват работата с хардуерни компоненти. Освен това, Python разполага с огромна и активна общност от разработчици, които споделят ресурси, примери и ръководства. Това осигурява бърза и лесна поддръжка при възникване на проблеми.

Този език позволява лесно интегриране с IoT платформи и услуги, което е от съществено значение за разработката, управлението и мониторинга на

автоматизираните системи. Python е съвместим с множество IoT платформи, което улеснява свързването на различни устройства и системи.

2.3.2 Хардуерни компоненти

Микроконтролер – Raspberry Pi

След проведеното проучване, за разработката на настоящата дипломната работа бе избран микроконтролерът Raspberry Pi. Това решение бе взето поради няколко ключови причини, които правят Raspberry Pi най-подходящия микроконтролер за осъществяването на автоматизирана система за хранене на домашни любимци.

Raspberry Pi предлага значителна изчислителна мощност и производителност, сравнима с тази на настолен компьютер, но в компактен размер. Това го прави идеален за сложни задачи като обработка на данни в реално време и изпълнение на софтуер с високи изисквания.

За разлика от Arduino микроконтролерите, Raspberry Pi могат да изпълняват множество операции наведнъж. Освен това, имат вградени Ethernet и USB портове, както и Wi-Fi модул, докато при Arduino те трябва да се прикачат допълнително.

Микроконтролерът Raspberry Pi е изключително гъвкав и лесен за разширяване благодарение на множеството GPIO (съкратено от General-Purpose Input/Output) пинове и широката поддръжка на различни периферни устройства.

Поддръжката на множество програмни езици, включително Python, който бе избран за основен програмен език за проекта, позволява лесна разработка и поддръжка на софтуера, благодарение на обширната библиотека от модули и наличието на много примери и ръководства. В допълнение, Raspberry Pi може да работи с различни операционни системи, като Raspberry Pi OS, което осигурява стабилна и сигурна среда за изпълнение на проекта.

Този вид микроконтролер е достъпен и предлага много добра цена за предоставяните възможности и мощност. Това го прави подходящ за проекти с ограничен бюджет, като същевременно предоставя висока производителност.

За разработката на автоматизираната система за хранене на домашни любимци бе избран Raspberry Pi 4 модел B.

Мотор – Серво мотор

На база прегледа на различни видове мотори, за осъществяването на автоматизираната система за хранене на домашни любимци бе избран серво

мотор. Серво моторите се оказаха най-подходящия избор за този проект поради възможността за прецизен контрол на позициите, движението и скоростта.

Серво моторите предлагат висока прецизност и контрол върху ъгловото движение. Те могат да се позиционират точно в предварително зададени ъгли, което е особено важно за автоматизираната система за хранене на домашни любимици, където е необходимо прецизно управление на дозата храна.

Освен това, серво моторите са известни със своята надеждност и дълготрайност. Те са проектирани да работят безпроблемно за дълги периоди, което ги прави подходящи за системи, изискващи постоянна и надеждна работа.

За разработката на системата бе избран серво мотор S3003.

2.3.3 Допълнителни компоненти

Захранване

Осигуряването на стабилно и надеждно захранване е от съществено значение за поддържането на постоянна и ефективна работа на системата. То трябва да гарантира, че тя ще функционира безупречно и ще осигури необходимата грижа за домашния любимец.

За микроконтролера Raspberry Pi 4 модел B, бе избрано захранване тип C с входно напрежение от 100-240VAC при честота 50/60Hz, изходно напрежение от 5.1VDC и изходен ток от 3A. Максималната мощност на захранването е 15.3W, което гарантира достатъчно енергия за безпроблемна работа на Raspberry Pi и свързаните към него компоненти.

Кабели

За връзката между микроконтролера и серво мотора бяха избрани свързващи кабели тип мъжко/женско, с дължина 20 см. Тези кабели осигуряват свързването на различните компоненти на системата, осигурявайки сигурна и стабилна електрическа връзка.

SD карта

За инсталация на операционна система на микроконтролера Raspberry Pi, бе избрана SD карта, с 32 гигабайта памет, за да се подсигури достатъчно пространство, както за операционната система, така и за съхранение на данни.

Глава 3

Проектиране

3.1 Основни изисквания

3.1.1 Функционални изисквания

Потребителски интерфейс:

- Системата трябва да поддържа уеб интерфейс и мобилно приложение, за да се предоставят максимален контрол и удобство за потребителя.

Възможност за ръчно хранене:

- Потребителят трябва да има възможност да дозира храната ръчно чрез мобилно приложение и да преценява точно времето по което да нахрани своя домашен любимец, посредством бутон в приложението.

Възможност за автоматично хранене:

- На потребителя трябва да бъде предоставена и възможност за автоматично хранене, с опции за дозировка на храната и часове, в които да бъде нахранен неговия домашен любимец.

История на храненето:

- Системата трябва да съхранява история на времената, в които е било извършено хранене. Историята трябва да бъде достъпна за потребителя чрез мобилно приложение.

Известия:

- Потребителят трябва да получава известие при успешно извършено хранене, за да се избегне повторно пускане на храна по невнимание.

3.1.2 Нефункционални изисквания

Надеждност:

- Системата трябва да функционира безотказно и да осигурява надеждно дозиране на храна в правилните часове.

Скорост:

- Мобилното приложение тряба да отговаря бързо на потребителските команди и да изпраща данни към системата в реално време.

Машабируемост:

- Системата тряба да може да поддържа множество потребители и устройства без загуба на производителността.

Удобство:

- Устройството трява да предоставя възможност за лесно преместване, почистване и следене на количеството оставаща храна в контейнера.

Енергийна ефективност:

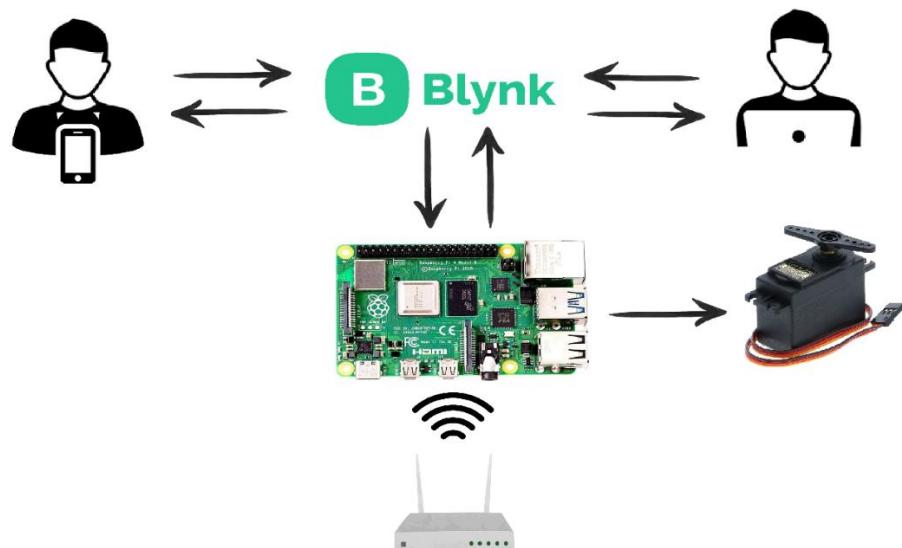
- Устройството трява да бъде енергийно ефективно, за да се минимизира разхода на електрическа енергия.

