

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКІЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА програмування ТА МАТЕМАТИКИ

Пояснювальна записка

до дипломної роботи

бакалавр

(Освітньо-кваліфікаційний рівень)

На тему: «Комп'ютерна система автоматизації теплиці»

Виконав: студент 4 курсу, групи СКС-15зн

Напряму підготовки: 6.050102 «Комп'ютерна
інженерія»

_____ Горьковець С.С.
(Підпис)

Керівник,

доцент, к.т.н., _____ Захожай О.І.
(Підпис)

Рецензент,

доцент, к.т.н., _____ Іванов В.Г.
(Підпис)

СЄВЕРОДОНЕБК
2019 року

СХІДНОУКРАЇНСЬКІЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет інформаційних технологій та електроніки

Кафедра програмування та математики

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 6.050102 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМ,

д.т.н., доцент

_____ Лифар В.О.

«___» _____ 2019 р.

завдання

НА дипломних роботах СТУДЕНТА

Горьковець Сергій Сергійович

1. Тема роботи: «Комп'ютерна система автоматизації теплиці»

Керівник роботи Захожай Олег Ігорович

ЗАТВЕРДЖЕНІ наказом ВИЩОГО навчального закладу від «___» ___ 2019 року № __

Строк подання студентом роботи 07 червня 2019 р.

Вихідні дані (технічні вимоги): створити програмно

апаратний модуль для автоматизації макета теплиці и моніторингу показників датчиків в режимі online.

Зміст ВКР: вибір апаратної бази для створення «розумної» теплиці, створення мікроклімату в теплиці, організація єдиної системи

автоматизації з веб-інтерфейсом с допомогою графічного конфігуратора Node-RED.

Перелік звітних матеріалів: Пояснювальна записка, ілюстративний матеріал.

Дата видачі завдання: 22 березня 2019 року.

КАЛЕНДАРНИЙ ПЛАН

№ з / п	Назва етапів дипломної роботи	рядків Виконання етапів роботи	Примітка
1	Огляд літератури за темою роботи	28.03.2019	
2	Вибір апаратної бази для створення «розумної» теплиці	4.04.2019	
3	Створення мікроклімату в теплиці	22.04.2019	
4	Організація єдиної системи автоматизації з веб- інтерфейсом за допомогою графічного конфігуратора Node-RED	11.05.2019	
5	Оформлення ілюстративного матеріалу	22.05.2019	
6	Оформлення пояснювальної записки	07.06.2019	

студент

(Підпис)

Керівник роботи

(Підпис)

ЛИСТ ПОГОДЖЕННЯ І оцінювання

Дипломної роботи студентки гр. СКС-15зн Горьковець С.С.

Керівник роботи, доцент, к.т.н. . _____ Захожай О.І.

Оцінка керівника: _____

рецензент _____
(ПБ, місто роботи, посада)

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

Голова ЕК _____ Меняйленко О.С.

РЕФЕРАТ

Пояснювальна записка містить: 76 с., 42 рис., 3 табл., 2 додатки, 20 джерел літератури.

Однопалатний КОМП'ЮТЕР, ВЕБ-Додаток, АВТОМАТИЗАЦІЯ, ДАТЧИКИ, RASPBERRY PI, NODE-RED, WEAVED, ARDUINO, ТЕПЛИЦЯ.

Мета роботи: аналіз і вибір апаратної частини з подальшою розробкою ПЗ для автоматизації макета теплиці і віддаленого моніторингу показників датчиків в режимі реального часу.

У випускній кваліфікаційній роботі наводиться огляд застосування IoT-концепції в різних галузях, з якого можна зробити висновок, що сільське господарство є однією з найбільш популярних галузей для впровадження нових рішень автоматизації управління в рамках цієї концепції.

У доповненні до цього були проведені порівняльний аналіз апаратних засобів, таких як мікроконтролери, Arduino, Raspberry Pi, для автоматизації макета теплиці.

Також здійснено огляд основних датчиків, які використовувалися для відстеження необхідних характеристик, таких як вологість ґрунту, освітленість і т.д., і можливих алгоритмів створення мікроклімату.

На основі вибраних апаратно-програмних засобів була створена система автоматизації на основі однопалатного комп'ютера Raspberry Pi, а також створено веб-додаток з призначенням для користувача інтерфейсом.

Ідеї, компоненти і програми можна перенести в реальні сільськогосподарські і приватні проекти.

Зміст

ВСТУП	8
1 Вибір апаратної бази для створення «розумної» теплиці	11
1.1 «Розумна» теплиця	11
1.2 Огляд технологій для управління автоматизацією	14
1.2.1 мікроконтролери	14
1.2.2 Arduino	16
1.2.3 Одноплатні комп'ютери на прикладі Raspberry Pi	23
2 Створення мікроклімату в теплиці	28
2.1 Моніторинг показників в теплиці	29
2.1.5 Аналогово-цифровий перетворювач ADS1115	36
2.1.6 MQ-2 Gas Sensor	38
2.1.7 Moisture sensor	39
2.1.8 Raspberry Pi Camera Board	40
2.2 Автоматизація створення мікроклімату	41
3. Організація єдиної системи автоматизації з веб- інтерфейсом за допомогою графічного конфігуратора Node-RED	43
3.1 Інтеграція периферійних пристроїв з Node-RED	45
3.2 Створення алгоритмів автоматизації теплиці	50
3.3 Розробка користувальницького інтерфейсу	53
3.4 Віддалений доступ до Raspberry Pi за допомогою Weaved	55
3.5 Опис схеми і алгоритму роботи	58
4. Матеріальні витрати	60
ВИСНОВОК	62
Список використаних джерел	64
Додаток А	66
Додаток Б	72

ВИЗНАЧЕННЯ, ПОЗНАЧЕННЯ І СКОРОЧЕННЯ

У цій пояснювальній записці застосовують такі терміни з відповідними визначеннями:

Bluetooth- виробнича специфікація бездротових персональних мереж (Wireless personal area network, WPAN).

GPIO - General-Purpose Input / Output, інтерфейс для зв'язку між компонентами комп'ютерної системи, наприклад мікропроцесором і різними периферійними пристроями.

IC - Integrated Circuit, інтегральна схема

IoT- Internet of Things, Інтернет речей

JSON - JavaScript Object Notation, текстовий формат обміну даними, заснований на JavaScript

NOR - один з основних типів флеш-пам'яті

OTP - One Time Programmable, одноразово програмовані

ROM - Read Only Memory, пам'ять тільки для читання

PCB - Printed Circuit Board, друкована плата

SD-карта - Secure Digital Memory Card, це довготривалий формат карт пам'яті

SoC- System-on-a-Chip, електронна схема, що виконує функції цілого пристрою (наприклад, комп'ютера) і розміщена на одній інтегральній схемі

АЦП - Аналогово-Цифровий Перетворювач

ОЗУ - Оперативне запам'ятовуючий пристрій

ПО - Програмне забезпечення

ВСТУП

Ми стоїмо на порозі технічної революції, яка докорінно змінить наше ставлення до життя, роботи і відношенню один до одного. Ми поки не знаємо, як вона буде розвиватися, але ясно одне: відповідь має бути комплексним і всеосяжним, в якому беруть участь всі зацікавлені сторони: від державного і приватного секторів до академічних кіл та громадянського суспільства.

Перша промислова революція використовувала енергію води і пара для механізації виробництва. Друга використовувала електроенергію для створення масового виробництва. Третя використовувала електроніку та інформаційні технології для автоматизації виробництва. На ній будується четверта промислова революція (індустрія 4.0). Вона характеризується злиттям технологій, які розмивають кордони між фізичною, цифровою і біологічною сферами.

Четверта промислова революція міняє саму концепцію виробництва. В рамках інтелектуального виробництва автоматизація, яка визначається програмним забезпеченням, дозволяє виробникам пов'язувати всі етапи ланцюжка створення вартості, швидко адаптуватися до мінливих ринків і створювати персоналізовані продукти в масовому масштабі. Ці машини повідомляють про свій стан в режимі реального часу, підвищуючи ефективність і пропускну здатність і зводячи до мінімуму час простою.

В основі четвертої промислової революції лежить Інтернет речей (IoT). Коротко цю концепцію можна описати так: збільшення кількості пристроїв, які з'єднано не тільки з користувачами, але й один з одним. «IoT» в застосуванні до автоматизації дозволяє створювати системні архітектури, які виявляються більш ощадливими, гнучкими і ефективними. Мета - безшовні комунікації і взаємодію з польовим введенням / висновком, включаючи

датчики, виконавчі пристрої,

аналізатори, приводи, системи машинного зору, відео, робототехніку, заради підвищення продуктивності і гнучкості виробництва.

Потенційні можливості застосування IoT численні і різноманітні. Вони проникають практично в усі сфери повсякденного життя людей, підприємств і суспільства в цілому. Це такі сфери, як:

- *Охорона здоров'я.* IoT-платформи забезпечують централізований моніторинг і агрегацію даних медичного обладнання та додатків.

- *Транспорт.* Процеси транспортування, дистрибуції та логістики потребують постійного контролю і аналітиці. IoT-платформа може в реальному часі інформувати про мінливі умови, допомагаючи при цьому приймати швидкі і обґрунтовані рішення.

- *Торгівля.* Інтернет речей робить можливим впровадження високих технологій на підприємствах роздрібної та оптової торгівлі, пропонуючи рішення по автоматизації виробництва.

- *Сільське господарство.* IoT-платформа дозволяє автоматизувати безліч аспектів діяльності сільгоспвиробництва, підвищуючи ефективність і фінансові показники на виробництві.

Сфери застосування інтернет речей не обмежуються вказаними, в кожній можна знайти найбільш ефективний сценарій використання.

Оцінок обсягу ринку IoT-технологій, що використовуються в сільськогосподарській галузі, не існує. Однак, є дані для деяких компонентів, які вже зараз задіяні фермерами. Наприклад, обсяг світового ринку безпілотників для виконання аграрних завдань до 2022 року складе \$ 3,69 млрд, повідомляють аналітики Research & Markets. Дрони показали свою ефективність в картографуванні, посівних роботах, дистанційного зондування і обприскуванні. У звіті PwC [1] вказується, що прогнозна вартість глобальних рішень з використанням безпілотних пристроїв в аграрній галузі становить

\$ 32,4 млрд. Таким чином, можна зробити висновок про можливість розвитку даної галузі і впровадження проектів з IoT-концепцією.

Аналітиками виділяються чотири основні причини, які схиляють сільськогосподарські підприємства використовувати технології Інтернету речей.

- *Оптимізація операційних витрат*

Інтернет речей дозволяє перейти до точного землеробства. Технології точного землеробства полягають у виявленні неоднорідностей окремо взятого земельної ділянки. Для цього використовуються різні бази геоданих, які підкажуть, в якій частині поля не вистачає певних мінеральних речовин, де шкідників більше, а де - менше і т.д. Фермер, використовуючи ці знання, вносить певну кількість коштів хімічного захисту рослин і мінеральних добрив. У процесі спостереження за земельною ділянкою за допомогою баз геоданих можна прогнозувати час дозрівання і обсяги врожаю.

- *Боротьба зі шкідниками*

Сучасні системи використовують датчики, які виявляють кількість шкідників на певній ділянці поля і здатні автоматично розпорошувати кошти хімічного захисту рослин тільки там, де це необхідно. Останнім часом також з'явилися дрони-вбивці, які здатні вести полювання на шкідників навіть в нічний час.

- *економія води*

Спеціальні сенсори, які встановлені на земельній ділянці, в певний час протягом доби проводять оцінку аналізу вологості ґрунту. Ґрунтуючись на цих даних, сенсори регулюють полив. Це дозволяє істотно скоротити витрату води особливо в посушливих територіях. Крім того, фермерам не треба оплачувати полив в

ручному режимі: тепер вони можуть зосередитися на інших справах.

- *зберігання врожаю*

Сучасне обладнання для складів з урожаєм дозволяє проводити виміри

вологості і температури в приміщенні.

Автоматичне регулювання цих параметрів призводить до того, що фрукти, овочі, ягоди і зернові не псуються тривалий час.

- *Автоматизація сільськогосподарських завдань за допомогою роботів*

Наприклад, компанія Harvest [2] розробила новий підхід до автоматизації цілого ряду традиційно ручних трудових завдань, заснованих на технології мобільних роботів. Команди невеликих, високоінтелектуальних машин безпечно працюють з робітниками, щоб виконувати найбільш фізично складні завдання і значно знизити витрати.

Метою даної дипломної роботи був розгляд застосування IoT-концепції в сільському господарстві, як однієї з найбільш популярних сфер її застосування, на прикладі створення системи автоматизації та віддаленого моніторингу показників датчиків на макеті теплиці в режимі реального часу.

Розроблений апаратно-програмний продукт можна використовувати як базис для сільськогосподарських і приватних рішень, а також заходи щодо навчання та демонстрації можливостей IoT.

1 Вибір апаратної бази для створення «розумної» теплиці

Перед створенням власної автоматизованої системи для управління теплицею необхідно ознайомитися з вимогами, які можуть до неї пред'являтися, а також розглянути варіанти апаратної бази, на основі якої може бути створена така система.

1.1 «Розумна» теплиця

Основним призначенням автоматизації теплиці є моніторинг різних показників датчиків і управління оптимальними умовами для росту рослин, які вимагають певного діапазону температури, вологості ґрунту, світла, вологості і складу повітря (кисень, двоокис вуглецю і азот), а також поживних речовин для росту. Тепличне покриття допомагає контролювати багато з цих факторів, щоб допомогти в розвитку рослин і збільшенні

кількості і якості врожаю. Але які саме чинники контролюються в теплиці?

«Розумна» теплиця [3] може контролювати:

- *тепло* - спроба запобігання замерзання або перегрівання - для томатів та інших теплолюбних овочів, ніякого зростання не відбувається нижче 10 ° С або вище 30 ° С, може погіршитися колір плода, якщо температура підніметься вище 29 ° С.

- *воду*- тому що немає прямих опадів на ґрунті в теплиці, то повинно бути забезпечено зрошення, потрібно управляти об'ємом води і часом поливу для оптимізації росту.

- *світло* - будь-яке покриття буде блокувати деякий відсоток світла (або певних довжин хвиль світла), доступних для рослин - затінення може бути корисним, коли занадто жарко.

Витрата повітря і вологість - якщо теплиця буде повністю закрита, вологість підвищиться, і оскільки здорові ростуть рослини є «чистими вуглецевими фіксаторами», рівень вуглекислого газу зменшиться, а кисню збільшиться. І навпаки, вночі, коли рослини використовують більше кисню і виділяють вуглекислий газ, оскільки вони

«Переварюють» їжу, яку вони зробили з фотосинтезу. Азот буде також зменшуватися для азотфіксуючих рослин, тому зменшення потоку повітря може бути шкідливим для росту рослин. Цвітіння може з'являтися на деяких рослинах при вкрай низькій або високій вологості.

- *шкідників* - теплиця може обмежити доступ деяких потенційних шкідників, але також може замкнути їх усередині, і фактично забезпечить відмінну середовище проживання для них.

- *запилення* - деякі рослини потребують комах для запилення, а деяким необхідний вітер. Навіть самоzapильних рослини можуть отримати користь від вітру або комах, або від ультразвукової обробки (вібрації) бджолами.

Більшість людей хочуть використовувати теплиці з метою контролю

тепла для певних рослин. Але одночасно ми може захотіти контролювати воду, світло, потоки повітря, шкідників і запилення одночасно. Факт в тому, що ми можемо спробувати захистити наші рослини від холоду, але, якщо ми не звернемо уваги, ми можемо також дозволити рослинам перегрітися в теплиці. Таким чином, ми або приділяємо дуже пильну увагу і коригуємо все вручну, або створюємо автоматично керовану систему для управління тепличної середовищем.

