

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»**

**Факультет компьютерных наук
Основная образовательная программа
«Прикладная математика и информатика»**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Программный проект на тему
Программно-аппаратный комплекс для
обработки данных климатических датчиков**

**Выполнил студент группы 197, 4 курса,
Кайралап Досбол**

**Руководитель ВКР:
Профессор, Департамент бизнес-информатики / Департамент
прикладной математики, Попов Виктор Юрьевич**

Оглавление

Аннотация	3
Annotation	4
Ключевые термины	5
Постановка задачи.	6
Актуальность темы.	7
Обзор аналогов.....	10
Функциональные требования	12
Нефункциональные требования	13
Выбор технологий и инструментов разработки.	14
Архитектура системы	16
Описание интерфейса приложения.....	16
Описание работы программы (алгоритм обработки запроса)	17
Описание серверной части.....	18
Тестирование программы	19
Результаты и заключение	20
Библиографический список.....	20

Аннотация

Идея данного проекта заключается в создании программно-аппаратного комплекса для помощи в уходе за комнатным растением или за целой теплицей, посредством удаленного мониторинга климатических данных. Это программный проект, основная область которого - “умный дом” / сенсорные компоненты, его можно поделить на аппаратную (цепь с микроконтроллером и датчиками/сенсорами), серверную (которая преобразовывает все поступающие данные с датчиков) и клиентскую (приложение на Android, на которое выводятся все готовые данные) части. Аппаратную цепь необходимо разместить как можно ближе к объекту (например, внутри теплицы/горшка), она считывает климатические данные окружающей среды, а затем эти данные преобразуются на микроконтроллере (в скетче) и передаются на смартфон для вывода на экран. Приложение будет уведомлять пользователя о любых отклонениях от нормальных показателей и давать советы, чтобы пользователь мог вовремя отследить и устранить проблему, угрожающую урожаю или здоровью цветка. Данные также можно передавать проводным путём на монитор порта (в приложение Arduino IDE, на ноутбук). Также имеются тестовые модели: на платформе tinkercad и блочная сборка приложения.

Annotation

The idea behind this project is to create a software/hardware package to help with the care of an indoor plant or an entire greenhouse, by remotely monitoring climate data. It is a software project, the main area of which is smart home / sensor components, it can be divided into hardware (circuit with microcontroller and sensors), server (which converts all incoming data from sensors) and client (Android application on which all finished data is displayed) parts. The hardware circuit needs to be placed as close as possible to the object (e.g. inside a greenhouse/pot), it reads the environmental climate data, and then this data is converted on the microcontroller (in a sketch) and transmitted to the smartphone for display. The app will notify the user of any deviations from normal values and give advice so that the user can track down and rectify the problem threatening the crop or the health of the flower in time. Data can also be transmitted by wire to the port monitor (to the Arduino IDE application, on a laptop). There are also test models: on the tinkercad platform and a block build application.

Ключевые термины

1. Сенсор (датчик) - первичный преобразователь, элемент, воспринимающий контролируемое воздействие и преобразующий контролируемый параметр в удобный для использования сигнал (как правило электрический).
2. Микроконтроллер (микросхема) — микросхема, предназначенная для управления электронными устройствами. Имеет собственный процессор и память.
3. Ардуино (Arduino) [1, 2] – торговая марка аппаратно-программных средств построения и прототипирования для систем автоматики и робототехники. Это инструмент для проектирования электронных устройств (электронный конструктор), плотно взаимодействующих с окружающей физической средой. В данной работе использую микроконтроллер ArduinoUno [3].
4. Arduino IDE [4]— это программная среда разработки, использующая C++ и предназначенная для программирования всех плат ряда Ардуино (Arduino).
5. Скетч (sketch) - обозначение кода (программы) для Arduino. Это единичный проект, который загружается и выполняется платой. В данной работе я регулярно буду пользоваться именно этим термином.

Постановка задачи.