Естетика:

- Устройството трява да бъде визуално привлекателно и да се вписва в различни домашни среди, като същевременно остане лесно за употреба.

3.2 Схема на системата за хранене на домашни любимци

Изграждането на схема е ключово за планирането и реализацията на един проект. Схемата осигурява структурирано представяне на всички взаимодействия и зависимости между компонентите и подпомага за успешната разработка и поддръжка на продукта.



Фигура 3.1 Схема на системата за хранене на домашни любимци

На фигура 3.1 са представени взаимодействията между потребителя, IoT платформата Blynk, микроконтролера Raspberry Pi и серво мотора.

Потребител:

- Потребителят изпраща инструкции към IoT платформата Blynk, посредством мобилно приложение или уеб интерфейс, а след това получава известия от системата за извършените хранения.

IoT платформа (Blynk):

- IoT платформата Blynk получава инструкции от мобилното приложение и ги предава към микроконтролера Raspberry Pi. След това Blynk приема информация от микроконтролера за извършеното хранене и я предава обратно към мобилното приложение, за да информира потребителя.

Микроконтролер (Raspberry Pi):

- Микроконтролерът Raspberry Pi получава команди от IoT платформата Blynk и ги изпълнява, като управлява серво мотора, за да дозира храната. След това микроконтролера изпраща данни за извършеното хранене обратно към Blynk.

Серво мотор:

- Серво моторът е свързан към Raspberry Pi микроконтролера и изпълнява командите за дозиране на храна. Моторът получава сигнали от Raspberry Pi и се завърта в съответствие с получените инструкции, за да отвори или затвори дозатора за храна.

Интернет връзка:

- Микроконтролерът Raspberry Pi е свързан към интернет чрез рутер, използвайки Wi-Fi, което позволява комуникацията с IoT платформата Blynk и мобилното приложение.

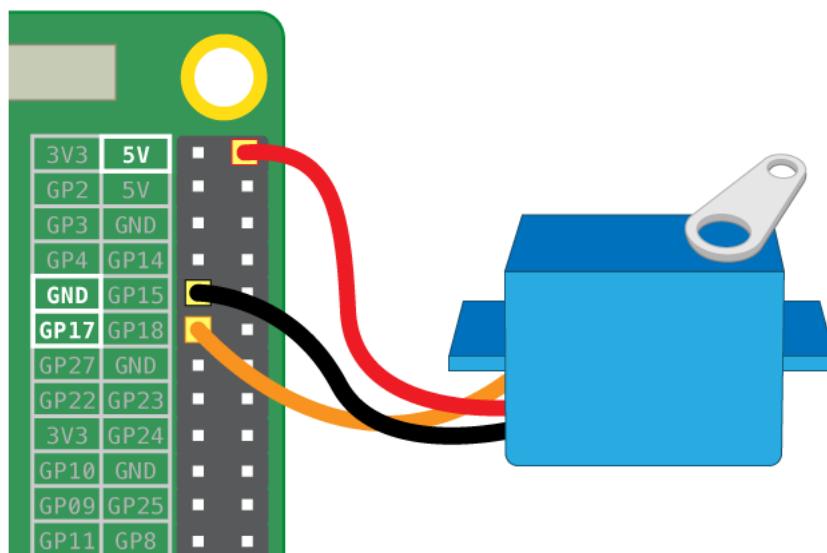
Тази комуникация гарантира, че потребителят може лесно и ефективно да управлява автоматизираната система за хранене на домашни любимци от разстояние. Потребителят може да контролира дозировката и времето за хранене чрез мобилното приложение на Blynk, което осигурява удобство и гъвкавост. Това също така позволява на системата да изпраща известия в реално време, гарантирайки, че домашният любимец получава необходимата грижа навреме.

Глава 4

Реализация

4.1 Свързване на хардуерните компоненти

За успешната реализация на системата за хранене на домашни любимци е необходимо внимателно планиране и изпълнение на връзките между различните хардуерни компоненти. Правилното им свързване е от съществено значение за осигуряването на надеждност и ефективност на системата. Предстои да бъдат описани подробно всички необходими стъпки и съображения при свързването на хардуера.



Фигура 4.1 Свързване на микроконтролера със servo мотора

За връзката между микроконтролера и серво мотора са необходими свързващи кабели тип мъжко/женско, с дължина 10cm или повече. В случая са използвани кабели с дължина 20cm, за да се осигури по-голямо удобство при употреба.

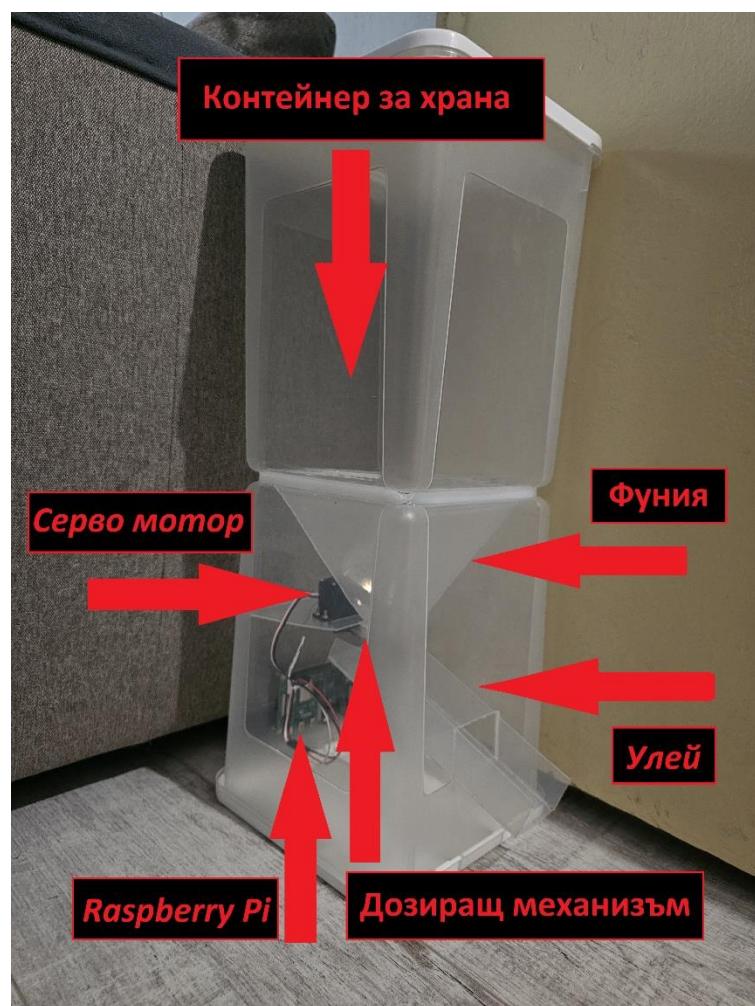
- Червеният проводник на серво мотора се свързва към 5V захранващ пин на микроконтролера Raspberry Pi.
- Черният проводник на серво мотора се свързва към GND пин за земя на микроконтролера Raspberry Pi.
- Белият проводник (илюстриран като оранжев на фигура 4.1), с който се управляват сигналите, се свърза към GPIO (съкратено от General-Purpose

Input/Output) пин на микроконтролера Raspberry Pi, като в случая е използван пин 11 (GP17).[20]

За проекта е използвано захранване тип С за Raspberry Pi, което се свързва директно към микроконтролера, осигурявайки стабилна и надеждна работа на устройството. Спецификациите на захранването включват входно напрежение от 100-240VAC при честота 50/60Hz, изходно напрежение от 5.1VDC и изходен ток от 3A. Максималната мощност на захранването е 15.3W, което гарантира достатъчно енергия за безпроблемна работа на Raspberry Pi и свързаните към него компоненти.