Види автоматизованих систем, які допомагають контролювати мікроклімат в теплиці:

1. *зрошення* - контроль надходження води за регулярним графіком або, коли рівень вологості ґрунту падає нижче заданого рівня.

2. Вентиляція - термостат може автоматично включати і вимикати вентилятори, або відкривати або закривати жалюзі.

3. Затінення - тінистий тканину з термостатичним контролем автоматично затінює рослини.

4. Охолодження - термостатичні система кондиціонування повітря.

5. Опалення - термостатичні система опалення - зазвичай пропан або природний газ.

6. Обприскування - автоматичне підвищення вологості, або в поєднанні з вентиляторами для охолодження шляхом випаровування.

7. Дозування поживних речовин - поживні речовини можуть автоматично розподілятися по системі зрошення - складні системи включають комп'ютерний аналіз ґрунту.

8. Боротьба зі шкідниками - програмовані системи, які автоматично розпорошують перетини так само, як дезодоранти в громадських туалетах.

9. Подвійні шари і система інфляції - дуже хороша система ізоляції - включають в себе подвійний шар пластикової плівки, яка підтримується тиском повітря від вентиляторів.

Які типи автоматизованих технологій ми включимо для контролю за теплицею, будуть в значній мірі бути функцією вартості. Також важливий

рівень контролю - наприклад, низький рівень автоматизації означає, що ми повинні виконувати всю роботу самі. Якщо стає занадто спекотна, потрібно відкрити або закрити теплицю, а також потрібно пам'ятати, що її потрібно закрити назад ввечері, перш ніж все тепло буде втрачено. Якщо цього не зробити, рослини і врожай постраждають.

1.2 Огляд технологій для управління автоматизацією

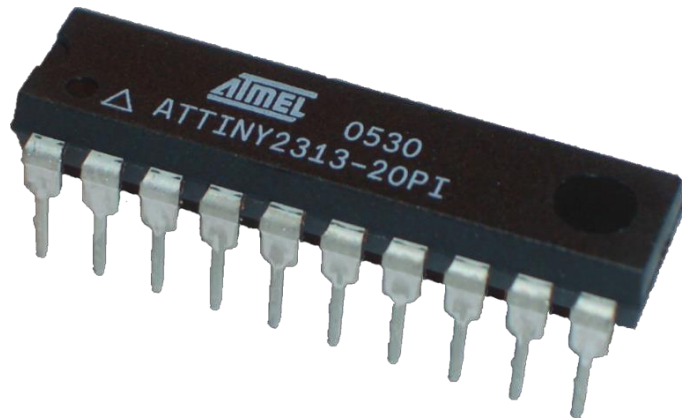
Після того, як ми вирішимо, за якими показниками ми хочемо стежити, а також ніж керувати, необхідно вибрати апаратну базу для реалізації нашої системи. Для цього розглянемо основні способи і технології, розберемо їх гідності, недоліки.

1.2.1 мікроконтролери

Мікроконтролер [4] - це спеціальна мікросхема, призначена для управління різними електронними пристроями. Це вже не процесор, але ще й не комп'ютер. У сучасній термінології це система на чіпі (SoC). Мікроконтролер містить один або кілька процесорних ядер, а також пам'ять і програмовані периферійні пристрої введення / виводу. Програмна пам'ять у вигляді сегнетоелектричного ОЗУ, флеш-пам'яті NOR або OTP ROM також часто включається в чіп, а також невеликий обсяг оперативної пам'яті. Мікроконтролери призначені для вбудованих систем, на відміну від мікропроцесорів, які використовуються в персональних комп'ютерах або інших додатках загального призначення, що складаються з різних дискретних чіпів. Мікроконтролери використовуються в автоматизованих пристроях, таких як автомобільні системи управління двигунами, імплантуються медичні пристрої, пульти дистанційного керування, офісні машини, електроприлади, електроінструменти, іграшки та інші вбудовані системи. За рахунок зменшення розміру і вартості в порівнянні з конструкцією, яка використовує окремі пристрої, як мікропроцесор, пам'ять і пристрої введення / виводу, мікроконтролери дозволяють більш економно

управляти ще більшою кількістю пристроїв і процесів. Таким чином, мікроконтролери забезпечують низьке енергоспоживання, достатню надійність системи, а також дуже гнучко-настроюється логіку завдяки їх програмування. На малюнку 1.1 представлений один з популярних мікроконтролерів. За рахунок зменшення розміру і вартості в порівнянні з конструкцією, яка використовує окремі пристрої, як мікропроцесор, пам'ять і пристрої введення / виводу, мікроконтролери дозволяють більш економно управляти ще більшою кількістю пристроїв і процесів. Таким чином, мікроконтролери забезпечують низьке енергоспоживання, достатню надійність системи, а також дуже гнучко-настроюється логіку завдяки їх програмування. На малюнку 1.1 представлений один з популярних мікроконтролерів. За рахунок зменшення розміру і вартості в порівнянні з конструкцією, яка використовує окремі пристрої, як мікропроцесор, пам'ять і пристрої введення / виводу, мікроконтролери дозволяють більш економно управляти ще більшою кількістю пристроїв і процесів. Таким чином, мікроконтролери забезпечують низьке енергоспоживання, достатню надійність системи, а також дуже гнучко-настроюється логіку завдяки їх програмування. На малюнку 1.1 представлений один з популярних мікроконтролерів. а також дуже гнучко-настроюється логіку завдяки їх програмування. На малюнку 1.1 представлений один з популярних мікроконтролерів. а також дуже гнучко-настроюється логіку завдяки їх програмування. На малюнку 1.1 представлений один з популярних мікроконтролерів.

Хоча мікроконтролери і володіють в більшості випадків низькою вартістю, але це гідність можна перекреслити складністю розробки і створення прототипу для непромислових та приватних проектів, а також відсутністю універсальності в деяких випадках.



Малюнок 1.1 - Мікроконтролер ATtiny2313 американської фірми Atmel

Але поява таких рішень, як, наприклад, Arduino і одноплатні комп'ютери, вивело домашню та непромислові автоматизацію на новий рівень. Істотно знизилася вартість даних систем, без шкоди надійності. Через масове впровадження в повсякденне життя, так само знизилася і вартість датчиків (сенсорів) для них.

1.2.2 Arduino

Arduino [5] - це платформа з відкритим вихідним кодом, яка використовується для створення проектів електроніки. Arduino складається з фізичної програмованої монтажної плати (часто званої мікро контролером) і програмного забезпечення, або IDE (Integrated Development Environment), яка працює на вашому комп'ютері, використовується для запису і завантаження комп'ютерного коду на фізичну плату. Платформа Arduino стала досить популярною серед людей, які тільки починають працювати з електронікою, і не без підстави. На відміну від більшості попередніх програмованих друкованих плат, Arduino не потребує окремого апаратному забезпеченні (так званому програматоре) для завантаження нового коду на плату - ви можете просто використовувати USB-кабель. На малюнку 1.2 представлена одна з моделей Arduino.



Малюнок 1.2 - Arduino Uno - одна з найбільш популярних плат в сімействі Arduino

Крім того, в середовищі IDE Arduino (малюнок 1.3) використовується спрощена версія C ++, що спрощує навчання розробки під цю плату.

A screenshot of the Arduino IDE interface. The window title is 'Blink | Arduino 1.0.3'. The code editor shows the following C++ code for the Blink example:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop(){
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

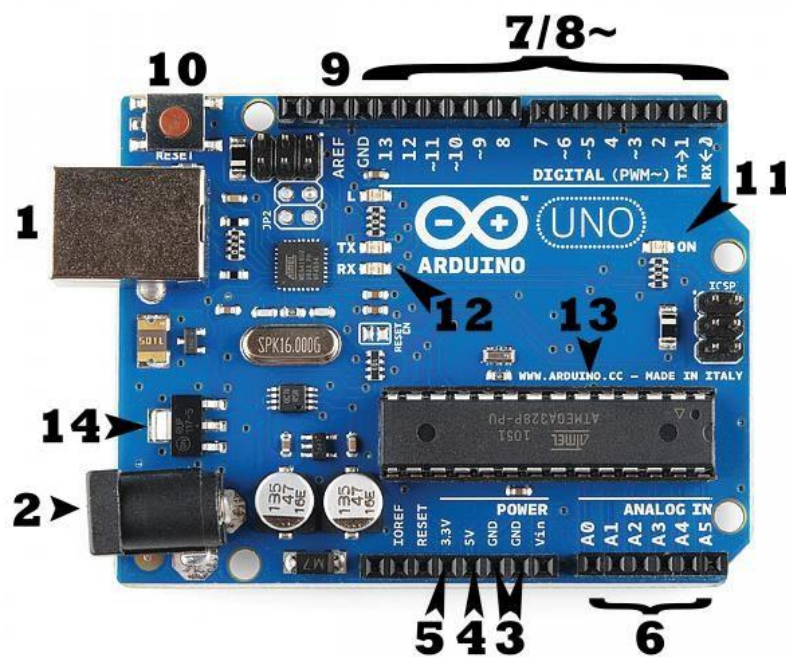
The status bar at the bottom indicates '24 Arduino Mega (ATmega1280) on /dev/tty.usbserial-A600enbz'.

Малюнок 1.3 - скріншот Arduino IDE

Апаратне та програмне забезпечення Arduino було розроблено для художників, дизайнерів, любителів, хакерів, новачків та всіх, хто зацікавлений в створенні інтерактивних об'єктів або середовищ. Arduino може взаємодіяти з кнопками, світлодіодами, моторами, гучномовцями, пристроями GPS, камерами, Інтернетом і навіть вашим смартфоном або телевізором. Ця

гнучкість в поєднанні з тим, що програмне забезпечення Arduino безкоштовне, апаратні плати досить дешеві, а програмне і апаратне забезпечення легко освоїти, призвело до великої спільноти користувачів, які написали код і випустили інструкції для величезного розмаїття проектів на базі Arduino. Для всього, що завгодно: від роботів і гріє коври для обігріву до розумних гаджетів, Arduino можна використовувати як «мозок» практично будь-якого проекту електроніки.

Існує безліч різновидів плат Arduino, які можуть використовуватися для різних цілей. Деякі плати трохи відрізняються від наведених нижче, але більшість мають наступні компоненти (малюнок 1.4):



Малюнок 1.4 - Компоненти плати Arduino
Живлення (USB / Barrel Jack)

Кожній платі Arduino необхідний спосіб підключення до джерела живлення. Пристрій Arduino UNO може харчуватися від USB-кабелю, що йде від вашого комп'ютера, або схожого джерела живлення. На малюнку вище USB- з'єднання позначено (1), а barrel jack позначено (2). USB-з'єднання - це також спосіб

завантаження коду на вашу плату Arduino. Рекомендоване напруга для більшості моделей Arduino становить від 6 до 12 вольт.

Піни (виводи) (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

Піни на Arduino - це місця, де ви поєднуєте дроти для побудови схеми (ймовірно, в поєднанні з макетом і деякими проводами. У них зазвичай є чорні пластикові «заголовки», які дозволяють просто вставляти провід прямо в плату. Є кілька різних видів контактів, кожен з яких позначений на платі і використовується для різних функцій.

- **GND (3):** скорочення від «ground». На Arduino є кілька висновків GND, кожен з яких може використовуватися для заземлення вашої ланцюга.

- **5V (4) і 3.3V (5):** як можна здогадатися, 5-контактний роз'єм живить 5 вольт, а 3,3-контактний - 3,3 вольта. Більшість простих компонентів, які використовуються з Arduino, працюють як від 5, так і від 3.3 вольт.

- **Analog (6):** контактна поверхня від A0 до A5 на UNO - це виводи «Аналоговий вхід». Ці виводи можуть зчитувати сигнал з аналогового датчика (наприклад, датчик температури) і перетворювати його в цифрове значення, яке ми можемо прочитати.

- **Digital (7):** навпроти аналогових контактів знаходяться цифрові виходи (від 0 до 13 в UNO). Ці контакти можуть використовуватися як для цифрового входу (наприклад, при натисканні на кнопку), так і для цифрового виходу (наприклад, при включенні світлодіода).

PWM (8): можливо, ви помітили тильду (~) поряд з деякими цифровими штифтами (3, 5, 6, 9, 10 і 11 в UNO). Ці виступи діють як звичайні цифрові контакти, але також можуть використовуватися для так званої Pulse-Width Modulation (PWM). Ці виводи можуть імітувати аналоговий вихід (наприклад, загасання світлодіода).

- **AREF (9):** його іноді використовують для установки зовнішнього

опорного напруги (від 0 до 5 вольт) в якості верхньої межі для аналогових вхідних контактів.

кнопка перезавантаження

У Arduino є кнопка скидання (10). Натискання цієї кнопки тимчасово підключить контакт скидання до землі і перезапустить будь-який код, завантажений в Arduino. Це може бути дуже корисно, якщо ваш код не повторюється, але ви хочете перевірити його кілька разів.

індикатор живлення

Під і праворуч від слова «UNO» на платі є крихітний індикатор поруч зі словом «ON» (11). Цей світлодіод повинен загорятися кожен раз, коли ви підключаєте Arduino до джерела живлення.

Світлодіодні індикатори TX RX

TX - для передачі, RX - для прийому. Ці маркування з'являються досить нечасто в електроніці, щоб вказати контакти, відповідальні за послідовну зв'язок. У нашому випадку в Arduino UNO є два місця, де з'являються TX і RX - один раз цифровими контактами 0 і 1, і другий раз поруч з світлодіодами індикаторами TX і RX (12). Ці світлодіоди будуть спалахувати щоразу, коли наш Arduino отримує або передає дані (наприклад, коли ми завантажуюмо нову програму на плату).

Основна ІС

Чорна частина з усіма металевими ніжками - ІС, або Integrated Circuit (13). Можна сказати, що це «мізки» Arduino. Основна ІС на Arduino трохи відрізняється на різних типах, але, як правило, це лінія ATmega від ІС від компанії ATMEL. Це може бути важливо, так як вам може знадобитися знати тип ІС (разом з типом вашої плати) перед завантаженням нової програми з програмного забезпечення Arduino. Цю інформацію зазвичай можна знайти в письмовій формі на верхній стороні ІС.

Регулятор напруги

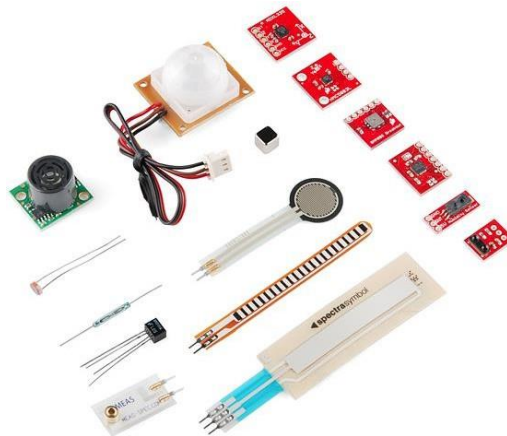
Регулятор напруги (14) насправді не є тим, з чим ви можете (або

повинні) взаємодіяти на Arduino. Але потенційно корисно знати, що він є і для чого він потрібен. Регулятор напруги виконує те, що сказано в його назві - про контролює кількість напруги, яка подається на плату Arduino. Він відключить додаткову напругу, яке може пошкодити ланцюг. Звичайно, у нього є свої обмеження, тому не варто підключати Arduino ні до чого більше 20 вольт.

Датчики

За допомогою деякого простого коду Arduino може управляти і взаємодіяти з широким спектром датчиків (малюнок 1.5) - вимір яскравості світла, температури, ступеня вигину, тиску, близькості, прискорення, чадного газу, радіоактивності, вологості, атмосферного тиску.