Идея проекта заключается в создании программно-аппаратного комплекса для ухода за комнатным растением или за целой теплицей. Это программный проект, основная область которого - “умный” дом и сенсорные компоненты, его можно поделить на аппаратную (цепь с микроконтроллером и датчики/сенсоры), серверную (которая преобразовывает все поступающие данные с датчиков) и клиентскую (приложение на Android, на которое выводятся все готовые данные) части. Аппаратную цепь необходимо разместить как можно ближе к объекту (например, внутри теплицы/горшка), она считывает климатические данные окружающей среды, а затем эти данные преобразуются на микроплате (в скетче) и передаются на смартфон для вывода на экран.

Актуальность темы.

Неурожай является серьезной проблемой, которая влияет на производительность сельского хозяйства во всем мире. Он может произойти по целому ряду причин, например, изменение погодных условий в результате изменения климата может вызвать сильные засухи, наводнения и непредсказуемые температурные сдвиги. Такая непредсказуемость затрудняет для фермеров планирование посевных и уборочных работ, что часто приводит к неурожаю. Для смягчения этих проблем использование теплиц может быть весьма полезным. Вот почему дешевые и удобные теплицы являются необходимостью:

- 1) Контролируемая среда: Теплицы обеспечивают контролируемую среду, в которой можно эффективно управлять температурой, влажностью и освещением. Это позволяет фермерам выращивать урожай круглый год, независимо от внешних погодных условий.
- 2) Борьба с вредителями и болезнями: В теплице легче бороться с вредителями и болезнями, поскольку среда является закрытой. Такая изоляция помогает предотвратить распространение вредителей и болезней из внешних источников.
- 3) Эффективность использования воды и почвы: Теплицы могут способствовать развитию устойчивых методов ведения сельского хозяйства, таких как гидропоника или аквапоника, которые используют меньше воды и не разрушают почву. Это особенно важно в районах с дефицитом воды.
- 4) Повышение урожайности: Благодаря контролируемой среде и возможности круглогодичного выращивания, теплицы могут значительно повысить урожайность с одного акра по сравнению с традиционным земледелием.
- 5) Доступность и дешевизна: Разработка и распространение дешевых, удобных теплиц может сделать преимущества тепличного хозяйства доступными для мелких фермеров и фермеров в развивающихся странах.

Это может значительно повысить их урожайность и экономическую стабильность.

- б) Разнообразие растений: В теплице фермеры могут выращивать более широкий спектр культур, включая те, которые обычно не подходят для климата их региона. Такое разнообразие может помочь обеспечить продовольственную безопасность и предоставить фермерам дополнительные возможности для заработка.

Более широкое использование доступных и удобных теплиц может существенно помочь смягчить многие проблемы сельского хозяйства, а для этого нужно бесперебойно и максимально точно отслеживать климатические данные. Для этих целей и появился мой проект “Программно-аппаратный комплекс для обработки данных климатических датчиков”, с помощью которого мы сможем легко следить за показаниями с датчиков и лучше ухаживать за огородом, теплицей или даже комнатным растением. Приложение будет уведомлять вас о любых отклонениях от нормальных показателей и давать советы, чтобы вы могли вовремя отследить и устранить проблему, угрожающую вашему урожаю или здоровью вашего цветка.

Для выполнения проекта необходимо было выполнить три комплексные задачи:

- ★ Сборка аппаратной части (цепи с Arduino Uno, датчиками и модулями WiFi и Bluetooth);
- ★ Написание Android приложения, для мониторинга данных и уведомлений(советов);
- ★ Реализация скетч-кода для работы микроконтроллера (обработки и передачи данных на сервер/приложение).