4.2 Изработка на контейнера за храна

Физическата част на системата за автоматизирано хранене на домашни любимци е изработена от два свързани прозрачни пластмасови контейнера, които създават функционална и естетична структура за съхранение и дозиране на храна за домашни любимци.



Фигура 4.2 Реална схема на автоматизираната система за хранене

Контейнер за храна:

- Горният контейнер служи като основен резерв за храната, осигурявайки достатъчен капацитет за съхранение. Той има изрязано дъно, което позволява на храната да преминава към долния контейнер.

Фуния и връзка между контейнерите:

- Дъното на горния контейнер е свързано с фуния, изработена от пластмаса, която насочва храната към долния контейнер. Фуниятата гарантира, че храната се движи плавно и контролирано от основния контейнер към дозиращия механизъм. В края на фуниятата е разположена пластмасова плоскост, която служи като клапан, контролиращ потока на храната.

Дозиращ механизъм:

- Пластмасовата плоскост в края на фуниятата е свързана със серво мотора, който управлява движението ѝ. Когато се подаде команда от IoT платформата Blynk, серво моторът отмества пластмасовата плоскост встрани, откривайки отвора на фуниятата и позволявайки на храната да се спусне към долния контейнер.

Долен контейнер и улей за храна:

- В долния контейнер е разположен улей, който е поставен под определен ъгъл, за да позволява на храната да се плъзга лесно след като изпадне от фуниятата и да излиза от контейнера, попадайки в купата на домашния любимец. Това гарантира, че храната достига до домашния любимец по безопасен и сигурен начин. В долния контейнер са монтирани и хардуерните компоненти.

Интеграция на електронни компоненти:

- В долния контейнер са монтирани както серво моторът, така и микроконтролерът Raspberry Pi. Благодарение на капака на дъното на долния контейнер се осигурява лесен достъп до всички необходими компоненти за управление на системата. В контейнера е предвидена и дупка за преминаване на захранващия кабел.

Този дизайн на контейнера за храна осигурява ефективно и удобно решение за автоматизирано хранене на домашни любимици, като гарантира надеждност и прецизност при дозиране на храната. Прозрачността на контейнера позволява на потребителите лесно да виждат колко храна има останала вътре и дали е

необходимо да се добави още, което улеснява поддържането на постоянни нива на храната.

4.3 Конфигурация на IoT платформата Blynk

Blynk е IoT платформа, която позволява на потребителите си да управляват IoT устройства чрез мобилно приложение. Правилната конфигурация на платформата е от съществено значение за осигуряването на ефективна комуникация между потребителя и устройството.

Предстои да бъде разгледан процеса на конфигурация на IoT платформата Blynk за автоматизираната система за хранене на домашни любимци.

Регистрация

Процесът на регистрация в IoT платформата Blynk е първата стъпка към създаването на интелигентната система. За може да се осъществи реализацията на автоматизираната система за хранене на домашни любимци бе създаден бесплатен акаунт за платформата Blynk.

Зона за разработчици

За започване на работа по проекта, първо е необходимо да се влезе в зоната за разработчици на Blynk. Тази зона е важен компонент, който позволява на потребителите да управляват и наблюдават своите IoT устройства и проекти ефективно. Тя предоставя множество инструменти и функции, които улесняват управлението на проекта.

Шаблони

Blynk шаблоните представляват структури, които позволяват лесното управление и конфигуриране на IoT устройства чрез стандартни и предварително зададени параметри. Включват настройки и компоненти като виртуални пинове, дашборди, автоматизации и други функции. Шаблоните също така позволяват създаването на автоматизирани сценарии, базирани на условия и действия.

Шаблоните позволяват централизирано управление на множество устройства, като всяко устройство, свързано с конкретен шаблон, наследява неговите настройки и функционалности. Това прави процеса на добавяне и конфигуриране на нови устройства много по-лесен и бърз.

В рамките на един шаблон могат да се създадат дашборди с различни визуални елементи като бутони, графики, текстови полета и други. Тези дашборди се настройват веднъж в шаблона и се използват за всички устройства, свързани с него.

За целта на проекта бе създаден шаблон именован „Automatic pet feeder”.

Дашборди

Дашбордите в Blynk, още наричани табла, са важна част от платформата, която позволява на потребителите да визуализират данни и да контролират IoT устройства си чрез удобен и интуитивен интерфейс. Дашбордите са част от всеки шаблон и предоставят възможност за създаване на персонализирани графични интерфейси, които могат да включват различни видове компоненти за мониторинг и управление.

Компоненти

Компонентите в платформата Blynk представляват предварително проектирани елементи за графичния потребителски интерфейс, които позволяват на потребителите да взаимодействат с IoT устройства по различни начини. Всеки компонент изпълнява специфична функция за вход или изход, комуникирачки с хардуера.

За да се осигури функционалността на автоматизираната система за хранене на домашни любимци, в дашборда бяха добавени следните пет компонента:

- бутон за превключване на режима на хранене между ръчен и автоматичен
- бутон за извършване на хранене при ръчен режим
- плъзгащ се бутон за избор на сутрешен час на хранене при автоматичен режим
- плъзгащ се бутон за избор на вечерен час на хранене при автоматичен режим
- плъзгащ се бутон за определяне на количеството храна което да бъде пуснато

Потоци от данни

Потоците от данни в Blynk са основни канали, чрез които данните се пренасят между IoT устройства и облака на Blynk. Те осигуряват структура за

регулярен поток на данни, като данните могат да бъдат събиращи от хардуерни компоненти, обработвани от микроконтролер и визуализирани и управлявани чрез Blynk.

За да се придае функционалност на петте компонента, за всеки един от тях бе създаден отделен поток от данни:

- за промяна на режима на хранене между ръчен и автоматичен
- за извършване на хранене при ръчен режим
- за задаване на сутрешен час на хранене при автоматичен режим
- за задаване на вечерен час на хранене при автоматичен режим
- за определяне на количеството храна което да бъде пуснато

За всички потоци от данни е включена настройката за синхронизация с последната стойност от сървъра, при всяко свързване на устройството с облака на Blynk. Тази настройка е полезна, тъй като гарантира, че всяко устройство получава най-актуалните стойности на всички параметри веднага след свързване с облака. Това предотвратява проблеми с остатели или несинхронизирани данни и осигурява надеждност при повторно свързване след временна загуба на интернет връзка.