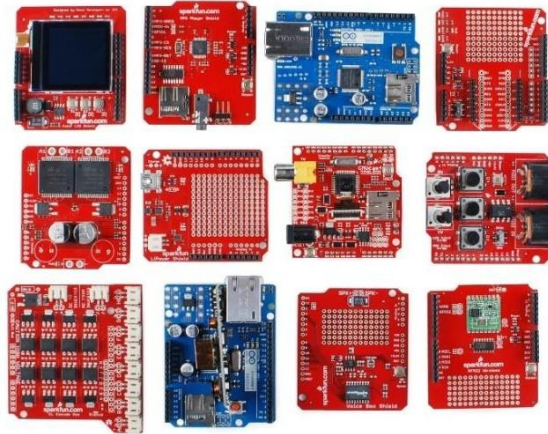
Малюнок 1.5 - Деякі датчики, які легко взаємодіють з Arduino



Плати розширення (shields)

Крім того, є такі плати (малюнок 1.6), які називаються shields - в основному це заздалегідь підготовлені монтажні плати, які поміщаються поверх Arduino і надають додаткові можливості управління двигунами, підключення до

Інтернету, надання стільникового або іншого бездротового зв'язку, управління ЖК-екраном і багато іншого.



Малюнок 1.6 - Частковий вибір доступних плат для розширення можливостей Arduino

Таким чином , можливо виділити наступні гідності використання Arduino:

- *Готовність до використання:* готова до використання структура робить її дуже простий у використанні. Вам не потрібно турбуватися про програматоре, настройках запобіжників, програмних засобах серійного монітора і т. Д.
- *Приклади коду:* Багато стандартних бібліотек, які полегшують і прискорюють розробку. Також варто відзначити, що це open-source продукт.
- *легкі функції:* В програмному забезпеченні Arduino є багато функцій, які роблять кодування настільки простим і швидким, що це неможливо при використанні простого мікроконтролера.
- *велике співтовариство:* Є багато форумів в Інтернеті, на яких люди говорять про Arduino. Інженери, любителі і професіонали роблять свої проекти через Arduino. Ви можете легко знайти допомогу у всьому. Крім того, сам сайт Arduino пояснює всі функції Arduino.
- Гарна підтримка IoT-концепції.

Також варто виділити недоліки цієї платформи:

- *структура* Arduino також є її недоліком. Зазвичай під час створення проекту ви повинні постаратися зробити його розмір якомога меншим. Але з великими структурами Arduino ми повинні дотримуватися великих розмірів друкованих плат. Якщо ви працюєте на невеликому мікроконтролері, такому як ATmega8, ви можете легко зробити свою РСВ настільки маленькою, наскільки це можливо.

- *вартість*. Найважливіший фактор, який не можна заперечувати - це вартість. Це проблема, з якою стикається кожен любитель, інженер або професіонал. Іноді для деяких рішень вигідніше організувати рішення на мікроконтролері.

1.2.3 Одноплатні комп'ютери на прикладі Raspberry Pi

Останнім часом в області автоматизації та робототехніки стали популярні одноплатні комп'ютери [6].

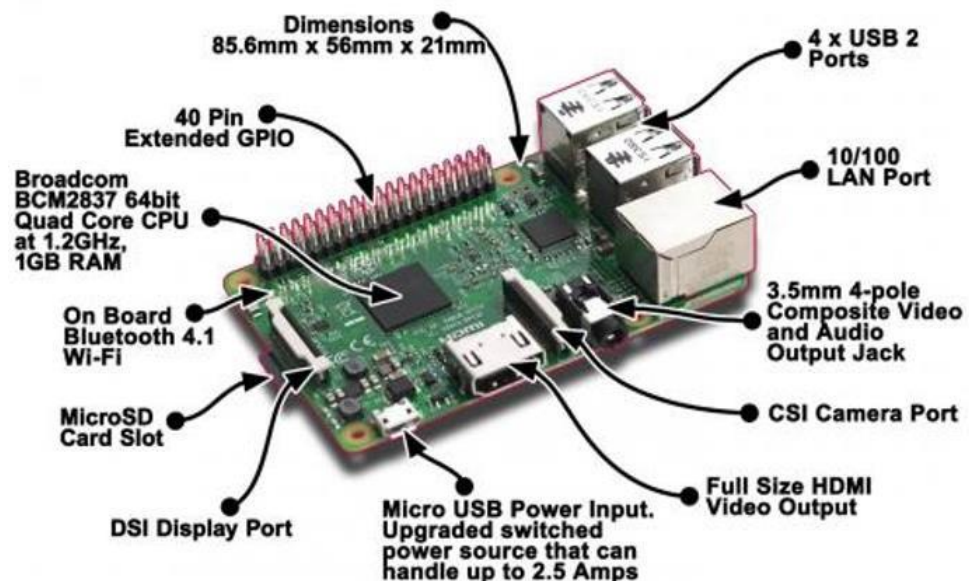
Одноплатний комп'ютер являє собою повноцінний комп'ютер, побудований на одній монтажній платі, з мікропроцесором, пам'яттю, введенням / висновком (І / О) і іншими функціями, необхідними для функціонального комп'ютера. Одноплатні комп'ютери були створені в якості демонстраційних або систем розробки, для освітніх систем або для використання в якості вбудованих комп'ютерних контролерів. Багато типів домашніх комп'ютерів або переносних комп'ютерів об'єднують всі свої функції на одній друкованій платі.

На відміну від настільного персонального комп'ютера, комп'ютери з одноплатними комп'ютерами часто не мають слоти розширення для периферійних функцій або пристроїв. Одноплатні комп'ютери побудовані з використанням широкого спектра мікропроцесорів. Прості конструкції, наприклад, створені комп'ютерними любителями, часто використовують статичне ОЗУ і недорогі 8- або 16-розрядні процесори. Інші типи, такі як блейд-сервери, працюють аналогічно серверного комп'ютера тільки в більш

компактному форматі.

В даний час на ринку досить багато популярних моделей від різних фірм. Одні з найпопулярніших (різні моделі і варіації): Raspberry Pi, BeagleBone, Orange Pi, Banana Pi, Asus Tinker Board. Всі вони відрізняються ціною, технічними характеристиками, дизайном і т.д. Ми будемо орієнтуватися на лідера з продажу та популярності - Raspberry Pi, який має відносно конкурентів середню ціну і продуктивність, але при цьому володіє найбільшим співтовариством і open-source ПО.

Raspberry Pi [7] - це крихітний комп'ютер розміром з кредитну карту, плата має процесор, ОЗУ і типові апаратні порти, які ви знайдете на більшості комп'ютерів. Головна оперативна система для Pi – Raspbian, яка заснована на Debian. Хоча основний підтримуваної операційною системою є Raspbian, ви можете встановити інші операційні системи, такі як Ubuntu mate, Ubuntu Core, OSMC, RIS OS, Windows 10 IoT і багато іншого. До теперішнього часу існують кілька версій цього комп'ютера, які відрізняються розмірами, ціною і технічними характеристиками.



Малюнок 1.7 - Raspberry Pi 3 model B
Raspberry Pi 3 (малюнок 1.7) є третім поколінням. Він замінив

Raspberry Pi 2 Model B в лютому 2016 року. У порівнянні з Raspberry Pi 2 він має:

- 1.2-гігабітний 64-розрядний чотирьохядерний процесор;
- ARMv8 802.11n Wireless LAN;
- Bluetooth 4.1;
- Bluetooth Low Energy (BLE); Як і Pi 2, він також має:
- 1 ГБ оперативної пам'яті;
- 4 порти USB;
- 40 контактів GPIO;
- порт Full HDMI;
- Порт Ethernet;
- комбінований 3,5-мм аудіороз'єм і композитний відеоінтерфейс;
- Інтерфейс камери (CSI);
- Інтерфейс дисплея (DSI);
- Гніздо для карти Micro SD (тепер push-pull, а не push-push);
- Графічне ядро VideoCore IV 3D;

The Raspberry Pi 3 має ідентичний форм-фактор з попередніми Pi 2 (і Pi 1 Model B +) і має повну сумісність з Raspberry Pi 1 і 2. Також всі сімейство має дуже багато зовнішніх модулів, які розширюють функціонал цього комп'ютера. Наприклад, плата Gertboard (малюнок 1.8) дозволяє розширити кількість ліній вводу / виводу краще за інших можливих рішень. Основною особливістю тут є 28-вивідний мікроконтролер Atmega (будь зі списку наступних моделей ATmega 48A / PA, 88A / PA, 168A / PA або 328 / P). Завдяки підключенню Atmega розширюваність, притаманна інтегрованому середовищі розробки плати Arduino, тепер доступна і Raspberry Pi.



1. Легко почати роботу з відмінною онлайн-підтримкою, швидке, просте створення прототипів.

2. Просте поєднання з датчиками і збір даних.

3. Дешевше, ніж Raspberry Pi (для продуктів, які не мають можливості підключення до Інтернету).

4. Можна відправляти дані без проводів, використовуючи Bluetooth, Rf і т. Д. На сервер через комп'ютер.

5. Безліч GPIO з можливостями PWM і дружнім до розробників.

6. Ви можете використовувати IDE, python, ruby, embedded C і т.д. для програмування.

7. Повністю відкритий вихідний код

мінуси Arduino:

1. Обмеження пам'яті (вкрай мало в порівнянні з Raspberry Pi).

2. Для підключення до Інтернету потрібні додаткові shields.

3. Менш потужний у порівнянні з Raspberry Pi. Не можна запустити багато важких алгоритмів, або програмувати сенсорний екран і використовуючи додаткові плати розширення. Наприклад, ви не можете запустити Open CV.

плюси Raspberry Pi:

1. Хороша продуктивність. Пам'ять із.
2. ОС, заснована на Linux, а тепер навіть Windows 10, яка може запускатися поверх неї, щоб зробити обробку більш зручною для користувача.
3. Безліч доступних GPIO і чим більше GPIO, тим більше датчиків ви можете з'єднати.
4. Якщо у вас є досвід роботи з Linux, дуже легко почати розробку, інакше буде потрібно якийсь час, щоб розібратися в цьому.
5. Python, C, C ++, Ruby, Go і багато інших можуть бути використані для програмування Pi саме так, як ви можете програмувати на будь-якому комп'ютері.
6. Люди успішно використовують Raspberry Pi для запуску Open CV, алгоритмів інтелектуального аналізу даних і т. Д. І пов'язують результати з різними додатками.
7. З точки зору вартості, краще, ніж arduino з платою розширення Ethernet.
8. Відмінне онлайн-спільнота і нескінченні можливості того, що можна зробити з його допомогою.

мінуси Raspberry Pi:

1. Потрібно добре знання Linux-систем.
2. Для більшості додатків продуктивність буде надмірною, оскільки використовуватиметься в основному тільки передача даних.
3. Закритий вихідний код.
4. Вимагає більше енергії.

Таким чином, можна зробити висновок, що обидві платформи підходять для вибору їх за апаратну основу. Але для автоматизації теплиці з вирішенням поставлених завдань, як моніторинг показників датчиків з будь-якої точки світу (переважно, за допомогою сайту) і корекцією поведінкою в реальному часі, доцільно використовувати Raspberry Pi або інший одноплатний комп'ютер з ОС з наступних причин:

– Один з головних плюсів: Raspberry Pi можна використовувати як веб-сервер, що знімає з нас завдання пошуку, настройки та управління додатковим обладнанням. До того ж це понесе додаткові витрати.

– Крім серверної складової, для Raspberry Pi є фреймворки з відкритим вихідним кодом для організації клієнтської частини без глибоких знань в цій області (як, наприклад, WeBIOPi, який також відповідає і за серверну частину).

– Доступних портів GPIO вистачить для підключення будь-яких цифрових датчиків, а для аналогових є можливість використання додаткових недорогих АЦП.

– Підтримка багатозадачності.

– Багато можливостей організації і з'єднання з різними веб-сервісами, які можуть виявитися корисними (SMS-оповіщення, зв'язок з twitter і ін.).

Після вибору апаратної бази нам необхідно визначитися з такими компонентами як датчики й устаткування для контролю мікроклімату теплиці.

2 Створення мікроклімату в теплиці

Для автоматизації теплиці в першу чергу потрібні дані, на основі яких будуть включатися модулі для створення необхідних умов мікроклімату для росту рослин. Для цього нам потрібні різні датчики, а також інше обладнання, наприклад, аналогово-цифровий передавач для підключення аналогових датчиків, а також камера, за допомогою якої ми будемо спостерігати за тим, що відбувається.

За керування теплицею, а якщо бути точним, створення потрібного мікроклімату, може відповідати різне периферійне устаткування в залежності від необхідних вимог і фінансових можливостей, але основні функції, які це обладнання повинно виконувати: полив, зволоження повітря, провітрювання, регулювання освітлення та підтримка необхідної температури .

У цьому розділі пояснюються використовувані датчики в нашому

проекті, а також можливі варіанти створення необхідних умов з допомогою автоматизації для створення мікроклімату в теплиці.

2.1 Моніторинг показників в теплиці

Датчиків, які називаються також сенсорами, існує величезна безліч і в першу чергу вони мають різне призначення.

Основні датчики (за призначенням):

- *Датчик тиску* - використовується для виявлення фізичного тиску, наприклад, при щипки, стисканні, поштовхах.

- *фотоелементи* - використовується для вимірювання рівня освітленості, виявлення простого об'єкта за принципом світлий / темний

- *Датчик температури* - використовується для визначення температури навколишнього середовища або, наприклад, рідини

- *датчик вібрації* - використовується для виявлення руху / вібрації і орієнтації

- *Датчик руху* - використовується для виявлення рухової активності, таких як тварин або людей

- *термопари* - використовується для вимірювання температури, як правило, вище 150 ° C.

- *ИК-приймачі* - використовується для виявлення інфрачервоних- сигналів від пульта дистанційного керування.

Це далеко не всі за призначенням датчики. Існують також датчики визначення вологості, вогню, диму і т.д. В принципі, можна підібрати датчики практично під будь-яке завдання.

Всі датчики мають свої власні методи взаємодії. Деякі можна просто підключити до плати, а підключення інших вимагає додаткових маніпуляцій. Наприклад, можуть знадобитися резистори або додаткові джерела живлення.

Датчики відрізняються за типом сигналу на виході: аналоговий або цифровий.

Вибираючи датчики, можна помітити, що найчастіше згадується сумісність з Arduino і не згадується сумісність з Raspberry Pi. Якщо датчик видає цифровий сигнал, то його можна підключати до GPIO-портам Raspberry Pi. Якщо ж датчик видає аналоговий сигнал, то підключити безпосередньо такий датчик до Raspberry Pi можна тільки за допомогою аналого-цифрового перетворювача (АЦП), який часто включають в плати розширення Raspberry Pi.

Різні датчики можуть сильно відрізнятися за ціною, вартість датчика може становити \$ 1 або \$ 5, датчики з дуже великою точністю визначення сигналів можуть коштувати значно дорожче.

Далі наведемо список обраних нами датчиків для моніторингу показників теплиці і опис того, що вони вимірюють.

цифрові:

- AM2320 [8] - вологість і температура повітря;
- BH1750 [9] - освітленість приміщення;
- BMP180 [10] - температура і тиск повітря (друге нас цікавить в першу чергу, тому що для вимірювання температури використовуються і інші датчики);
- DS18B20 [11] - температура ґрунту.

аналогові:

- MQ-2 Gas Sensor [12] - детектор витoku газу;
- Moisture Sensor [13] - вологість ґрунту.

Оскільки Raspberry Pi не має аналогових входів, то нам знадобився аналогово-цифровий перетворювач, в ролі якого виступав ADS1115 [14].

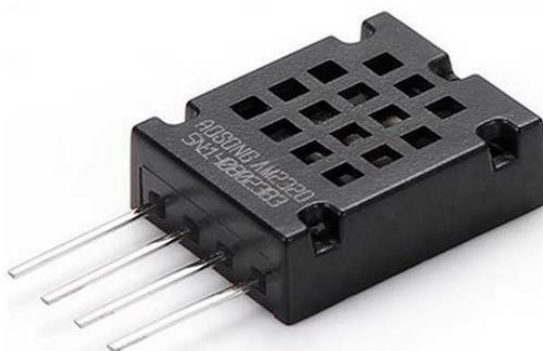
Також була використана камера Raspberry Pi Camera Board [15], за допомогою якої будуть робитися знімки і виводиться на призначений для користувача інтерфейс.