Я сформулировал следующий список подзадач для выполнения основных задач:

1. исследование и анализ уже существующих решений в данной области;
2. знакомство с Arduino IDE и выбор технологий для разработки программного обеспечения;
3. создание простой модели цепи с использованием онлайн-платформы Tinkercad [5];
4. приобретение необходимых компонентов, включая плату ArduinoUno, различные сенсоры и модули для беспроводной связи, а также сборка аппаратной части;
5. подготовка скетч-кода для микроконтроллера и загрузка его на плату;
6. создание приложения на Android, удовлетворяющего функциональным и нефункциональным требованиям;
7. реализация сервера для хранения данных с датчиков;
8. Обеспечение передачи данных с платы на смартфон по Bluetooth и WiFi (через сервер);
9. Тестирование каждого элемента проекта и корректировка программы/кода/сборки при необходимости;
10. Подготовка итогового документа и всей необходимой документации.

Обзор аналогов

❖ Koubachi WiFi cloud sensor [6]

Сенсоры Koubachi, способные отслеживать параметры жизнеобеспечения каждого растения индивидуально, в помещении и на открытом воздухе. В системе используется датчик растений, работающий по Wi-Fi, который, будучи помещенным в горшок или землю рядом с растением, отслеживает различные факторы окружающей среды, такие как температура, интенсивность света и влажность почвы. Собранные датчиком данные отправляются в облако Koubachi, где они анализируются и используются для предоставления пользователю конкретных инструкций по уходу через приложение Koubachi.

Датчик Koubachi использует научные модели ухода и может настроить план по уходу за разными растениями. Мониторинг в режиме реального времени. Простота использования, удалённый доступ по Wi-Fi, именно этими характеристиками я вдохновлялся в своей работе.

Но есть минусы: довольно дорогостоящая система, совсем не подходящая для больших грядок. Лишь на одно растение нужно целое устройство, что в сумме с дороговизной ставит крест на использовании данной системы в теплице. Зато идеально подойдет для комнатных растений в горшках.

❖ Life Control [7]

Умный датчик ухода за растениями от LifeControl. Имеет те же плюсы, что и предыдущий пример, но считывает меньше показателей. Компактный и удобный для посадки, управляется через мобильное приложение. Данный датчик довольно удобен. Но не имеет хранилища данных и сильно зависит от постоянной бесперебойной работы Wi-Fi, а это может ограничить полезность системы в районах с ненадежным интернет-соединением. Поэтому я решил добавить в свой проект не только Wi-Fi, но и в первую очередь Bluetooth подключение. При хорошей антенне можно будет не переживать за

работоспособность такой системы даже вдали от города, а там, как правило, земля под теплицы гораздо дешевле.

Приведенные выше примеры больше подходят скорее для комнатных растений в горшках или небольших огородов. Поэтому приведу пример полноценной умной теплицы:

❖ **SmartGreenhouse от SolarImpulse [8]**

Умная теплица площадью целых 22 квадратных метра выращивает безопасные фрукты и овощи в ограниченном пространстве с минимальными затратами на обслуживание и площадь. Теплица включает в себя датчики уровня воды и климатических условий, а также оснащена самоуправляемой вертикальной системой аквапоники. Она отправляет данные непосредственно в облако с помощью ИОТ-соединения и обрабатывает собранные данные с помощью удаленной машины искусственного интеллекта, чтобы направлять пользователю умные уведомления (указывать необходимые корректирующие действия). Теплица продаётся только во Франции и стоит огромных денег.

Эти примеры имеют большой функционал и закрытый код. В своём проекте я попробовал взять лучшие идеи от данных систем и совместить их (кроме систем авто-полива/аквапоники), но сделал проект сильно дешевле аналогов и способный работать по Bluetooth. Моя система будет не только отображать климатические показатели окружающей среды в реальном времени, но и давать советы и оповещения, а главное временную статистику, чего не хватает аналогам.

Функциональные требования

Организация входных и выходных данных:

Данные должны считываться датчиками, преобразовываться на микроплате и отправляться в приложение для вывода на экран телефона. Есть три варианта передачи данных: 1) через bluetooth модуль непосредственно на телефон; 2) через специальный USB провод, подсоединив arduino к ноутбуку, тогда данные передаются в приложение Arduino IDE[4] и их можно наблюдать на мониторе порта; 3) данные передаются по WiFi на сервер, затем приложение берёт их (простыми запросами) и выводит на экран устройства.