Включена е и настройката за история на данните. Тя осигурява съхранение на исторически данни, което позволява анализа на събраната информация във времето. Предоставя възможност за проследяване на събития и откриване на проблеми.

Типът на данните при всеки един от потоците е зададен на цяло число. Зададени са също така минимални, максимални и стойности по подразбиране. За потоците от данни за промяна на режима и извършване на хранене при ръчен режим са зададени „0“ за минимална стойност, „1“ за максимална стойност и „0“ за стойност по подразбиране. За потоците от данни за настройка на количеството и часа на хранене са зададени стойностите както следва: за количество храна – „1“ за минимална, „10“ за максимална и „1“ по подразбиране; за сутрешен час – „0“ за минимална, „12“ за максимална и „0“ по подразбиране; за вечерен час – „13“ за минимална, „23“ за максимална и „13“ по подразбиране. Това позволява на потребителя да задава часове на хранене и да избира степен за количеството храна, което да бъде пуснато на неговия домашен любимец.

Потоците от данни за избор на сутрешен и вечерен час имат зададени мерни единици, а именно часове, докато останалите потоци нямат.

Всеки един от потоците е свързан с индивидуален виртуален пин, което позволява изпращане и получаване на данни между устройството и Blynk.

След създаването на потоците от данни, всеки един от тях се свързва с компонентите (бутоните) в дашборда.

Събития и известия

Blynk предоставя система за управление на събития и известия, която позволява на потребителите да бъдат информирани в реално време за важни събития, свързани с техните IoT устройства. Събитията могат да бъдат настроени да изпращат известия по различни канали, включително имейли, известия на смартфони или SMS.

За да може потребителя да получава известие при извършено хранене, бе създадено събитие, което изпраща известие към мобилното устройство на потребителя и го информира, че домашният му любимец е бил нахранен.[21]

За събитието е включена настройката за показване на известие на телефона на потребителя, както и настройката за изпращане на събитието към линията на събитията.

Линия на събитията

Линията на събитията в Blynk е инструмент, който съхранява и визуализира всички събития, свързани с IoT устройствата в мрежата, в хронологичен ред. Тя предоставя преглед на събитията и позволява на потребителите да видят кога и какво се е случило с устройствата им. Всички събития се показват в последователност, започвайки от най-новото към най-старото. Това улеснява проследяването на събитията и разбирането на тяхната времева рамка.

Ограничения на платформата

Blynk е мощна IoT платформа, която предлага разнообразни функции за управление и наблюдение на устройства в реално време. Въпреки своите предимства обаче, платформата има някои ограничения, особено в бесплатната ѝ версия:

- ограничение от 100 събития на устройство на ден
- не позволява да бъдат използвани повече от 5 компонента
- не предоставя голям избор на видове компоненти

Въпреки ограниченията в бесплатната версия на Blynk, системата за хранене на домашни любимици включва всички необходими компоненти и функционира

според очакванията. Ограничението от 100 събития на устройство на ден и ограниченият избор на компоненти не възпрепятстват основната функционалност на системата. Всички основни функции са налични и системата работи ефективно.

4.4 Реализация на мобилно приложение чрез Blynk

За реализацията на мобилния потребителски интерфейс е необходимо да се свали приложението „Blynk IoT“ от Google Play Store или Apple App Store. След инсталацията трябва да се влезе във вече създадения акаунт и да се използва съществуващия шаблон.

След изпълнението на стъпките по-горе е необходимо да бъдат добавени същите компоненти, като тези присъстващи в уеб интерфейса и да бъдат свързани с вече съществуващите потоци от данни. Компонентите могат да бъдат с персонализирани размери, цветове и разположение.

За реализацията на мобилния потребителски интерфейс на системата за автоматизирано хранене на домашни любимци, бе свалено приложението „Blynk IoT“ от Google Play Store, като бяха използвани вече създадените акаунт и шаблон. В мобилния дашборд бяха добавени и персонализирани компонентите, които присъстват и в уеб интерфейса, а след това бяха свързани със съответните потоци от данни. Така мобилното приложение бе успешно реализирано посредством IoT платформата Blynk.

4.5 Реализация на функционалността в Raspberry Pi

4.5.1 Първоначална инициализация на Raspberry Pi

За да може да се започне работа по проекта, бяха извършени първоначални конфигурации и настройки на микроконтролера Raspberry Pi.

Посредством програмата „Raspberry Pi Imager“[22], на SD картата бе инсталирана 64-битовата операционна система Raspberry Pi OS, базирана на Debian, а след това картата бе поставена в микроконтролера. Преди инсталацията бяха зададени стойности както за потребител и парола, така и за име и парола на Wi-Fi мрежата, която ще бъде използвана.

4.5.2 Връзка с микроконтролера

Връзката с микроконтролера Raspberry Pi бе осъществена посредством програмата “TigerVNC“. Тази програма представлява клиент за VNC (съкратено от

Virtual Network Computing), който позволява отдалечно управление на графичната среда на компютър, в случая микроконтролер, използвайки VNC протокол.

За осъществяване на връзка с Raspberry Pi е необходимо да се стартира програмата TigerVNC и в нея да се въведе IP адреса на микроконтролера. IP адресът може да се намери в списъка с всички свързани устройства в мрежата.

4.5.3 Програмна реализация

Предстои да бъде разгледана програмната реализация на автоматизираната система за хранене на домашни любимци, разработена посредством избрания програмен език – Python.

Виртуална среда на Python

Бе използвана функционалността на Python „venv” (съкратено от virtual environment) за създаване на специализирана виртуална среда, с цел изолиране на проекта и използваните пакети от глобалната инсталация на Python. По този начин бяха предотвратени потенциални конфликти с други проекти.[23]

За да се улесни управлението на всички библиотеки и пакети, необходими за функционирането на системата, бе създаден файлът “requirements”. Този файл гарантира, че всички разработчици използват едни и същи версии на Python библиотеките, елиминирайки потенциални проблеми и несъвместимости. Чрез команда „pip install -r requirements.txt“ всички необходими пакети могат да бъдат инсталирани наведнъж, което спестява време и усилия при настройката на нова среда.[24]

Библиотеки

Използвани са библиотеките BlynkLib (от пакета blynk-library-python[25]) и RPi.GPIO, съответно за взаимодействие с IoT платформата Blynk и за управление на GPIO (General-Purpose Input/Output) пиновете на Raspberry Pi микроконтролера.

Основни константи и променливи

BLYNK_AUTH:

- Всяко устройство притежава уникален токен за аутентикация, който се получава при неговото първоначално регистриране в Blynk платформата.

GPIO_SERVO_CONTROL_PIN:

- Номера на GPIO пина, от който ще бъде управляван серво моторът.

GPIO_SERVO_FREQ:

- Честота на PWM (съкратено от Pulse Width Modulation) сигнала за серво мотора. PWM позволява да се контролира широчината на импулсите в един период на сигнала.[26]

GPIO_SERVO_OPEN и GPIO_SERVO_CLOSE:

- Стойности за широчината на импулса в рамките на един период, които определят завъртането на мотора за отворено и затворено състояние.

servoOpenTime:

- Времето, в секунди, за което серво моторът остава отворен, в зависимост от зададената степен на хранене.