Для всіх датчиків були написані скрипти на мові Python, приклад коду деяких представлений в додатку А. Підключення ж самих датчиків з описом контактів описано і показано в додатку Б.

AM2320

AM2320 - це цифровий комбінований датчик температури і вологості (малюнок 2.1).

Датчик складається з ємнісного елемента вимірювання вологості і інтегрального елемента вимірювання температури. Обидва ці елементи з'єднуються з високопродуктивним процесором, який перетворює виміряні аналогові значення в цифровий сигнал. Датчик має хорошу якість виготовлення, швидкий час відгуку і володіє помехозащищенностью для застосування в промисловості.



Малюнок 2.1 - AM2320

Для з'єднання контролера з датчиком необхідно використовувати стандартну шину I2C. Мале енергоспоживання датчика дозволяє встановити кілька датчиків на шину на відстані до 20 метрів від контролера.

Таблиця 2.1 - Специфікація AM2320

характеристика	значення
Напруга живлення	3,1 - 5,5 В
струм	20 мА
Діапазон вимірювання температури	від - 40 до 80 ⁰ З
Максимальна похибка вимірювання температури	0,5 ⁰ З
Діапазон вимірювання вологості	від 0 до 100%
Усереднена похибка вимірювання вологості	3%

Характеристики:

- Дуже малий розмір;
- Економічна робота;
- Наднизьке напруга;

- Відмінна довготривала стабільність;
- Стандарт I2C і вихід на одну шину;
- Похибка $\pm 0,5 \text{ }^{\circ} \text{C} / \pm 3\% \text{ RH}$;
- Менший розмір, ніж DHT11 / 12, але більш висока точність.

ВН1750

Іноді виникає необхідність отримати дані про рівень освітленості конкретного місця, об'єкта. Для таких цілей використовуються спеціальні прилади - люксметри. Але це не таке вже й поширене обладнання. Освітленість можна спробувати вимірювати фоторезистором GL5516, наприклад. Але в такому випадку необхідно робити вимірювальну схему і калібрувати отримані дані з такого датчика. Спростити завдання нам допоможе датчик, який за допомогою цифрового інтерфейсу видаватиме

готові дані в певних одиницях виміру - ВН1750 (малюнок 2.2).

Датчик цей укладений в корпусі для поверхневого монтажу WSOF61. Усередині безпосередньо сам фотодатчик у вигляді фотодіода, підсилювач сигналу фотодіода, АЦП (аналого-цифровий перетворювач) і якась логіка, яка обробляє дані, переводить все в одиниці виміру Люкс і передає по I2C до пристрою.

Малюнок 2.2 - ВН1750



ВН1750 володіє наступними характеристиками:

- Цифровий інтерфейс - I2C
- Висока роздільна здатність - до 0,5 Лк
- Малий споживаний струм і функція сплячого режиму
- Фільтрація світлових шумів 50/60 Гц
- Мала залежність від джерела світла (лампа розжарювання, світлодіод

і так далі)

- Малий вплив інфрачервоного випромінювання
- Можливо вибрати 2 адреси мікросхеми для I2C інтерфейсу (можна підключити одночасно два таких датчика до однієї шині)

• Не потребує калібрування, що максимально зручно для застосування в будь-яких проектах

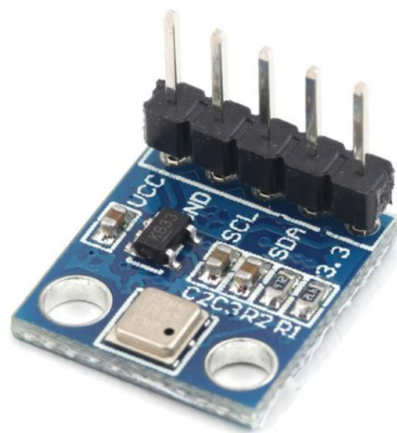
- Дуже малі габарити датчика Електричні характеристики:
- Напруга живлення - 2,4 - 3,6 В
- Струм споживання - 120 мкА

- Струм споживання в режимі очікування - 0,01 мкА
- Вимірюється довжина хвилі - 560 нм
- Точність в режимі високої роздільної здатності - 1 Лк
- Точність в режимі низького дозволу - 4 Лк
- Період вимірювання в режимі високої роздільної здатності - 120 мс
- Період вимірювання в режимі низького дозволу - 16 мс
- АЦП - 16 біт

ВМР180

Цей високоточний датчик від Bosch - краще недороге рішення для вимірювання барометричного тиску і температури. Датчик припаяний до друкованої плати з регулятором 3,3 В, перемикачем рівня

I2C і навантажувальними резисторами на висновках I2C. ВМР180 (малюнок 2.3) являється датчиком нового покоління від Bosch і замінює ВМР085. Він повністю ідентичний ВМР085 з точки зору прошивки / програмного забезпечення. Ця плата відповідає стандарту 5 В - включений регулятор 3,3 В і схема перемикання рівня I2C, тому можна безпечно використовувати цей датчик з логікою і потужністю 5 В.



Малюнок 2.3 - ВМР180 Характеристики датчика ВМР180:

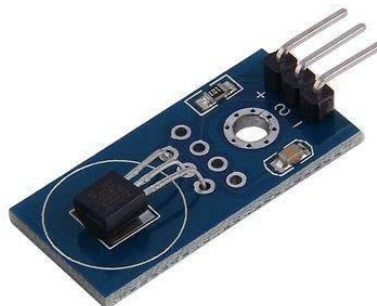
- Діапазон вимірювання тиску: 300 - 1100hPa
- Напруга живлення: від 3.3 і 5V
- низьке енергоспоживання: 3 мкА (режим ультранизького енергоспоживання)

- точність: режим зниженого енергоспоживання, Дозвіл 0.06hPa (0,5 м)

- Високий лінійний режим з дозволом 0.02hPa
- Підключення: протокол зв'язку I2C
- Можливість вимірювання температури в діапазоні -40 ... +85 ° C
- Час відгуку: 5ms
- Струм в режимі очікування: 0,1 мкА

DS18B20

DS18B20 (малюнок 2.4) - цифровий термометр з програмованим дозволом, від 9 до 12-bit, яке може зберігатися в EEPROM пам'яті приладу. DS18B20 обмінюється даними по 1-Wire шині і при цьому може бути, як єдиним пристроєм на лінії, так і працювати в групі. Всі процеси на шині управляються центральним мікропроцесором. Діапазон вимірювань від -55 ° C до +125 ° C і точністю 0.5 ° C в діапазоні від -10 ° C до +85 ° C. На додаток, DS18B20 може харчуватися напругою лінії даних ("parasite power"), при відсутності зовнішнього джерела напруги. Кожен DS18B20 має унікальний 64-бітний послідовний код, який дозволяє, спілкуватися з безліччю датчиків DS18B20 встановлених на одній шині. Такий принцип дозволяє використовувати один мікропроцесор, щоб контролювати безліч датчиків DS18B20, розподілених за великим ділянці. Додатки, які можуть отримати вигоду з цієї особливості,



Малюнок 2.4 - DS18B20 Характеристики DS18B20:

- Діапазон вимірюваних температур: $-55 \dots + 125 \text{ }^\circ\text{C}$
- Точність: $\pm 0,5 \text{ }^\circ\text{C}$ (в межах $-10 \dots + 85 \text{ }^\circ\text{C}$)
- Час отримання даних: 750 мс при 12-бітному дозволі; 94 мс при 9-бітному дозволі
- Напруга живлення: 3-5,5 В
- Струм при бездіяльності: 750 нА
- Струм при опитуванні: 1 мА

2.1.5 Аналогово-цифровий перетворювач ADS1115

ADS1115 (малюнок 2.5) - відмінний аналого-цифровий перетворювач, який дає можливість легко використовувати з Raspberry Pi аналогові датчики, використовуючи його комунікаційну шину I2C. ADS1115 є високоточний 16-розрядний АЦП з 4-ма каналами. Він має програмований посилення від 2 / 3х до 16х, тому можна посилити невеликі сигнали і читати їх з більшою точністю.

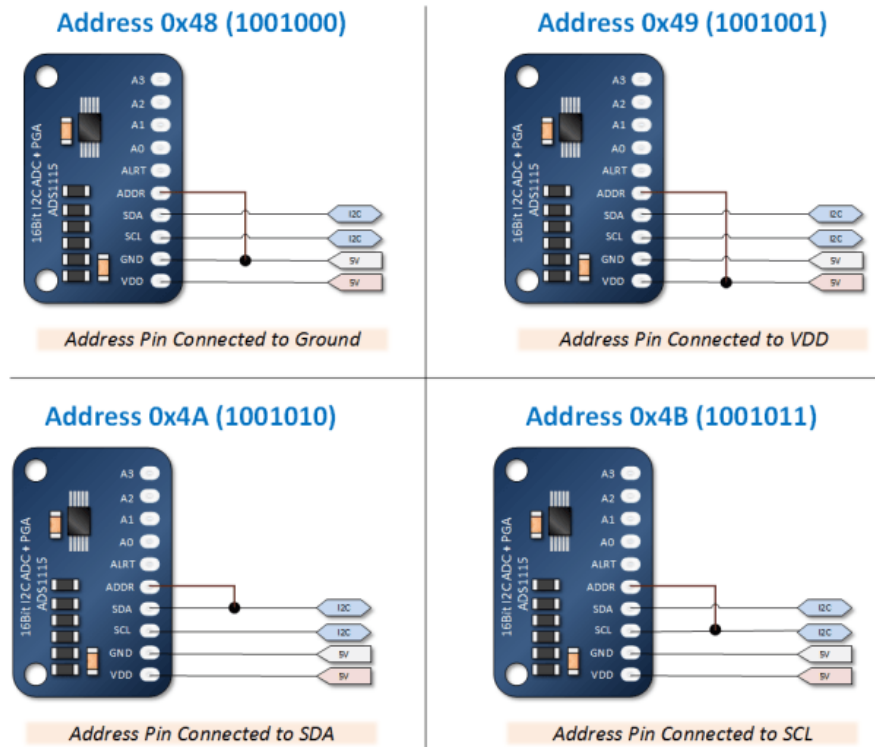
АЦП видає 16-розрядний знакового значення напруги на вході. Т. е. Максимальна величина шкали напруга не 65535, а 32768. Відповідно якщо необхідно використовувати шкалу на все 16-біт можна тільки за умови диференціального вимірювання, де на один з входів буде підключений зовнішній джерело опорного напруги, а другий буде вимірювальним.



Малюнок 2.5 -АЦП ADS1115

Цей пристрій передбачає установку адреси для I2C шини за допомогою замикання виведення ADDR на цифрові сигнальні виводи або виводи шини живлення (малюнок 2.6). Установка різних адрес I2C призначена для того,

щоб при підключенні двох і більше різних пристроїв не відбувалося конфлікту між ними, т. К. В кращому випадку модулі у яких адреси співпадуть просто не будуть працювати, а в гіршому - робота всіх модулів на шині може стати непередбачуваною.



Малюнок 2.6 - Вибір адреси для АЦП

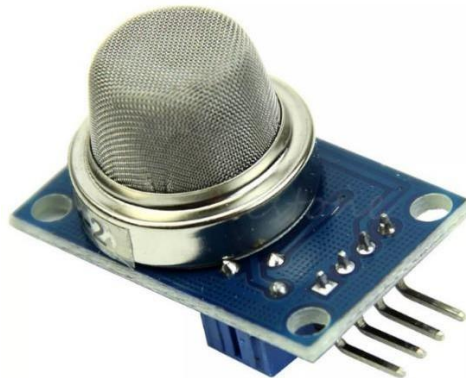
ADS1115 Характеристики АЦП ADS1115:

- Широкий діапазон електроживлення: від 2,0 до 5,5 В
- Мале споживання струму, безперервний режим: тільки 150 мкА
- Програмована швидкість передачі даних: 8sps - 860sps
- Внутрішнє низькотемпературне напруга
- внутрішній осцилятор
- Внутрішній інтерфейс PGA
- I2C інтерфейс: пін-які обираються адреси
- Чотири одиночних або два диференціальних входи
- програмований компаратор
- Ця плата / чіп використовує 7-бітові адреси I2C між 0x48-0x4b,

обрані за допомогою перемичок

2.1.6 MQ-2 Gas Sensor

Датчик газу, побудований на базі газоаналізатора MQ-2 (малюнок 2.7) дозволяє виявляти наявність в навколишньому повітрі вуглеводневих газів (пропан, метан, н-бутан), диму (зважені частинки, які є результатом горіння), водню.



Малюнок 2.7 - MQ-2

Датчик можна використовувати для виявлення витоків промислового газу і задимлення. Вихідним результатом є аналоговий сигнал, пропорційний змісту газів, до яких чутливий газоаналізатор. Чутливість може бути налаштована за допомогою триммера на платі датчика.

Показання сенсора схильні до впливу температури і вологості навколишнього повітря. Тому в разі використання датчика газу в мінливому середовищі, при необхідності отримання точних показань, знадобиться реалізувати компенсацію цих параметрів.

Діапазон вимірювань:

- Пропан: 0,2 - 5 проміле
- Бутан: 0,3 - 5 проміле
- Метан: 5 - 20 проміле
- Водень: 0,3 - 5 проміле
- Пари спиртів: 0,1 - 2 проміле

Характеристики
MQ-2:

- Напруга живлення нагрівача: 5 В
- Напруга живлення датчика: 3,3-5 В
- Струм: 150 мА
- Габарити: 25,4 × 25,4 мм

2.1.7 Moisture sensor

Moisture Sensor (малюнок 2.8) - датчик для вимірювання вологості ґрунту. Виріб виконаний у вигляді вилки, яку для проведення вимірювань легко застромити в ґрунт. Опір сухого ґрунту дуже велике. При збільшенні вологості опір ґрунту знижується. Датчик побудований на основі транзисторного підсилювача струму. Він включений в ланцюг між колектором і базової біполярного прп транзистора. При зміні вологості ґрунту змінюється опір між колектором, до якого підключений позитивний полюс джерела живлення, і базою. Змінюється струм бази і, відповідно, висока напруга, яка від колектора через емітер на землю. В результаті змінюється вихідна аналогове напруга, яке може бути оцифровано за допомогою АЦП. Виріб може застосовуватися в системах автоматичного поливу, що нам і потрібно.



Малюнок 2.8 - Moisture Sensor

Характеристики Moisture Sensor:

- Глибина виявлення: 38мм;
- Напруга живлення: 2В - 5В;
- Монтажні отвори: 2 мм;
- Розміри: 20 х 51 мм.

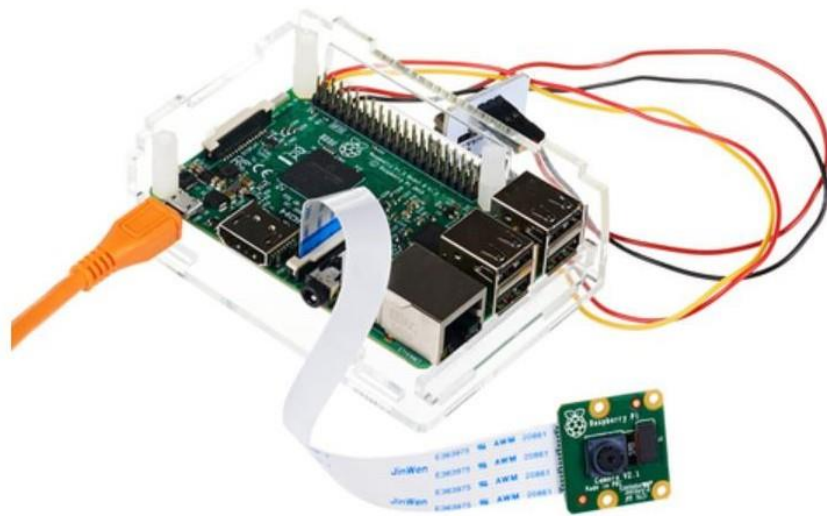
2.1.8 Raspberry Pi Camera Board

Нам необхідно робити і зберігати знімки за допомогою Raspberry Pi. Однак, з міркувань розмірів, а також, способу підключення, стандартна веб-камера може не підійти. Справа в тому, що підключення по USB буде витрачати значну частину ресурсу центрального процесора Raspberry Pi.