Необходимое оборудование:

У пользователя обязательно должна быть собранная, полностью рабочая аппаратная часть с загруженным на микроплату скетч-кодом, очевидно, что без неё приложение не сможет получать необходимые данные.

Интерфейс приложения:

Необходим удобный, плоский дизайн, понятный любому человеку. Необходима возможность легко рассматривать данные с любого датчика и за любой доступный период.

Временные характеристики:

Приложение должно быстро загружаться и работать плавно, без задержек и сбоев. Необходима достаточная скорость обработки и передачи данных на смартфоне.

Аппаратная часть должна иметь возможность работать достаточно долгое время без подзарядки.

Нефункциональные требования

Производительность и надежность:

Приложение должно всегда работать правильно, с минимальным временем простоя или ошибками. Минимальная необходимая производительность устройства пользователя должна быть как можно меньше.

Удобство использования:

Приложение должно иметь интуитивно понятный, дружелюбный интерфейс. В нем должны легко ориентироваться пользователи, не обладающие техническими знаниями.

Безопасность:

Приложение не будет хранить никакие важные пользовательские данные/информацию. Оно будет хранить лишь данные с датчиков, поэтому нет необходимости к надёжности хранения данных. Также данные будут храниться на серверной части.

Выбор технологий и инструментов разработки.

Для проекта я использовал самую удобную версию (по личному мнению) микроплаты Arduino - Arduino Uno. Для неё написано огромное количество удобных библиотек, которыми я активно пользовался в ходе выполнения задания. А также у этой модели хорошая вычислительная мощность и скорость передачи данных за небольшую плату.

Для создания и отладки кода я использовал официальное программное обеспечение Arduino IDE 2.1.0 (новейшая версия, на момент написания диплома). А также эту платформу можно использовать для передачи (загрузки) скетч-кода непосредственно на плату и наблюдать данные с датчиков с помощью монитора порта, чем я пользовался во время отладки кода и сборки.

Я уже использовал это программное обеспечение на майноре "Умный дом", лекциями которого я пользовался при подготовке аппаратной части. Там же я и изучил сенсорные компоненты, язык программирования Arduino и микроконтроллер, а также на этом курсе я научился пользоваться tinkercad-ом и строить там тестовые модели сборок.

В проекте были использованы следующие 4 датчика:

- BH1750 - освещенности (цифровой люксметр);
- 080МН - влажности почвы;
- DHT11 - влажности воздуха, также измеряет температуру;
- GY-BME_P_280 - температуры, давления и влажности воздуха;

А также два модуля:

- BK32-31(китайский аналог модуля HC05) - модуль передачи данных bluetooth;
- ESP_32 – плата разработки на базе ESP32-WROOM-32D, WiFi + Bluetooth модуль

Также я подключил потенциометр, который в будущем планирую использовать для некой регулировки работы датчиков (например,

чувствительность).

В аппаратном сегменте использовал соединительные провода типа "папа-папа" и "мама-папа", а также резисторы с разным сопротивлением (нужны были для подключения некоторых датчиков). Для удобства и подключения использовал большую плату на восемьсот тридцать входов. Два датчика необходимо было паять с помощью специальных соединительных штекеров. Всё получилось не с первого раза, но в целом обошлось без казусов. При создании первой версии приложения использовал MITAppInventor [9] - очень удобный сервис с простой поблочной сборкой.

Архитектура системы

Данная система обработки данных для климатических датчиков, как следует из ее названия, состоит из аппаратных и программных элементов. Программный компонент состоит из клиентского сегмента и серверного сегмента, которые последовательно взаимодействуют между собой. Клиентский сегмент представлен приложением, которое используется пользователями для доступа к данным датчиков путем выбора конкретного датчика из списка и последующего получения необходимой информации. Серверный сегмент отвечает за выполнение запрошенных вычислений и возвращает данные клиентскому сегменту. Серверный сегмент включает в себя микроплату ArduinoUno, загруженный на неё скетч-код и сервер, куда периодически передаются все данные датчиков после обработки.