TIMER_MIN_INTERVAL и TIMER_MAX_INTERVAL:

- Интервали за проверка на графика за хранене, позволяващи по-голяма гъвкавост за честотата на извършваните проверки.

BLYNK_OPERATION_MODE_VPIN, BLYNK_MANUAL_FEED_VPIN и т.н.:

- Виртуални пинове за комуникация с IoT платформата Blynk. Номерацията им отговаря на съответните потоци от данни, дефинирани в Blynk шаблона.

operationMode, morningScheduleHour, eveningScheduleHour и feedTimer:

- Променливи за управление на режима и графика за хранене.

Инициализация

Функцията „initialize_gpio()“ настройва GPIO пина за управление на серво мотора и стартира PWM сигнала. Задава се режима на номериране на GPIO пиновете, а след това се задава пинът за управление на серво мотора да работи като изходящ. Накрая се настройва честотата на PWM сигнала и първоначалната широчина на импулса.

```
def initialize_gpio():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(GPIO_SERVO_CONTROL_PIN, GPIO.OUT)
    servoControl = GPIO.PWM(GPIO_SERVO_CONTROL_PIN, GPIO_SERVO_FREQ)
    servoControl.start(0)
    return servoControl
```

Фигура 4.3 Инициализация на GPIO пиновете

Извиква се функцията „initialize_gpio()“, която връща обект, чрез който може да се управлява PWM сигнала. Инициализира се и Blynk библиотеката, като се подават токена за аутентикация и адреса на облачната платформа.

```
servoControl = initialize_gpio()
blynk = BlynkLib.Blynk(BLYNK_AUTH, server='blynk.cloud', log=print)
```

Фигура 4.4 Инициализация на Blynk библиотеката

Дефиниция на Blynk функции

Посредством декоратора „@blynk.ON("connected")“ е дефинирана функцията „blynk_connected()“, която да бъде извиквана автоматично когато устройството се свърже и аутентицира към облака. Функцията синхронизира виртуалните пинове с актуалните им стойности от облака.

```
@blynk.ON("connected")
def blynk_connected():
    print("Syncing virtual pin values from cloud...")
    blynk.sync_virtual(
        BLYNK_OPERATION_MODE_VPIN,
        BLYNK_MORNING_HOUR_VPIN,
        BLYNK_EVENING_HOUR_VPIN,
        BLYNK_FEED_RATE_VPIN
    )
```

Фигура 4.5 Функция за синхронизация на виртуалните пинове

Чрез декоратора „@blynk.VIRTUAL_WRITE(BLYNK_OPERATION_MODE_VPIN)“ е дефинирана функцията „on_operation_mode_change(value)“ за следене на виртуалния пин за режима на работа. Когато настъпи промяна, тя се отразява в съответната променлива и в таймера за автоматично хранене.

```

@blynk.VIRTUAL_WRITE(BLYNK_OPERATION_MODE_VPIN)
def on_operation_mode_change(value):
    global operationMode

    if int(value[0]) == 1:
        operationMode = 'automatic'
        createTimer()
    else:
        operationMode = 'manual'
        clearTimer()

```

Фигура 4.6 Функция за обработка на промени на режима на хранене

С помощта на „@blynk.VIRTUAL_WRITE(BLYNK_MANUAL_FEED_VPIN)“ декоратора е дефинирана функцията „on_manual_feed(value)“, която да бъде изпълнявана при получаване на промяна от „0“ към „1“ за стойността на пина за ръчно хранене. При изпълнение тя също така стартира и функцията за хранене, която пропуска храна към купата на домашния любимец.

```

@blynk.VIRTUAL_WRITE(BLYNK_MANUAL_FEED_VPIN)
def on_manual_feed(value):
    if int(value[0]) == 1 and operationMode == 'manual':
        print('Performing manual feed...')
        run_feed_procedure()

```

Фигура 4.7 Функция за изпълнение на хранене при ръчен режим

Дефинирани са две функции за обработка на промени в сутрешния и вечерния час на хранене при автоматичен режим. Освен да запазват новите стойности, двете функции нулират и отново стартират таймера.

```

@blynk.VIRTUAL_WRITE(BLYNK_MORNING_HOUR_VPIN)
def on_morning_schedule_change(value):
    global morningScheduleHour
    morningScheduleHour = int(value[0])
    if operationMode == 'automatic':
        clearTimer()
        createTimer()

```

```

@blynk.VIRTUAL_WRITE(BLYNK_EVENING_HOUR_VPIN)
def on_evening_schedule_change(value):
    global eveningScheduleHour
    eveningScheduleHour = int(value[0])
    if operationMode == 'automatic':
        clearTimer()
        createTimer()

```

Фигура 4.8 Функции за обработка на промени в сутрешния и вечерния час на хранене при автоматичен режим

Чрез декодера „@blynk.VIRTUAL_WRITE(BLYNK_FEED_RATE_VPIN)“ е дефинирана функцията „on_feed_rate_change(value)“, обновяваща степента на хранене и съответно количеството храна, което да бъде пропуснато. Времето за което серво моторът да остава в отворено състояние се определя по линейната формула „степен * 0.5 секунди“.

```
@blynk.VIRTUAL_WRITE(BLYNK_FEED_RATE_VPIN)
def on_feed_rate_change(value):
    global servoOpenTime

    servoOpenTime = int(value[0]) * 0.5

    print('servoOpenTime changed to {}'.format(servoOpenTime))
```

Фигура 4.9 Функция за обработка на промени на степента на хранене

Главна функция за хранене

Храненето се осъществява като серво моторът се завърти в отворено състояние, при което отворът на фунията се открива и позволява пропускане на храната. Изчаква се определено време, отговарящо на избраната степен на хранене, след което моторът се завърта обратно в затворено състояние и се изпраща събитие към Blynk. След като получи събитието, Blynk изпраща известие на потребителя за извършеното хранене.

```
def run_feed_procedure():
    servoControl.ChangeDutyCycle(GPIO_SERVO_OPEN)
    sleep(servoOpenTime)
    servoControl.ChangeDutyCycle(GPIO_SERVO_CLOSE)
    sleep(1)
    servoControl.ChangeDutyCycle(0)
    blynk.log_event('pet_has_been_fed')
```

Фигура 4.10 Функция за извършване на хранене

Таймер

Дефинирана е функцията „createTimer(sleepTime = TIMER_MIN_INTERVAL)“, която създава обект от тип Timer, като след определено време този таймер извика функцията „check_feed_schedule()“.

```

def createTimer(sleepTime = TIMER_MIN_INTERVAL):
    print('Create feed timer in {} seconds'.format(sleepTime))
    global feedTimer

    feedTimer = Timer(sleepTime, check_feed_schedule)
    feedTimer.start()