Тому, в схемі Raspberry Pi був спочатку закладений роз'єм під спеціальний паралельний інтерфейс, мінімально вповні процесор. Саме під нього була розроблена Raspberry Pi Camera Board.

Версія камери v.2.1 (малюнок 2.9) оснащена восьмимегапіксельним сенсором Sony IMX219 Exmor, що дозволяє захоплювати, записувати і транслювати відео в форматах 1080p, 720p і VGA. Для фотографій максимальний дозвіл кадрів становить 3280 × 2464 пікселів.

Модуль сумісний з усіма моделями Raspberry Pi 1, 2 і 3.



Малюнок 2.9 - Raspberry Pi Camera

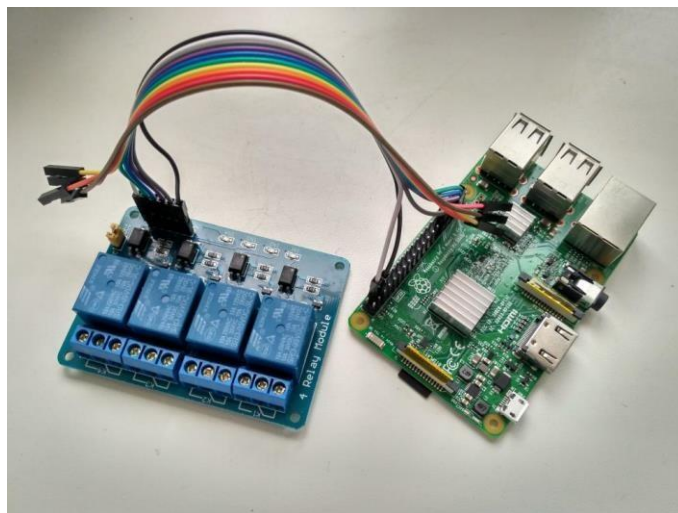
Board v2.1 Характеристики Raspberry Pi Camera Board:

- Тип сенсора: Sony IMX 219 PQ CMOS, ¼ дюйма;
- Максимальна роздільна здатність: 8 Мп (3280 × 2464);
- Підтримувані формати відео: 1080p (30fps), 720p (60fps), 640 × 480p (90fps);

- Еквівалентна фокусна відстань: 33 мм;
- Світлосила об'єктива: $f / 2$;
- Розміри: 2592×1944 ;
- Вага: 3 р

2.2 Автоматизація створення мікроклімату

Якщо процес моніторингу показників ще можна якось стандартизувати і створити універсальну систему, то процес створення мікроклімату для кожної конкретної теплиці унікальний. Це виходить з того, що буде в такій теплиці вирощуватися, які у неї розміри, які контрольні значення будуть для кожного з показників і головне, які процеси необхідно автоматизувати і яким обладнанням необхідно управляти. Але незважаючи на відмінність самого обладнання, зазвичай використовуються модулі реле, за допомогою якого це обладнання буде включатися / вимикатися. Також ми не обмежені в кількості каналів, в даний час на ринку представлені модулі як на 4 канали (малюнок 2.10), так і на 8, 16 і більше. Проблема підключення реле до портів GPIO Raspberry Pi не складає ніяких проблем, і нам залишається тільки посилати логічні «0» або «1» для зміни стану потрібного модуля.



Малюнок 2.10 - Модуль реле з 4 каналами

Але в незалежності від обладнання, самі алгоритми автоматизації

можна узагальнити. Тому далі зробимо невеликий огляд на варіанти реалізацій деяких функцій, які можуть бути використані для автоматизації теплиці, не розглядаючи конкретну обладнання.

Автоматичний полив: найпростіший варіант для автоматизації поливу - це задати час початку поливу і його тривалість. Але в такому випадку система не є гнучкою, ніяк не враховуються показання з датчиків. Тому більш оптимальний варіант - полив при досягненні вологості ґрунту нижче контрольного значення. При цьому треба врахувати факт того, що наш датчик вологості ґрунту може перестати давати коректні значення (наприклад, поломка), через що може виникнути небезпека затоплення. Можливі варіанти виходу з цієї ситуації:

- Після чергового поливу відключати його можливість на деякий проміжок часу;

- Задати деяке число поливів і кожен раз через малий проміжок часу робити контрольний замір вологості ґрунту. Якщо вологість ґрунту не змінюється - значить з нашим датчиком щось трапилося, тому слід відключити можливість поливу до усунення причини несправності.

провітрювання: проблема аналогічна поливу - можна відкривати вікна або включати вентилятори на певний час в потрібний час, а можна орієнтуватися на показники датчиків, щоб досягти потрібної вологості і температури повітря з налаштуванням запасного варіанту на випадок відмов датчиків. Також при наявності датчика-аналізатора газів можна орієнтуватися на певний рівень вуглекислого газу або створити певний алгоритм для повного провітрювання в разі виявлення токсичних речовин або диму.

Підтримка потрібної температури: як уже було сказано, досягти потрібної температури в деяких випадках можна за допомогою провітрювання приміщення. Але якщо теплиця знаходиться в холодній зоні, то може знадобиться зробити додатковий обігрів, який буде досягатися

включенням нагрівальних модулів.

Автоматичний контроль освітлення: параметри освітлення підбираються під кожен вид рослин індивідуально, для досягнення потрібного ефекту додаткове освітлення може включатися тільки в певний час доби, а може повністю контролюватися з гнучкою настроюванням яскравості завдяки датчику, який був описаний раніше.

Після того, як будуть точно визначені необхідні умови, можна створити алгоритм для нашої системи. Відразу варто сказати, що з першого разу навряд чи вийде врахувати всі чинники і правильно організувати автоматизацію, її треба буде доопрацьовувати методом проб і помилок. Сам же алгоритм можна реалізувати за допомогою створення програми. Але в нашому випадку використовувався графічний конфігуратор Node-RED [16], який дозволив об'єднати скрипти для зняття показань з датчиків в єдину систему, реалізувати наочний алгоритм автоматизації периферійного обладнання для створення мікроклімату, а також виступив платформою для створення веб-додатки. Про все це підемо мова в наступному розділі.

3. Організація єдиної системи автоматизації з веб- інтерфейсом за допомогою графічного конфігуратора Node-RED

Після того, як ми визначилися з нашими датчиками і пристроями, підключили їх до Raspberry Pi, навчилися їх використовувати (прочитувати свідчення, відправляти сигнали на реле), необхідно створити систему, яка буде об'єднувати всі ці пристрої і скрипти до них, створювати необхідні умови мікроклімату за допомогою відправки керуючих сигналів на периферійні пристрої та виводити інформацію на призначений для користувача інтерфейс, до якого потрібно мати віддалений доступ.

Для цього можна організувати сервер на іншому пристрої (ПК) або орендувати його у сторонньої організації, а потім на нього з нашого Raspberry Pi відправляти інформацію з наступною обробкою на сервері і

виведення її на призначений для користувача інтерфейс за допомогою класичних методів організації веб-додатки за допомогою програмування . Але це понесе як додаткові витрати, так і додаткові труднощі з налаштуванням серверної і клієнтської частини, адже для побудови якісного і функціонального веб-додатки необхідно знати багато технологій і мови програмування.

Інший спосіб - організація сервера на самому Raspberry Pi, тому що на ньому встановлений дистрибутив Linux, що дозволяє його використовувати як повноцінний комп'ютер. Ми можемо встановити популярні сервера, як, наприклад, Apache, який вміє виводити HTML-файли (через HTTP), а якщо підключити до нього додаткові модулі, то він зможе виводити і динамічні веб-сторінки, використовуючи для цього скриптові мови на зразок PHP.

Або ж ми можемо використовувати фреймворки, як, наприклад, WebIOPi. Він був спеціально розроблений для роботи з портами GPIO Raspberry Pi і дозволяє створювати на своїй основі різні користувацькі додатки. WebIOPi має наступні можливості:

- Вбудований Web-сервер, реалізований на мові Python;
- Вбудована підтримка більше ніж 30 пристроїв з інтерфейсами UART, SPI, I2C, 1-Wire;
- Бібліотеки Javascript / HTML для створення Web-інтерфейсу;
- Бібліотеки Python / Java для створення додатків під Android;
- Сумісність з мовою Python версій 2 і 3;
- Можливість адаптації під потреби користувачів. Структура фреймворка WebIOPi приведена на малюнку 3.1:



Малюнок 3.1 - Структура WebIOPi

Хоча WebIOPi і включає в себе вбудований сервер, зручні бібліотеки і інтеграцію з GPIO, але використання цього фреймворка передбачає все ще добре знання Javascript / HTML для створення веб-інтерфейсу.

Доброю і цікавою альтернативою є Node-RED, що набирає останнім часом великої популярності для нескладних систем автоматизації, інтегрованих з мережею інтернет, адже він спочатку розроблявся для IoT, що і стало основою вибору програмної частини нашої теплиці.

3.1 Інтеграція периферійних пристроїв з Node-RED

Топологія зв'язку пристроїв один з одним, а також з зовнішнім світом за допомогою мережі Інтернет може бути будь-який - від найпростішої до дуже складної. Все обмежується лише фантазією автора такої системи. Однак навіть для побудови дуже простий системи потрібне знання основ програмування, а також розробка алгоритму поведінки того чи іншого пристрою. Щоб спростити процес створення такої системи, інженерами IBM була створене таке середовище програмування як Node-RED. Вона дозволяє навіть людині з мінімальними навичками програмування зв'язати між собою різні пристрої класу IoT і запрограмувати їх поведінку. Зручність цього середовища полягає в тому, що сам алгоритм роботи програми можна вибудувати у вигляді візуальних блоків, а також тим, що в репозиторії Node-RED є величезна кількість вже створених модулів, які можна використовувати,

Дане рішення зручно як проміжне для зв'язку пристроїв різного типу між собою і / або ж з системою автоматизації або, наприклад, СУБД або іншим хмарою. З використанням додаткових пакетів Node-RED можна використовувати для створення простих систем автоматизації.

Node-RED працює на Node.JS, і був розроблений для роботи на відносно малопродуктивних системах, таких як:

- Raspberry Pi;
- BeagleBone Black;
- Arduino.

З урахуванням озвучених факторів Node-RED зручно використовувати на шлюзах між різними мережами пристроїв інтернету речей функціонують на власних, як правило, простіших протоколах і традиційним інтернетом, побудованих на TCP / IP, UDP. У цьому випадку він дозволить більш оптимально використовувати вільні ресурси шлюзу, що працює, як правило, на Linux.

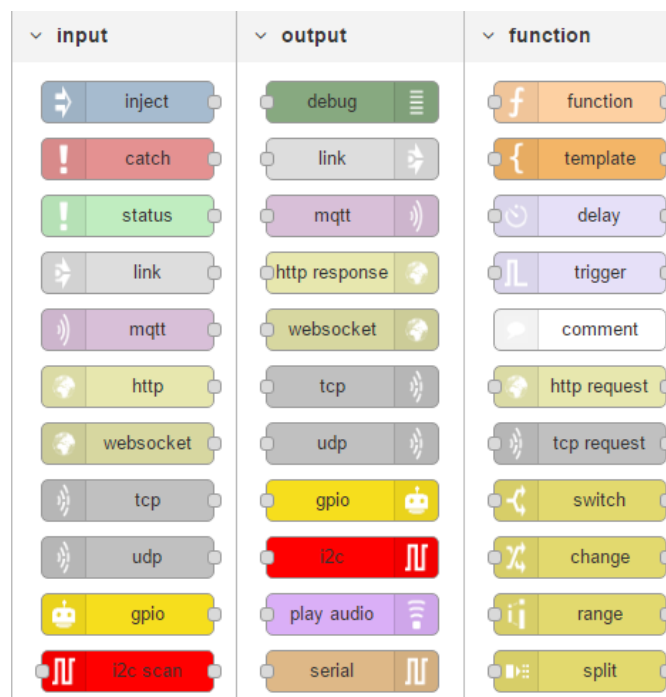
Після запуску сервера графічний інтерфейс буде доступний по <http://localhost:1880/>. Приклад робочої області Node-RED наведено на малюнку 3.2.



Малюнок 3.2 - Графічний інтерфейс Node-RED

Якщо Node-RED встановлений на самому Raspberry Pi (або іншому Одноплатний комп'ютері), то при його запуску ми вже маємо необхідний нам сервер, який працює, як уже говорилося, на Node.JS. Для того, щоб мати доступ до цього сервера не тільки в локальній мережі, але з глобальної, можна звертатися до Raspberry Pi по статичному IP. Але наша теплиця може перебувати в місці, де доступ до мережі інтернет доступний тільки за допомогою мобільних пристроїв, у яких IP-адреса динамічний. Для таких випадків існують «провідники», як, наприклад, Weaved, який буде розглянуто трохи пізніше.

Вся система будується з вузлів (блоків), які розташовані з лівого боку. При першому запуску є досить велика кількість стандартних вузлів (частина представлена на малюнку 3.3).

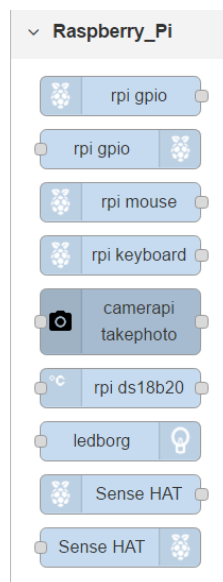


Малюнок 3.3 - Приклад доступних вузлів

Для нас особливе зручність представляють вузли, створені спеціально для роботи з Raspberry Pi (малюнок 3.4), які полегшують взаємодію, наприклад, з GPIO, що дозволить посилати сигнали на реле для включення / вимикання пристроїв. Для використання вузлів їх необхідно перенести в центральну частину, яка називається flow. Можна створювати необхідну

кількість цих робочих областей, називати їх необхідними іменами, а також зберігати в бібліотеку або на SD-карту. Всі інформація представлена у вигляді формату JSON. Справа знаходиться інформаційний блок, в якому можна знайти описи вузлів і налагодження.

У репозиторії, як говорилося раніше, можна знайти готові рішення для датчиків, які дають нам можливість без програмного коду зчитувати з них свідчення або відправляти команди. Приклад такого використання представлений на малюнку 3.5, де за допомогою вузла `rpi-ds18b20` відбувається зчитування температури з датчика з використанням протоколу 1-Wire з подальшою передачею їх вузлу для виведення показань на призначений для користувача інтерфейс.