Описание интерфейса приложения

Пользовательский интерфейс прост и удобен, состоит всего из нескольких строк. Первая строка представляет собой название приложения, набранное крупным шрифтом - фиксированный элемент. Далее идет строка состояния подключения, набранная более мелким шрифтом, а за ней - тумблер. Если он выключен, приложение перестанет отправлять или получать запросы на плату Arduino или от нее. После этого находится кнопка выбора датчика, и, наконец, сами данные, которые выводятся на экран.

Описание работы программы (алгоритм обработки запроса)

Работа приложения начинается с того, что пользователь запускает приложение, соединяет свое мобильное устройство с ArduinoUno через bluetooth/wifi, а затем он может выбирать датчик на главном экране, показания с которого он хочет проверить (и также может проверить статистику). Каждый из сенсоров имеет порядковый номер. Затем на микроплату отправляется запрос (номер датчика, показатели с которого хотим узнать, передается со смартфона либо на саму плату, либо на сервер, в зависимости от типа подключения). На микроплате(серверной части) в это время работает постоянный цикл void, который получает запрос и выполняет необходимый блок скетч-кода. Для каждого датчика прописана своя функция получения данных, преобразования этих данных в привычный вид и вывод в приложение или на сервер. Пользователь получает эти данные на экране своего смартфона. При необходимости приложение может предложить совет (и уведомление), например, “полейте растение”, если уровень влажности почвы ниже нормы и т.д.

Описание серверной части

Серверная часть представлена в виде скетча на микроконтроллере, написанного в ардуино IDE, и сервера с хранящимися данными. Я разделил код на удобные для чтения блоки (по блоку на каждый сенсор) и прокомментировал каждый блок в скетче. Почти все датчики имеют свои уникальные библиотеки, которые упрощают их использование и вывод данных. Я использовал некоторые пины микроплаты в виде отдельных переменных, чтобы упростить использование и модификацию кода, ведь так отсутствует необходимость менять пины по всему коду при переключении проводов. Я расписал цикличную часть скетча в одной удобной функции `void loop`, а инициирование датчиков и пинов поместил в объемную функцию `void setup`

Тестирование программы

Перед сборкой аппаратной цепи построил простую модель сборки и расписал отдельный скетч на виртуальной платформе tinkercad [10], там же и тестировал начальную версию скетч кода. Для тестирования аппаратной сборки подключил микроконтроллер к ноутбуку и выводил все показатели датчиков на монитор порта в Arduino IDE. В самом скетче установил `delay(2000)`, то есть интервал передачи данных 2 секунды. Затем я активно взаимодействовал с окружением аппаратной части: орошал растение, светил на люксметр с разного расстояния, нагревал датчик температуры сжимая в руке или наоборот охлаждал на балконе и тому подобное. Всё это время я наблюдал как меняются данные передаваемые с Arduino Uno и отлаживал систему. Для начальной версии приложения (клиентской части) выбрал платформу MIT App Inventor 2, на которой поблочно собрал код и интерфейс будущего приложения, скачал и тестировал непосредственно работу приложения и передачу данных между серверной и клиентской составляющими.

Полная версия кода и apk итогового приложения будут добавлены в закрытый репозиторий [11].

Результаты и заключение

Был хорошо продуманный план, разделённый на 10 подзадач, благодаря которому мне удалось достичь своих целей и получить рабочую версию проекта в срок. Данные запрашиваются, считываются, передаются и выводятся корректно, приложение имеет понятный интерфейс и радует глаз, аппаратная часть довольно некомпактная, но зато задействует почти всю плату. Реализовано хранилище данных и отображение статистики, добавлен модуль wifi + bluetooth передачи и написан соответствующий скетч-код, зацикленный на микроконтроллере Arduino Uno. В итоге я доволен результатами проделанной работы, все блоки проекта протестированы и отредактированы. Все части проекта (аппаратная, серверная и клиентская) а также все цели выполнены в полной мере.