```

Фигура 4.11 Таймер за проверка на часа

Във функцията „check_feed_schedule()“ се взима текущото време и се сравнява със зададените часове за сутрешно и вечерно хранене. Ако някой от тези часове съвпадне с текущото време, се извиква функцията за извършване на хранене. На края на функцията отново се стартира таймер за следваща проверка. Като време за изчакване на таймера се използва или TIMER_MIN_INTERVAL (когато не е било извършено хранене) или TIMER_MAX_INTERVAL (когато е било извършено хранене).

```

def check_feed_schedule():
    print('Check feed schedule')
    global morningScheduleHour
    global eveningScheduleHour

    currTime = localtime()

    hasFed = False

    print('Check morning time {} == {}'.format(currTime.tm_hour, morningScheduleHour))
    if currTime.tm_hour == morningScheduleHour:
        print('Trigger morning feed time at {}:{}'.format(currTime.tm_hour, currTime.tm_min))
        run_feed_procedure()
        hasFed = True

    print('Check evening time {} == {}'.format(currTime.tm_hour, eveningScheduleHour))
    if currTime.tm_hour == eveningScheduleHour:
        print('Trigger evening feed time at {}:{}'.format(currTime.tm_hour, currTime.tm_min))
        run_feed_procedure()
        hasFed = True

    createTimer(TIMER_MAX_INTERVAL if hasFed else TIMER_MIN_INTERVAL)

```

Фигура 4.12 Функция за проверка на часа

Основен цикъл

В основния цикъл се изпълнява функцията „`blynk.run()`“, която поддържа връзката с Blynk сървъра и обработва събитията. Когато се получи грешка в програмата или настъпи външно прекъсване се изчиства настройката на GPIO пиновете и създадения таймер.

```
try:  
    while True:  
        blynk.run()  
    except:  
        print("An exception occurred. Shutting down ...")  
        servoControl.stop()  
        GPIO.cleanup()  
        clearTimer()
```

Фигура 4.13 Основен цикъл, поддържащ връзка с Blynk сървъра

4.6 Тестване

Разработената IoT система за хранене на домашно любимци бе тествана чрез различни видове тестове, включително:

Функционално тестване:

- Проверено беше дали системата изпълнява основните си функции, като ръчно и автоматично хранене по зададен график.

Тестване на интеграцията:

- Проверени бяха комуникацията и взаимодействието между Raspberry Pi, серво мотора и Blynk приложението.

Тестване за издръжливост:

- Системата бе подложена на продължителна работа, за да се гарантира, че може да функционира стабилно за дълги периоди от време.

Тестване на потребителския интерфейс:

- Извършени бяха тестове за използваемост на Blynk приложението, за да се потвърди, че потребителите могат лесно да управляват и настройват системата. Беше тествано и дали потребителят получава известия за извършените хранения.

Тестване на производителността:

- Тествано бе времето за реакция на системата при различни сценарии на натоварване и честота на храненето, за да се подсигури безотказната ѝ работа.

Тестване на хардуера:

- Проверено бе дали серво моторът и микроконтролерът са свързани правилно и здраво, както и дали изпълняват работата си коректно.

Тестване на мрежата:

- Тествана беше стабилността на мрежовата връзка и способността на системата да работи при различни състояния на мрежата.

Поетапно тестване:

- След всяка промяна в програмния код или конфигурацията на IoT платформата Blynk, бяха провеждани тестове, за да се гарантира, че новите промени не нарушават съществуващата функционалност.

Тестване с различни видове храна:

- Системата бе тествана с различни видове храна за домашни любимци, за да се увери, че механизъмът за хранене работи безпроблемно с различни типове гранули.

Тестване на наклона на улея:

- Проведени бяха тестове с различни наклони на улея, докато се избере подходящия наклон. Това осигурява безпроблемното плъзгане на храната към купата, като се гарантира, че храната не засяда.

Тестване на завъртането на серво мотора:

- Проведени бяха тестове, за да се определи оптималното завъртане на серво мотора при различните степени на хранене.

Тестване на обема:

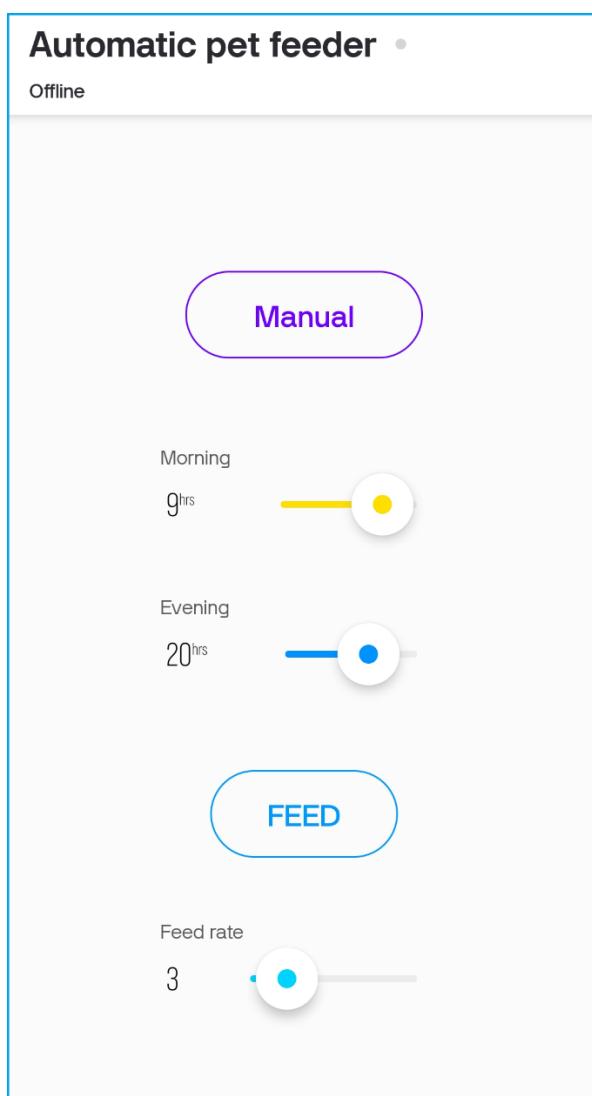
- Беше проведен тест с максимално пълен горен контейнер, за да се подсигури функционалността на системата, при пълно натоварване. Бяха проведени тестове, за да се гарантира, че механизъмът за хранене може да обработва и доставя храна без проблеми, дори когато контейнерът е напълно зареден.