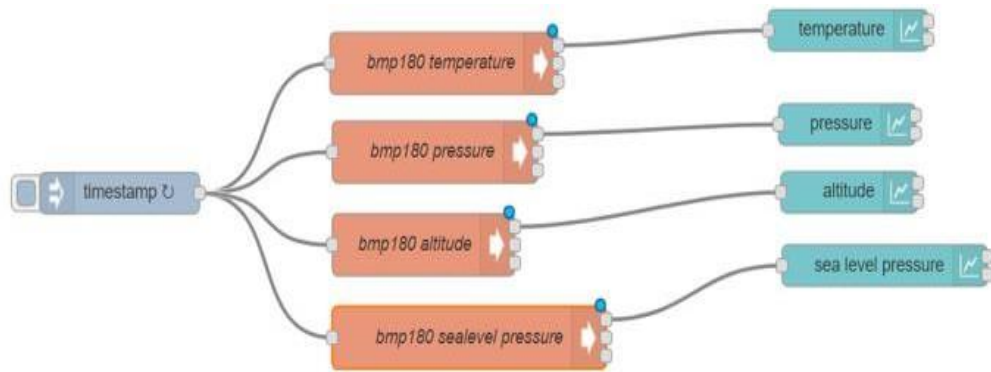
Малюнок 3.4 - Вузли для роботи з Raspberry Pi



Малюнок 3.5 - Приклад використання розробленого вузла для зчитування показань

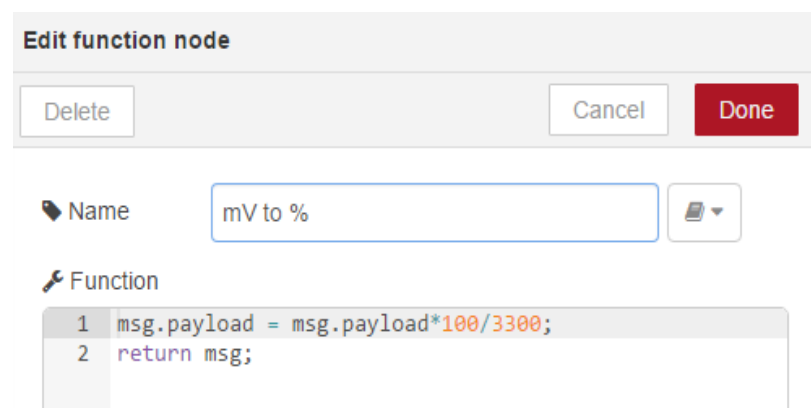
Але не для всіх датчиків існують вузли, а написання нових відбувається за допомогою мови Javascript, що в деяких випадках ускладнює завдання. Тоді зручніше використовувати вузол `exec`, які дозволяє виконувати команди терміналу Linux, що дасть нам можливість

використовувати раніше написані скрипти на мові Python для зчитування показань датчиків. Приклад використання таких вузлів наведений на малюнку 3.7.

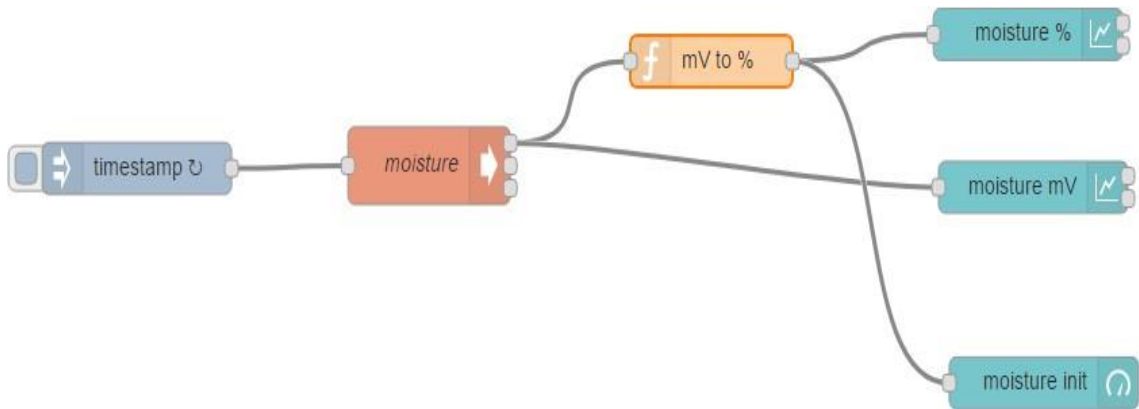


Малюнок 3.7 - Приклад використання вузла для запуску скрипта Python

Для деяких модифікацій зручно використовувати функціональні вузли, в яких можна писати код на мові Javascript (малюнок 3.8). У нашому випадку ми використовували це, щоб виводити значення з аналогового датчика Moisture в двох видах (показник напруги з датчика і перекладене значення в відсотки) без зміни коду скрипта на мові Python (малюнок 3.9).

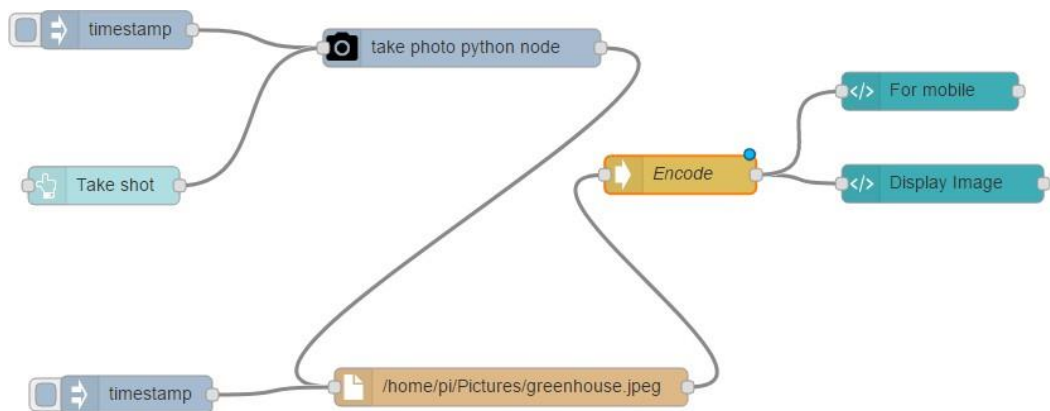


Малюнок 3.8 - Приклад настройки функціонального вузла



Малюнок 3.9 - Приклад використання функціонального вузла

Для збереження знімків з камери був використаний спеціальний вузол, який крім створення файлів різних форматів і дозволів може зберігати дані з камери в буфер. Далі збережений файл переводиться в формат base64 для подальшого виведення на призначений для користувача інтерфейс (малюнок 3.10).



Малюнок 3.10 - Збереження знімків з камери з подальшим переведенням

в формат base64

3.2 Створення алгоритмів автоматизації теплиці

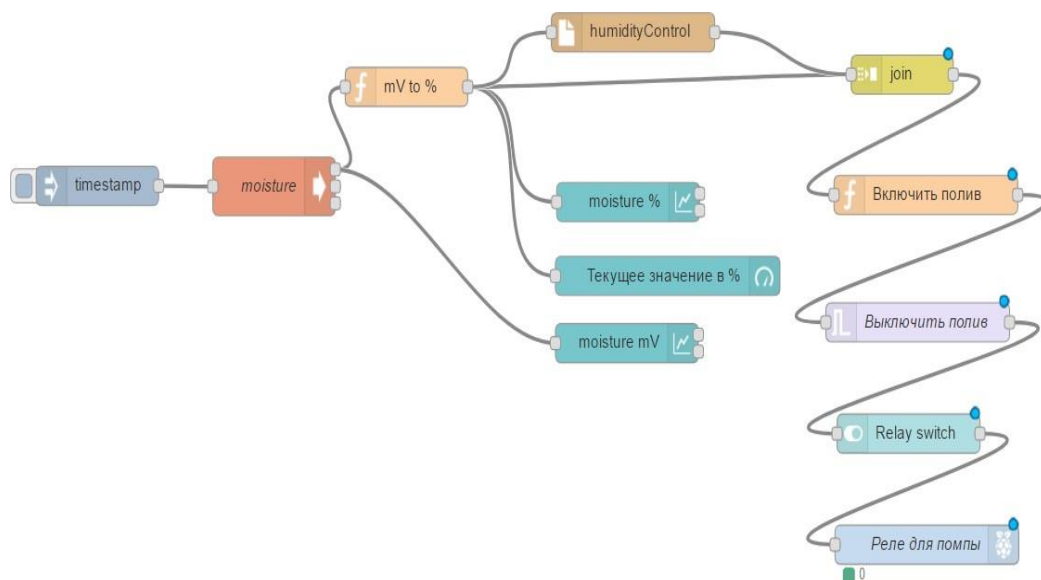
Після об'єднання наших пристроїв в єдину систему, необхідно реалізувати алгоритми для автоматизації створення мікроклімату. Як говорилося в попередньому розділі, ці алгоритми повинні підбиратися індивідуально для конкретних завдань. Ми розглянемо два з основних

реалізованих алгоритмів для нашого макета - автоматичний полив і регулювання освітлення.

Автоматичний полив в нашому випадку проводиться в тому випадку, якщо показання з датчика вологості ґрунту виявляються менше контрольного значення, введеного користувачем на сторінці з інтерфейсом (більш детальна реалізація призначеного для користувача інтерфейсу розглянута в главі 3.3). Показання датчика зчитуються раз в 30 секунд за допомогою скрипта на мові Python, потім за допомогою функціонального вузла «mV to%» значення перекладається у відсотки, що характеризують вологість ґрунту. За допомогою вузла

«Join» відбувається об'єднання показань з датчика і контрольного значення, яке читається з файлу за допомогою вузла «humidityControl». Якщо виявляється, що значення менше необхідного (за допомогою функціонального вузла «Включити полив»), відбувається перемикання реле для поливу за допомогою водяної помпи. Через 5 секунд за допомогою вузла-тригера

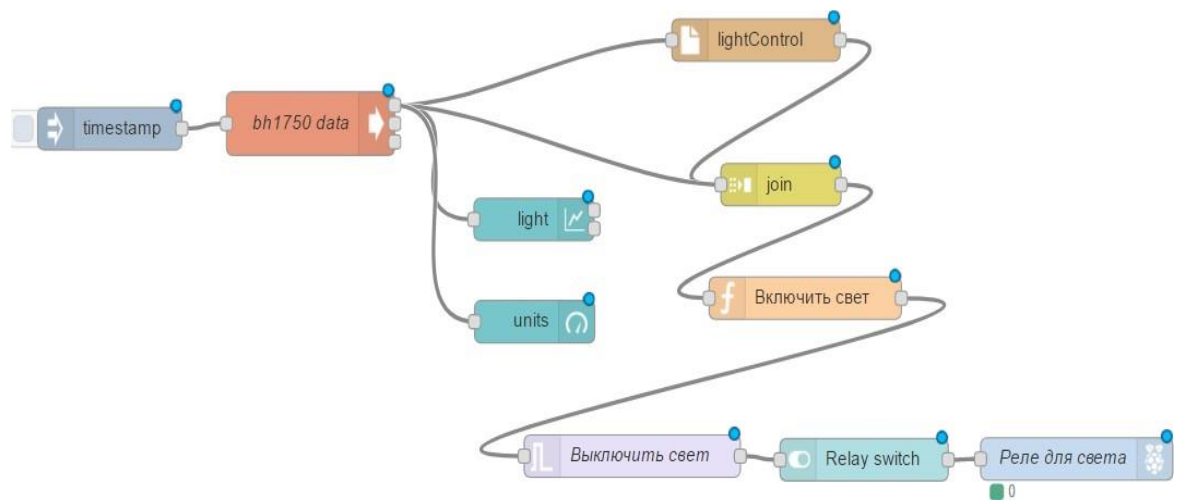
«Вимкнути полив» відбувається перемикання реле в початковий режим. Було вирішено відмовитися від варіанту з роботою помпи до того моменту, поки показання датчика не перевищать контрольне значення, тому що для того, щоб це сталося, вода повинна вбратися в ґрунт. Тим більше наш варіант економить витрачається кількість води. На малюнку 3.11 представлена схема управління автоматичним поливом. На цій схемі присутні також вузли для побудови графіків і відстежуємо стану реле на призначеному для користувача інтерфейсі, які будуть розглянуті далі.



Малюнок 3.11 - Схема управління автоматичним поливом

Алгоритм управління освітленням (малюнок 3.12) дуже схожий на алгоритм автоматичного поливу, за тим лише винятком, що зняття показань відбувається набагато рідше, а перед зняттям нових даних світло вимикається для того, щоб додаткове освітлення не використовувалося, якщо природного світла буде вистачати. Для додаткового освітлення в макеті використовувалася світлодіодна стрічка, яка також підключалася до модуля реле.

Як видно, схеми досить прості і наочні, легко піддаються модифікаціям під конкретні потреби, при цьому не потрібні якісь особливі знання мов програмування.



Малюнок 3.12 - Схема управління освітленням в теплиці

Алгоритми для управління температурою і вологістю повітря (або ж будь-які інші в рамках нашого завдання) реалізуються схожим способом.

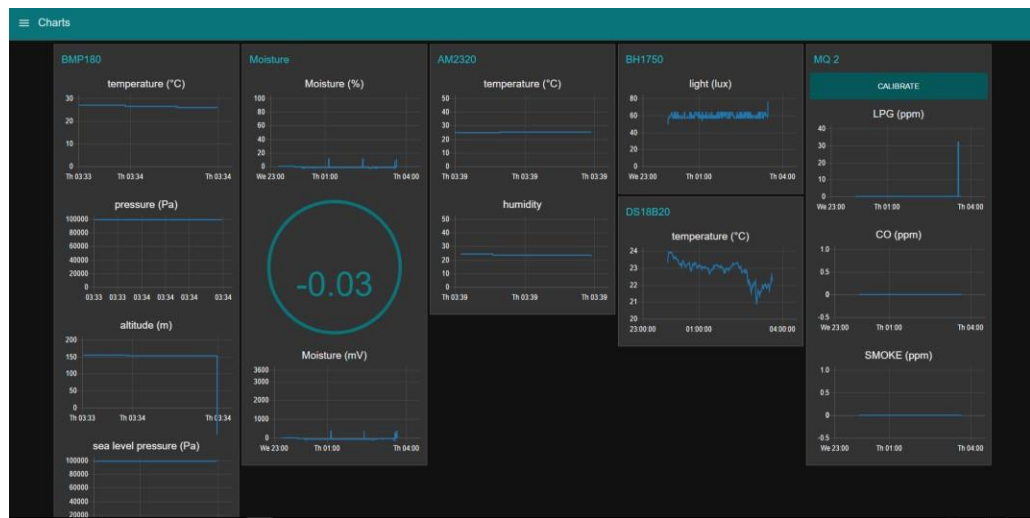
3.3 Розробка користувацького інтерфейсу

Node-RED надає спеціальні вузли для розробки користувацького інтерфейсу, який доступний, якщо до URL в браузері додати `"/ ui"`. Доступні вузли представлені на малюнку 3.13. Використання цих вузлів можна побачити на малюнках в попередньому пункті.

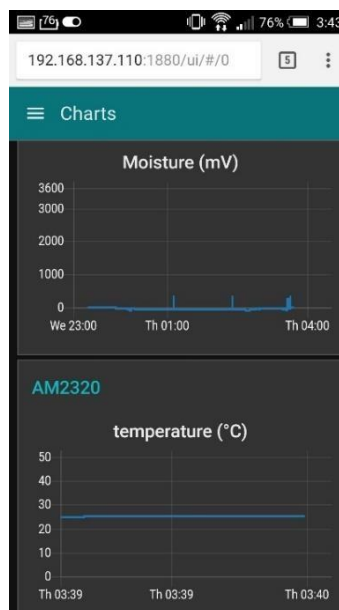


Малюнок 3.13 - Вузли для настройки призначеного для користувача інтерфейсу

Для виведення показників датчиків зручно використовувати графіки та індикатори, які будуються за допомогою вузлів chart і gauge відповідно. Ці вузли мають багато налаштувань для визначення зовнішнього вигляду, також таких характеристик, як обмеження полів, легенди і т.д. Також на цій сторінці знаходиться кнопка «CALIBRATE» для датчика MQ-2, яку можна позначити за допомогою вузла button. Вигляд сторінки з графіками показаний на малюнку 3.14.