В будущем планирую приобрести больше более совершенных датчиков, усложнить систему и подключить их к более мощной микроплате. Собрать всю аппаратную сборку в удобную и защищенную оболочку и тогда получится полностью реализованный продукт, готовый к распространению.

Хочется отметить, что несмотря на все трудности, работать над проектом было очень полезно и познавательно. Я рад, что смог применить знания, полученные за все 4 года обучения на практике, написал приложение на android, поработал с микроконтроллером и датчиками. В особенности хочу поблагодарить Сорокина Игоря Михайловича за его курс “Умный дом” (майнор за 2-3 курсы) на котором я получил важные знания о сенсорных системах, языке Arduino, среде разработки Arduino IDE и платформе tinkercad, которыми я воспользовался в данном проекте. Именно этим курсом я вдохновлялся при выборе данной инициативной темы на выпускную квалификационную работу. Я впервые работал с сервером и реализовал полноценное приложение на Android, получил массу полезных знаний об умных теплицах и передаче данных.

Библиографический список

1. Arduino.cc [Электронный ресурс] - официальный сайт Arduino.
Режим доступа: <https://www.arduino.cc/>, свободный. Дата обращения - 20 января 2023 г.
2. Статья «Arduino» [Электронный ресурс].
Режим доступа: <https://ru.wikipedia.org/wiki/Arduino>, свободный.
Дата обращения - 2 февраля 2023 г.
3. ArduinoUno [Электронный ресурс] / раздел «Контроллеры Arduino» / arduino.ru. Дата публикации - 7 окт. 2010 г.
Режим доступа: <http://arduino.ru/Hardware/ArduinoBoardUno>, свободный.
4. Arduino IDE [Электронный ресурс] / blog-programmista.ru. Режим доступа: <https://blog-programmista.ru/post/80-arduino-ide-cto-eto-za-programma.html>, свободный.
5. Autodesk Tinkercad [Электронный ресурс] - онлайн платформа для создания моделей.
Режим доступа: <https://www.tinkercad.com/>, свободный.
Дата обращения - 2 фев. 2023 г.
6. Статья «Уникальная система по уходу за растениями от Koubachi» [Электронный ресурс] / medgadgets.ru. Дата публикации - 25 окт. 2013 г.
Режим доступа: <https://medgadgets.ru/main/unikalnaya-sistema-po-uxodu-za-rasteniyami-ot-koubachi.html>, свободный.
7. Товар «Датчик ухода за растением», [Электронный ресурс] / раздел «Устройства» / lifecontrol.ru
Режим доступа: <https://lifecontrol.ru/devices/plant-sensor/>, свободный.
Дата обращения - 7 фев. 2023 г.
8. Solar Impulse [Электронный ресурс] / раздел smart-greenhouse - небольшая Французская умная теплица; Дата обращения - 15 янв. 2023 г.
Режим доступа: <https://solarimpulse.com/solutions-explorer/smart-greenhouse>, свободный.

9. Удобная среда создания приложения по блокам
Режим доступа: <http://ai2.appinventor.mit.edu/>, свободный.
Дата обращения - 2 марта 2023 г
10. Первоначальный макет проекта на платформе тинкеркад
Режим доступа: <https://www.tinkercad.com/things/keTjWcIiHep-proekt-umnoe-rastenie/editel?sharecode=-d3h8xe8rZP6BH4eVJaMWBohimyY6Sg2SdcRxBizSfM>, закрытый,
доступен по ссылке.
11. Github репозиторий (закрытый) с исходным кодом (приложение, сервер, скетч) доступен по ссылке: <https://github.com/DosBall/Software-and-Hardware-Complex-for-Processing-Climate-Sensor-Data>