Глава 5

Представяне на готовия продукт

5.1 Мобилно приложение

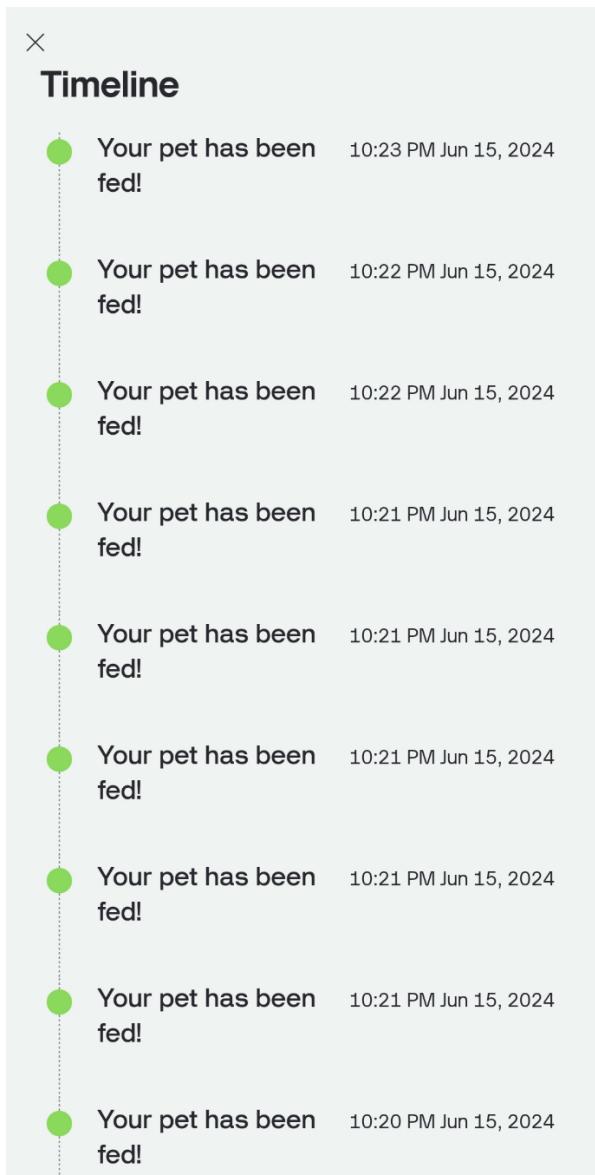
С помощта на IoT платформата Blynk бе реализирано мобилно приложение за потребителите на автоматизираната система за хранене на домашни любимци. Приложението позволява лесно управление на системата, като предоставя различни функционалности за настройка и контрол. Потребителите могат да задават график за хранене, да извършват ръчно хранене и да регулират количеството на подаваната храна.



Фигура 5.1 Мобилен потребителски интерфейс

На фигура 5.1 се вижда, че приложението е в онлайн режим, тъй като системата за хранене на домашни любимци е изключена. Когато системата бъде включена, приложението преминава в активен режим и става функционално.

След успешно извършено хранене, потребителят получава известие на мобилния си телефон, а часа в който е бил нахранен неговия домашен любимец се записва във времева линия в приложението, както е показано на фигура 5.2.



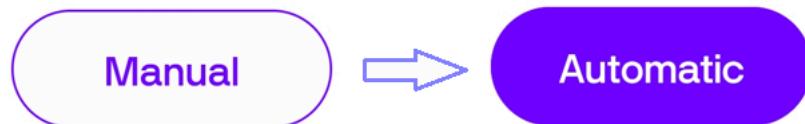
Фигура 5.2 История на храненията (времева линия)

Описание на бутоните

Manual/Automatic:

- Този бутон превключва системата между ръчен и автоматичен режим на работа, когато бъде натиснат, както може да се види на фигура 5.3. По този начин потребителят може да избере между това дали да контролира ръчно

храненето на неговия домашен любимец или да зададе график по който храненето да бъде извършвано автоматично.



Фигура 5.3 Превключване на режима на хранене

Morning:

- Този плъзгач позволява настройка на часа за сутрешно хранене. Потребителят може да избере конкретния час, в който автоматизираната система да подава храна сутрин.

Evening:

- Този плъзгач се използва за настройка на часа за вечерно хранене. Потребителят може да зададе конкретния час, в който системата да подава храна вечер.

FEED:

- Чрез този бутоон потребителят може да храни домашния си любимец, когато системата е в ръчен режим. При натискане на бутона, системата незабавно изпълнява процедурата за хранене, като подава зададеното количество храна в купата на домашния любимец.

Feed rate:

- Този плъзгач позволява регулиране на количеството храна, което се подава при всяко хранене. Потребителят може да настройва степента на подаване, за да контролира порцията храна в зависимост от нуждите на домашния му любимец.

Мобилното приложение, реализирано с Blynk, осигурява лесен и интуитивен интерфейс, който прави управлението на автоматизираната система за хранене на домашни любимци удобно и ефективно.

5.2 Автоматизирана система за хранене на домашни любимици

Разработената автоматизирана система, представлява интелигентно устройство, което улеснява грижата за домашните любимци, като осигурява редовно и точно хранене. Системата е проектирана да бъде надеждна, лесна за използване и да предлага гъвкави опции за настройка, съобразени с нуждите на различни видове домашни любимци. Тя включва мобилно приложение, което позволява на собствениците да контролират процеса на хранене от разстояние. Системата изпраща известия при всяко извършено хранене, осигурявайки спокойствие и сигурност, че домашният любимец е добре нахранен.



Фигура 5.4 Автоматизираната система за хранене в реална обстановка

Основният контейнер за храна е прозрачен, което позволява лесно да се вижда колко храна остава и кога е необходимо да бъде добавена още. По този начин се гарантира, че домашния любимец на потребителите никога няма да остане без храна.

Контейнерът за съхранение на храна осигурява достатъчно количество за няколко дни. Както може да се види по-долу на фигура 5.5, контейнерът е свързан с фуния и наклонен улей, който води храната към купата на домашния любимец. Наклонът на улея е оптимизиран, за да се осигури безпроблемно пълзгане на храната към купата.

Когато се активира процесът на хранене, серво моторът завърта пластмасовата плоскост, която блокира фунията и позволява пропускане на храна по улея към купата. След успешно извършено хранене, потребителят получава известие на мобилния си телефон, а часа в който е бил нахранен неговия домашен любимец се записва във времева линия в приложението.



Фигура 5.5 Функционалност на автоматизираната система за хранене

В мобилното приложение потребителят може да настрои график за хранене на домашния си любимец. Пъзгачите „Morning“ и „Evening“ позволяват задаване на конкретни часове за сутрешно и вечерно хранене. Потребителят може да регулира количеството храна, което се подава при всяко хранене, чрез „Feed rate“ пъзгача. Тази функционалност осигурява прецизно дозиране на храната в зависимост от нуждите на домашния любимец.

Системата поддържа както автоматично, така и ръчно хранене. При автоматичния режим системата подава храна в зададените часове, като проверява текущото време и активира механизма за хранене. Потребителя има възможност да активира ръчния режим по всяко време и да нахрани домашния си любимец посредством “FEED” бутона в мобилното приложение.



Фигура 5.6 Автоматизираната система за хранене в действие

Реализираната автоматизирана система за хранене осигурява редовно и точно подаване на храна, улеснявайки грижата за домашните любимци и осигурявайки им здравословен начин на живот. Тя намалява стреса и притесненията на собствениците, свързани с храненето на техните любимци, като предоставя удобство и сигурност.

Заключение

Разработена бе автоматизирана IoT система за хранене на домашни любимци от разстояние, която покрива всички поставени функционални и нефункционални изисквания. Системата може да се управлява посредством мобилно приложение и предоставя удобство и надеждност на потребителите, осигурявайки редовно и точно хранене на техните домашни любимци.

Преди започване на работа по проекта, бе проведено проучване на вече съществуващи подобни продукти, за да се разберат нуждите и очакванията на потребителите. Бяха разгледани софтуерни технологии, с помощта на които да се създаде IoT системата за хранене на домашни любимци. Резултатите от тези проучвания послужиха като основа за формулирането на изискванията и спецификациите на проекта, гарантирайки, че разработената система ще отговаря на нуждите на потребителите.