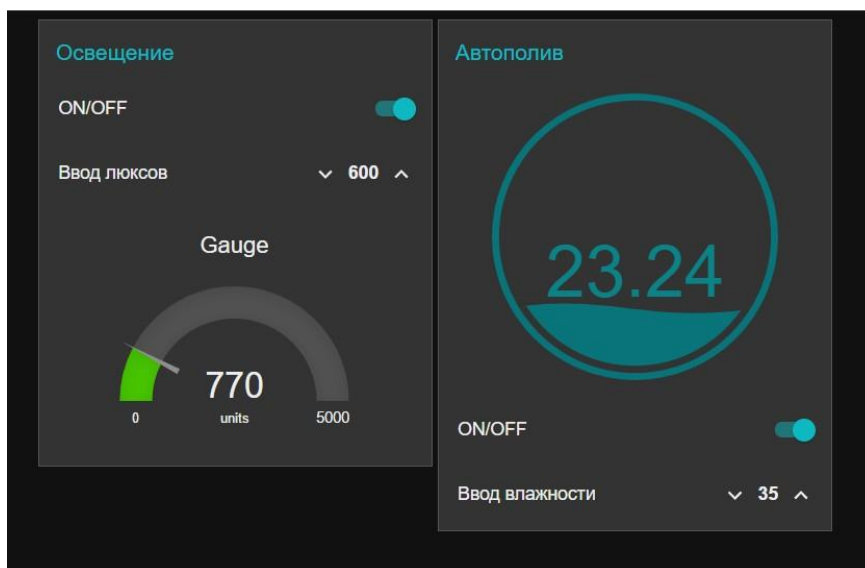


Малюнок 3.14 - Сторінка з графіками призначеного для користувача інтерфейсу. Інтерфейс має адаптивний дизайн, що зручно при зміні масштабу сторінки або зміні пристрою виведення. Приклад, як ця сторінка виглядає на екрані смартфона, показаний на малюнку 3.15.



Малюнок 3.15 - Вид призначеного для користувача інтерфейсу на екрані

смартфона



Малюнок 3.17 - Сторінка з перемикачами реле

Для включення / вимикання пристроїв, які керують освітленням, потоком води, циркулювання повітря і т.д., потрібні значення показників, які будуть використовуватися в алгоритмах управління. Ці значення налаштовуються на цій же сторінці, а далі автоматично зберігаються у відповідних файли. Таким чином, можна налаштувати систему для створення певних кліматичних умов без залучення програміста для зміни програмного коду, спираючись тільки на знання фахівців в галузі сільського господарства.

3.4 Віддалений доступ до Raspberry Pi за допомогою Weaved

Для організації віддаленого управління через глобальну мережу Інтернет необхідно мати IP адресу. Як правило, провайдери дають динамічний адресу, який періодично змінюється. Статичний адреса надається не всіма провайдерами і, найчастіше, ця послуга є платною.

Офіційний сайт Raspberry Pi [17] дає деякі поради по налаштуванню віддаленого доступу. Один із способів - налаштувати перенаправлення портів на маршрутизаторі. Для цього ви повинні змінити конфігурацію свого маршрутизатора для пересилання всього вхідного трафіку з Інтернету на

конкретному порту на локальний IP-адресу вашого Raspberry Pi. Більшість маршрутизаторів мають цю функцію. Одним з недоліків переадресації портів є те, що він надає мережевий порт у вашій приватній локальній мережі в загальнодоступний Інтернет. Це відома вразливість безпеки, і її необхідно ретельно контролювати.

Замість того щоб використовувати переадресацію портів, існує цілий ряд альтернативних інтернет-послуг. Всі вони є Dynamic DNS - спеціалізовані сервіси, які застосовуються для «прив'язки» динамічного IP адреси до інших мережних пристроїв. Принцип їх роботи полягає в наступному - ви входите на такому сервісі і отримуєте унікальну адресу для доступу до мережевого пристрою і прописуєте параметри з'єднання в цьому пристрої. При віддаленому виклику, незалежно від того, який в даний момент ваш поточний зовнішній IP, буде відбуватися переадресація на локальний IP вашого пристрою (Raspberry Pi, роутера, або ін. Пристроїв). Перевірений доступ з віддаленого комп'ютера мережі Інтернет до Raspberry Pi с встановленим Node-RED був виконаний за допомогою безкоштовного сервісу Weaved IoT [18], який також може підтримувати зв'язок через:

- SSH - дозволяє увійти в Raspberry Pi з будь-якої точки світу з SSH;
- Web (http) on port 80 - можна переглядати web - сторінки з будь-якої точки світу, розміщені на Raspberry Pi.

New Service Plans Available

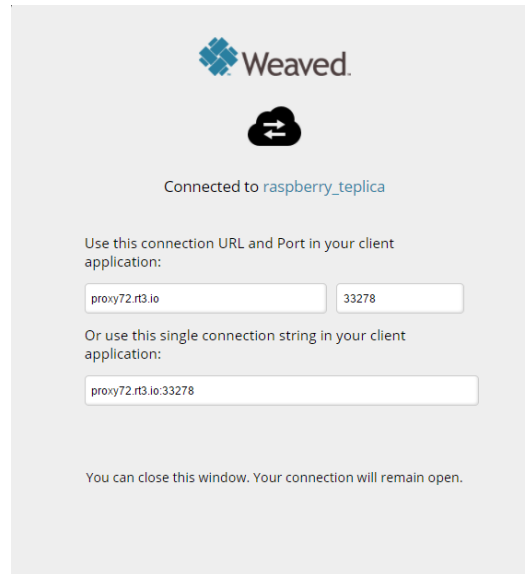
Go to [My Account](#) to choose a new service plan.

Your current list of services

Click on service names to connect. Your account allows for 10 registered services and 30 minute connections on up to 1 concurrent service(s). You can [Upgrade Now](#) to get more services, longer connection times and more concurrent connections.

raspberrypi_teplica	Generic TCP	online	Share Settings
teplica	Generic TCP	offline	Share Settings

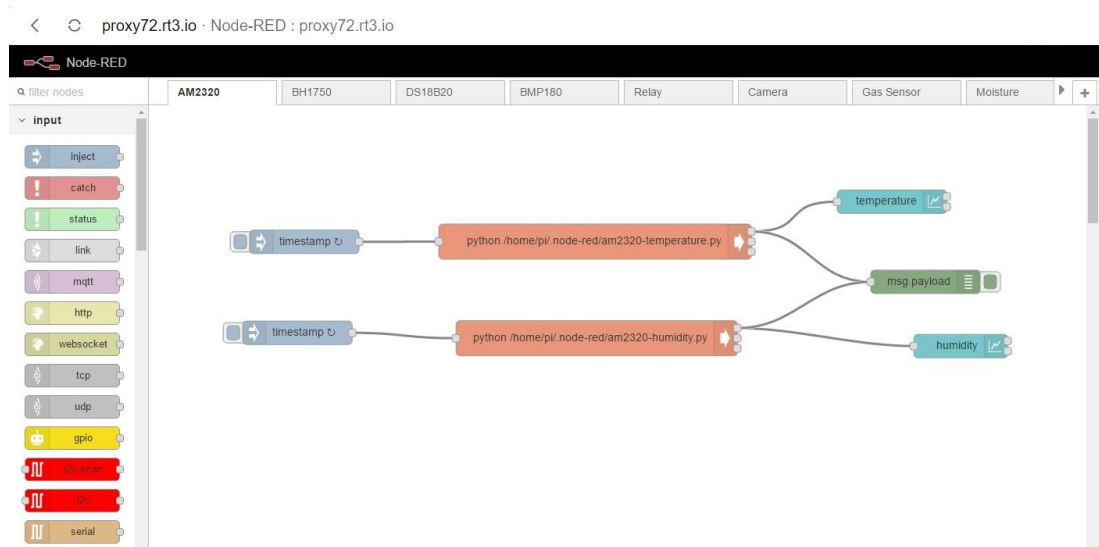
Малюнок 3.18 - Лістинг створених сервісів Weaved



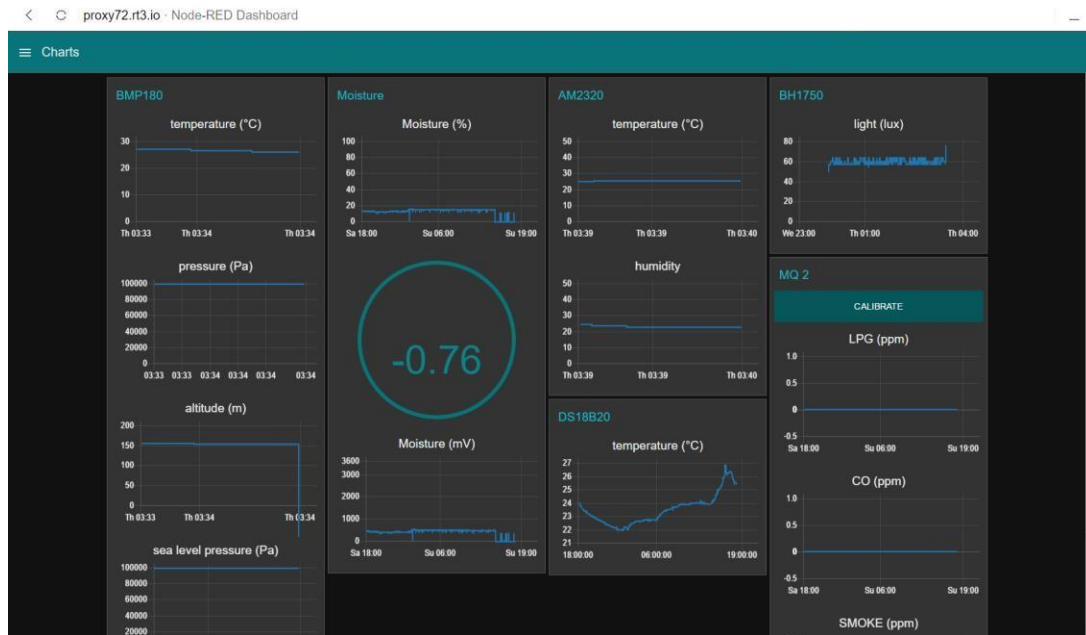
Малюнок 3.19 - Створений адреса для віддаленого доступу

Після створення облікового запису, установки і настройки Weaved, в особистому кабінеті будуть доступні наші пристрої (малюнок 3.18). Після вибору сервіс генерує URL (малюнок 3.19) для віддаленого доступу до Node-RED, що працює на нашому Raspberry Pi:

При введенні цієї адреси в рядок браузера будь-якого комп'ютера (або мобільного пристрою) з доступом до мережі інтернет відкриється сторінка з нашим проектом (малюнок 3.20). Якщо додати до URL «/ ui», то відкриється до розробленого призначеним для користувача інтерфейсом (малюнок 3.21).



Малюнок 3.20 - Віддалений доступ до інтерфейсу Node-RED



Малюнок 3.21 - Віддалений доступ до призначеного для користувача інтерфейсу

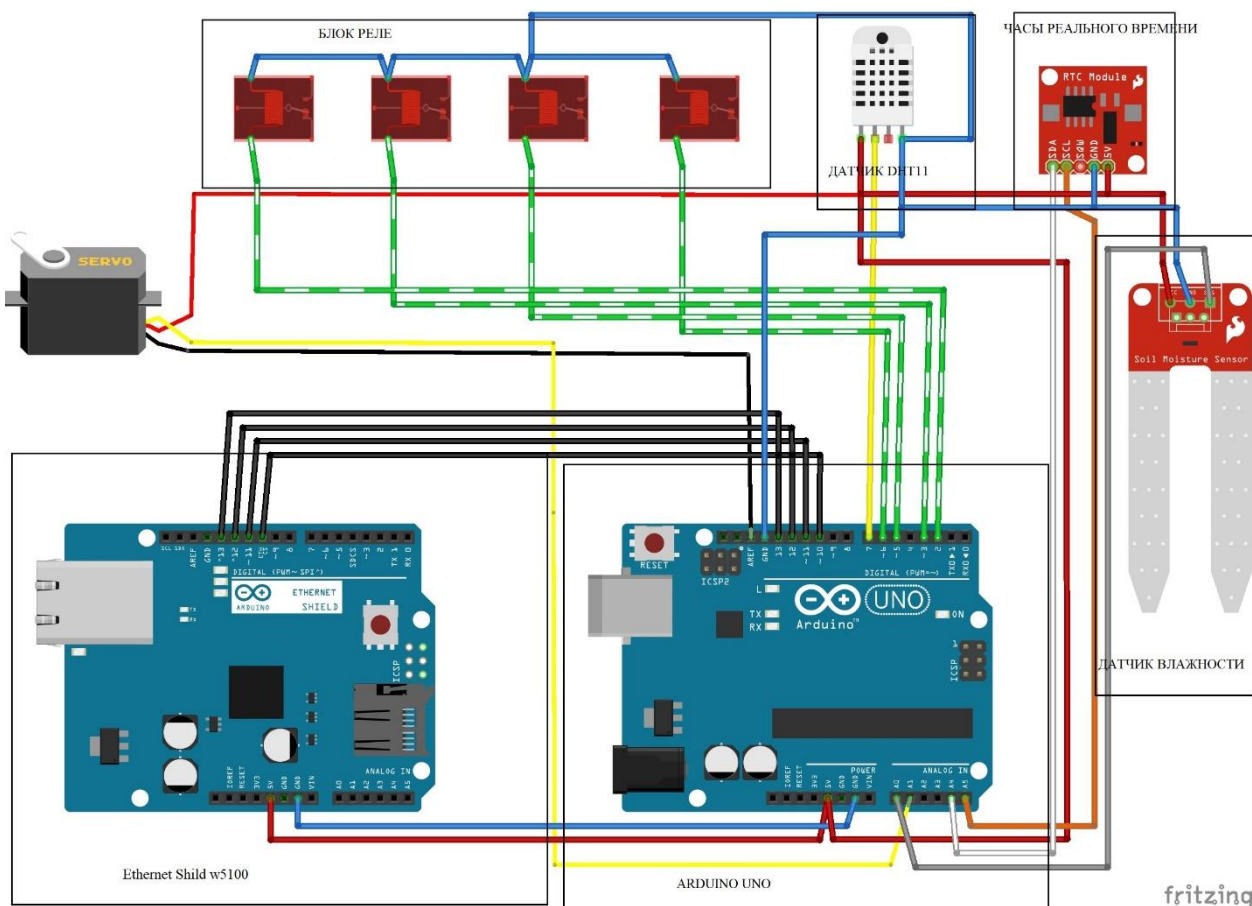
3.5 Опис схеми і алгоритму роботи

Підключення датчиків станься з допомогою дроту перемичок (тато-тато), все датчики живляться від 5 В.

Підключення датчиків до плати Arduino:

1. Ethernet Shild w5100 підключон до цифрових пінам (10,11,12,13 і 4 пін служить для зв'язку з SD картою)
2. Блок реле підключається до цифрових пінам (2,3,4,5)
3. Датчик температури DHT11 підключається до цифрового піну (7)
4. Датчик воложності ґрунту подлючається до аналогового піну (a0)
5. Серво привід для імітіції відкриття кватирки, підключається до аналогового піну (a1)
6. Годинник реального часу підключаються по I2C інтерфейсу до пінам SDA і SCL.