Бяха проведени обширни тестове, за да се гарантира стабилността и издръжливостта на устройството при различни условия и натоварвания. Освен това, бе изследвано как различни видове храна за домашни любимци взаимодействват с механичните компоненти, като се обърна особено внимание на надеждността и устойчивостта на използваните материали и компоненти.

Мобилното приложение отговаря бързо на потребителските команди и изпраща данни към системата в реално време, осигурявайки незабавен контрол и обратна връзка. В допълнение, след всяко извършено хранене, потребителят получава известие на мобилния си телефон, а датата и часа на храненето се запазват във времева линия в приложението.

Автоматизираната система за хранене на домашни любимци, в крайния си вид, предоставя на потребителя възможност да храни домашния си любimeц от разстояние, посредством мобилно приложение. Потребителят може да избира между ръчен или автоматичен режим, като храни любimeца си посредством бутон в приложението или чрез предварително зададен график.

Бъдещото развитие на системата може да включва осигуряване на резервно захранване, посредством батерия, която да подсигури непрекъсната работа дори при прекъсване на електрозахранването. Също така, могат да бъдат интегрирани сензори за следене на нивото на храната в контейнера и да бъдат изпращани известия към потребителя, когато храната намалява. Друго възможно развитие на системата би било добавянето на видеонаблюдение, което ще позволи на потребителите да наблюдават своите домашни любимци в реално време, осигурявайки им допълнителен контрол и спокойствие.

**ДЕКЛАРАЦИЯ
ЗА АВТОРСТВО НА ДИПЛОМНА РАБОТА**

Долуподписаният/ата:

Виктория Радославова Радкова

(име, презиме, фамилия)

специалност: Информатика и софтуерни науки

фак. № 471220010

ДЕКЛАРИРАМ:

Представената от мен дипломна работа на тема:

Автоматизирана купичка за хранене на домашен любимец

е лична моя авторска разработка, резултат от собствени изследвания.

Потвърждавам, че тя в нейната цялост и отделни части не е била използвана за придобиване на образователна и/или научна степен в ТУ - София или в други университети.

Формулировки, идеи и текстове, взети от други източници, са цитирани точно и с коректно посочване на техните автори. Дипломната работа не е публикувана на друго място.

Декларирам, че предоставям правото на Факултет Приложна математика и информатика при ТУ - София, съгласно процедурите и правилниците на университета, да архивира и съхранява тази дипломна работа с цел доказване във времето на моето авторство.

Дата: 14.06.2024 г.

Дипломант:

София

Литературни източници

[1] – Интернет на нещата – АЕНО:

<https://aeno.com/blog/bg/kakvo-e-internet-na-neschata/#:~:text=%D0%9A%D0%BE%D0%BD%D1%86%D0%B5%D0%BF%D1%86%D0%B8%D1%8F%D1%82%D0%B0%20%D0%B7%D0%B0%20%E2%80%9C%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%20%D0%BD%D0%B0%20%D0%BD%D0%B5%D1%89%D0%B0%D1%82%D0%B0,%D1%81%20%D0%BA%D0%BE%D0%BC%D0%BF%D1%8E%D1%82%D1%8A%D1%80%20%D0%B8%D0%BB%D0%B8%20%D0%B4%D1%80%D1%83%D0%B3%D0%B8%20%D1%85%D0%BE%D1%80%D0%B0.>

[2] – Internet of Things – Britannica:

<https://www.britannica.com/science/Internet-of-Things>

[3] – Internet of Things – IBM:

<https://www.ibm.com/topics/internet-of-things>

[4] – Best automatic cat feeders – TheSprucePETS:

<https://www.thesprucepets.com/best-automatic-cat-feeders-4175145>

[5] – Best automatic dog feeders – DogProductPicker:

<https://www.dogproductpicker.com/best-automatic-dog-feeders/>

[6] – Best automatic cat feeders – POPULAR SCIENCE:

<https://www.popsci.com/gear/best-automatic-cat-feeders/>

[7] – Blynk software reviews – Software Advice:

<https://www.softwareadvice.com/iot/blynk-profile/reviews/>

[8] – ThingsBoard software reviews – G2:

<https://www.g2.com/products/thingsboard/reviews#survey-response-4791313>

[9] – AWS IoT Analytics software reviews – G2:

<https://www.g2.com/products/aws-iot-analytics/reviews>

[10] – Advantages and disadvantages of C++ for embedded – EMBEDDED DEVELOPMENT TALK:

<https://www.qt.io/embedded-development-talk/c-for-embedded-advantages-disadvantages-and-myths>

[11] – Pros and cons of Python – Serokell ·

<https://serokell.io/blog/python-pros-and-cons>

[12] = Raspberry Pi = Wikipedia:

https://en.wikipedia.org/wiki/Raspberry_Pi

[13] – Pros and cons of Raspberry Pi – ROBU.IN

<https://robu.in/5-pros-and-5-cons-of-raspberry-pi/>

[14] – Arduino – Wikipedia

<https://en.wikipedia.org/wiki/Arduino>

[15] – Advantages and disadvantages of Arduino – Linuxhint

<https://linuxhint.com/advantages-and-disadvantages-arduino/>

[16] – DC motor – Wikipedia:

https://en.wikipedia.org/wiki/DC_motor

[17] – Brushless DC motors – Renesas:

<https://www.renesas.com/us/en/support/engineer-school/brushless-dc-motor-01-overview>

[18] – Stepper motor – Wikipedia:

https://en.wikipedia.org/wiki/Stepper_motor

[19] – Серво мотор – Wikipedia:

<https://bg.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%BE%D0%BC%D0%BE%D1%82%D0%BE%D1%80>

[20] – Using a servo motor with Raspberry Pi – Raspberry Pi Foundation:

<https://projects.raspberrypi.org/en/projects/grandpa-scarer/3>

[21] – Blynk events – Blynk.Documentation:

<https://docs.blynk.io/en/getting-started/events-tutorial>

[22] – Getting started with Raspberry Pi – Raspberry Pi:

<https://www.raspberrypi.com/documentation/computers/getting-started.html>

[23] – Python virtual environments – Python docs:

<https://docs.python.org/3/library/venv.html>

[24] – Python requirements file – pip documentation:

<https://pip.pypa.io/en/stable/reference/requirements-file-format/>

[25] – Python Blynk library – GitHub:

<https://github.com/vshymanskyy/blynk-library-python>

[26] – Controlling a servo motor with PWM – WOKWI:

<https://blog.wokwi.com/learn-servo-motor-using-wokwi-logic-analyzer/>

Използвани съкращения

IoT - Internet of Things - Интернет на нещата

AI - Artificial Intelligence - Изкуствен интелект

M2M - Machine-to-Machine - Машина-към-машина

IoMT - Internet of Medical Things - Интернет на медицинските неща

DC - Direct Current - Постоянен ток

BLDC - Brushless Direct Current - Безчетков постоянен ток

GPIO - General-Purpose Input/Output - Вход/Изход с общо предназначение

VNC - Virtual Network Computing - Виртуални мрежови изчисления

PWM - Pulse Width Modulation - Широчинно-импулсна модулация

Приложения

Пълният изходен код на проекта може да бъде намерен в GitHub:

<https://github.com/vikir01/AutomaticPetFeeder>