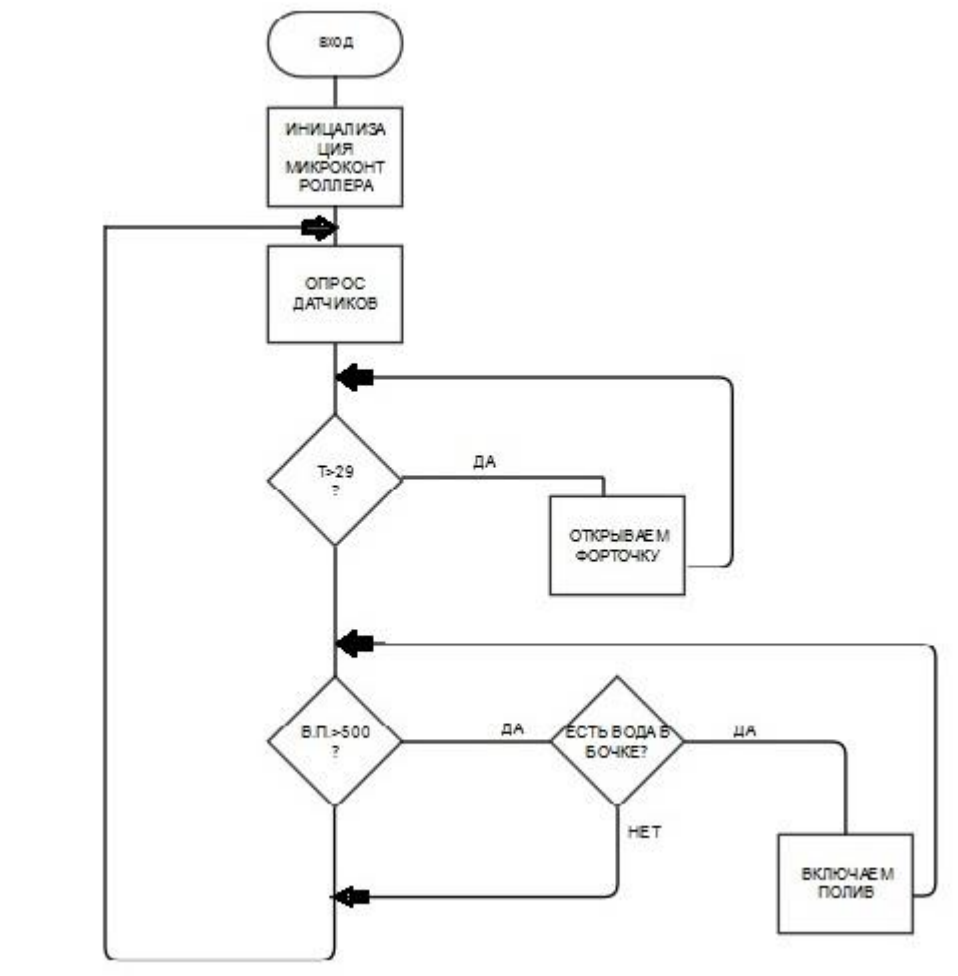
Опис схеми: датчик DHT11 підключений за стандартною схемою - живлення датчика + 5В і Gnd, провід data підключений до 7-му піну і підтягнутий до харчування (+ 5в) резистором 4,7к, пін на Arduino налаштований на вхід (на прийом даних). датчик вологості (ємнісний) підключений до аналогового піну A0 і працює від напірного напруги Ардуіно що дорівнює 1.1в.Реле спрацьовую при подачі 1 на цифровий порт, порт налаштований на вихід.



Малюнок 3.22 - Схема підключення датчиків

Алгоритм програми

При включенні живлення відбувається ініціалізація мікроконтролера потім йде зчитування датчиків вологості ґрунту і датчика температури DHT11, якщо температура більше 29 C то срабатует серво мотор і кватирка відкривається для провітрювання, якщо температура впала нижче 28 C то кватирка закривається (встановлений гістерезис на 1C для того що б не було помилкових спрацьовувань), якщо спрацював датчик вологості (значення підбираються під час налагодження) включається насос поливу, полив відбувається до заданого значення датчика вологості.



Малюнок 3.23 - Блок-схем

4. Матеріальні витрати

Для розробки системи автоматизації теплиці необхідно закупити комплектуючі, на основі яких ця система буде створена. У вартість входить тільки сама плата Raspberry Pi і датчики для моніторингу показників, тому що управляє обладнання купується і настраюється окремо для потреб клієнта. Основні компоненти із закупівельними цінами на них представлені в таблиці 4.1:

Таблиця 4.1 - Матеріальні витрати

Найменування	Ціна грн ../ шт.
Raspberry Pi 3 model B	1900
AM2320	74,60

BH1750	32
DS18B20	30
MQ-2 Gas Sensor	30
Moisture Sensor	28
ADS1115	100
Raspberry Pi Camera Board	959
SD карта 8 Гб	190
Інші витратні матеріали	200
Разом	3543,60

ВИСНОВОК

У дипломній роботі розглянуто застосування IoT- концепції в сільському господарстві на прикладі автоматизації управління і моніторингу показників датчиків на макеті теплиці.

В першу чергу був проведений огляд і вибір апаратної бази для подальшої розробки системи автоматизації теплиці з можливістю моніторингу та управління через мережу Інтернет. В ході даного огляду були розглянуті переваги і недоліки таких рішень як мікроконтролери, Arduino і одноплатні комп'ютери, після чого був зроблений вибір на користь останніх, через отримання відразу «все в одному»: базу для підключення датчиків, продуктивну обчислювальну систему, сервер з ОС Linux, доступ через Інтернет у разі неможливості використання реальних IP-адрес і доменних імен.

Після вибору апаратної бази були описані використовувані датчики і інше периферійне устаткування для моніторингу показників теплиці, а також варіанти алгоритмізації створення мікроклімату на основі цих показників.

Ключовим результатом випускної кваліфікаційної роботи є побудова автоматизованої системи з веб-інтерфейсом за допомогою графічного конфігуратора Node-RED, який об'єднав раніше написані програми на мові Python для зчитування показань з датчиків, дозволив реалізувати наочні алгоритми створення мікроклімату, а також використовувався для побудови веб-додатки з графіками і інтерфейсом для моніторингу та управління обладнанням в режимі online.

Надалі можлива доопрацювання та удосконалення алгоритмів

створення мікроклімату, використання спеціальних вузлів Node-RED для оповіщення користувача за допомогою відправки повідомлень в соціальні мережі і за допомогою SMS, а також апробація системи не на макеті, а на реальних сільськогосподарських об'єктах.

Список використаних джерел

1. Інтернет речей готується до землі. Новини Інтернету речей - <https://iot.ru/selskoe-khozyaystvo/internet-veshchey-gotovitsya-k-zemle>
2. Harvest automation, MOBILE ROBOTS FOR INDUSTRIAL PRODUCTIVITY - <https://www.public.harvestai.com/>
3. Postscapes, Smart Greenhouse Remote Monitoring Systems - <https://www.postscapes.com/wireless-open-source-hydroponics-harvestgeek/>
4. Wikipedia, Microcontroller - <https://en.wikipedia.org/wiki/Microcontroller>
5. What is an Arduino? - <https://www.arduino.cc/en/Guide/Introduction>
6. Wikipedia, Single-board computer - https://en.wikipedia.org/wiki/Single-board_computer
7. Raspberry Pi foundation, офіційний сайт Raspberry Pi - <https://www.raspberrypi.org>
8. AOSONG, AM2320 Product Manual - <http://akizukidenshi.com/download/ds/aosong/AM2320.pdf>
9. ROHM Semiconductor, BH1750 Digital 16bit Serial Output Type Ambient Light Sensor IC - http://cpre.kmutnb.ac.th/esl/learning/bh1750-light-sensor/bh1750fvi-e_datasheet.pdf
10. Bosch Sensortec, BMP180 Digital pressure sensor - <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>
11. Maxim Integrated Products, DS18B20 Programmable Resolution 1-Wire Digital Thermometer - <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
12. HANWEI ELETRONICS CO., TECHNICAL DATA MQ-2 GAS SENSOR - <http://www.mouser.com/ds/2/321/605-00008-MQ-2-Datasheet-370464.pdf>

13. Seeed Studio, Moisture Sensor User Manual -
http://www.mouser.com/ds/2/744/Seeed_101020008-838655.pdf
14. Texas Instruments Incorporated, Ultra-Small, Low-Power, 16-Bit Analog-to-Digital Converter with Internal Reference -
<http://www.ti.com/product/ADS1115>
15. Raspberry Pi foundation, CAMERA MODULE -
<https://www.raspberrypi.org/products/camera-module-v2/>
16. JS Foundation, Running on Raspberry Pi -
<https://nodered.org/docs/hardware/raspberrypi>
17. Raspberry Pi foundation, ACCESS YOUR RASPBERRY PI OVER THE INTERNET -
<https://www.raspberrypi.org/documentation/remote-access/access-over-Internet/>
18. Weaved, IoT Kit for the Raspberry Pi - <https://www.weaved.com/raspberrypi-remote-connection/>
19. Завлін, П.Н. Оцінка ефективності інновацій / П.М. Завлін, А.В. Васильєв. - СПб .: Видавничий дім «Бізнес-преса», 1998. - с. 216.
20. Беренс, В. Керівництво з підготовки промислових техніко-економічних досліджень. / В. Беренс, П. М Хавранек; СПб .: Інтерексперт, 1995. - 205с.

Додаток А

Вихідні тексти програм

В даному додатку наведено код програм для MQ-2 Gas Sensor і AM2320.

1) MQ-2 Gas Sensor:

```
impo
rt
time
impo
rt
math
impo
rt
sys
import Adafruit_ADS1x15

# Class for work with
ADS1115 class ADS ():

##### Hardware Related Macros #####
###
    PIN = 0          # Define which analog input channel you are going
                    # to use RL_VALUE = 5          # Define the load resistance on the
                    # board, in kilo ohms RO_CLEAN_AIR_FACTOR = 9.83
# RO_CLEAN_AIR_FACTOR = (Sensor resistance in clean
air) / RO, # which is derived from the chart in
datasheet

# Choose a gain of 2/3 for reading voltages from 0 to 6.144V.
# Or pick a different gain to change the range of voltages that
are read: # - 2/3 = +/- 6.144V
# - 1 = +/-
4.096V # - 2 =
+/- 2.048V # -
4 = +/- 1.024V
# - 8 = +/-
0.512V # - 16
= +/- 0.256V
# See table 3 in the ADS1015 / ADS1115 datasheet for more info on gain.
GAIN = 2/3

##### Software Related Macros #####
###
    CALIBARAION_SAMPLE_TIMES = 50
# Define how many samples you are going to take in the
calibration phase CALIBRATION_SAMPLE_INTERVAL = 500
# Define the time interal (in milisecond) between each
samples in the # cablibration phase
```

```

    READ_SAMPLE_INTERVAL = 50
# Define how many samples you are going to take in normal
  operation READ_SAMPLE_TIMES = 5
# Define the time interval (in milisecond) between each
samples in # normal operation

##### Application Related Macros
##### GAS_LPG      = 0
GAS_CO          = 1
GAS_SMOKE       = 2

def init (self, Ro = 10,
  analogPin = 0): try:
  f = open (
    'mq2R0.txt')
  self.Ro = float
    (f.read ())
except:
  self.Ro = 10;

  self.PIN = analogPin
  self.adc = Adafruit_ADS1x15.ADS1115 ()

  self.LPGCurve = [2.3,0.21, -0.47]
#two points are taken from the curve.

#with these two points, a line is formed which is "approximately
equivalent" #to the original curve.

#data format: {x, y, slope}; point1: (lg200, 0.21), point2: (lg10000,
  -0.59) self.COCurve = [2.3,0.72, -0.34]
#two points are taken from the curve.

#with these two points, a line is formed which is "approximately
equivalent" #to the original curve.

#data format: [x, y, slope]; point1: (lg200, 0.72), point2:
  (lg10000,
  ) self.SmokeCurve = [2.3,0.53, -0.44]
#two points are taken from the curve.

#with these two points, a line is formed which is "approximately
equivalent" #to the original curve.

#data format: [x, y, slope]; point1: (lg200, 0.53), point2: (lg10000, -
  0.22)

  #print ( "Calibrating ...")
  # Self.Ro = self.Calibration
  (self.PIN) #print (
  "Calibration is done ... \n")
  #print ( "Ro =% f kohm"%
  self.Ro)

def Percentage
  (self):
  val = {}
  read = self.Read (self.PIN)
  val [ "GAS_LPG"] = self.GetGasPercentage (read / self.Ro,
  self.GAS_LPG) val [ "CO"] = self.GetGasPercentage (read /

```

```

        self.Ro, self.GAS_CO) val [ "SMOKE"] = self.GetGasPercentage
        (read / self.Ro, self.GAS_SMOKE) return val

##### ResistanceCalculation #####
#
#Input:   raw_adc - raw value read from adc, which represents the
voltage #Output:  the calculated sensor resistance
#Remarks: The sensor and the load resistor forms a voltage
divider. Given #the voltage across the load resistor and its
resistance, the resistance # of the sensor could be derived.

#####
#####
def ResistanceCalculation (self, raw_adc):
    return float (self.RL_VALUE * (32767.0-raw_adc) / float (raw_adc));

##### Calibration #####
#####
# Input:   pin - analog
channel # Output: Ro of
the sensor
# Remarks: This function assumes that the sensor is in clean air.
It use # ResistanceCalculation to calculates the sensor resistance
in clean air # and then divides it with RO_CLEAN_AIR_FACTOR.
RO_CLEAN_AIR_FACTOR
# Is about 10, which differs slightly between different sensors.
#####
#####
def Calibration
    (self, pin): val
    = 0.0
    for i in range
    (self.CALIBARAION_SAMPLE_TIMES): # take
multiple samples
        val += self.ResistanceCalculation
(self.adc.read_adc (pin, gain = self.GAIN))
        time.sleep (self.CALIBRATION_SAMPLE_INTERVAL / 1000.0)

    val = val /
self.CALIBARAION_SAMPLE_TIMES #
calculate the average value

    val = val / self.RO_CLEAN_AIR_FACTOR
# Divided by RO_CLEAN_AIR_FACTOR yields
the Ro # according to the chart in the
datasheet

    return val;

##### Read
#####
# Input:   pin - analog
channel # Output: Rs of
the sensor
# Remarks: This function use ResistanceCalculation to caculate the
sensor # resistance (Rs).
# The Rs changes as the sensor is in the different consentration
of the # target gas. The sample times and the time interval between
samples # could be configured by changing the definition of the
macros.

```

```

#####
#####
def Read
    (self,
    pin): rs
    = 0.0

    for i in range (self.READ_SAMPLE_TIMES):
        rs += self.ResistanceCalculation
(self.adc.read_adc (pin, gain = self.GAIN))
        time.sleep (self.READ_SAMPLE_INTERVAL

    / 1000.0)rs = rs /

    self.READ_SAMPLE_TIMES

    return rs

##### GetGasPercentage
#####
# Input:  rs_ro_ratio - Rs
divided by Ro #      gas_id -
target gas type # Output:  ppm
of the target gas
# Remarks: This function passes different curves to the GetPercentage
# Function which calculates the ppm (parts per million) of the
target # gas.

#####
#####
def GetGasPercentage (self, rs_ro_ratio,
    gas_id): if (gas_id == self.GAS_LPG):
        return self.GetPercentage (rs_ro_ratio,
self.LPGCurve) elif (gas_id == self.GAS_CO):
        return self.GetPercentage (rs_ro_ratio,
self.COCurve) elif (gas_id == self.GAS_SMOKE):
        return self.GetPercentage (rs_ro_ratio,
self.SmokeCurve) return 0

##### GetPercentage
#####
# Input:  rs_ro_ratio - Rs divided by Ro
#      pcurve - pointer to the curve of the target gas
# Output: ppm of the target gas
# Remarks: By using the slope and a point of the line. The x
(logarithmic #value of ppm)
#of the line could be derived if y (rs_ro_ratio) is provided. As
it is a #logarithmic coordinate, power of 10 is used to convert
the result to # non-logarithmic value.

#####
#####
def GetPercentage (self, rs_ro_ratio, pcurve):
    return (math.pow (10, (((math.log (rs_ro_ratio) -pcurve [1]) /
    pcurve [2])
+ Pcurve [0])))

```

2) AM2320

```

import posix
from fcntl

```

```

import ioctl
import time

class AM2320:
    I2C_ADDR =
    0x5c
    I2C_SLAVE =
    0x0703

    def init (self, i2cbus = 1):
        self._i2cbus = i2cbus

    @staticmethod
    def _calc_crc16
    (data): crc =
    0xFFFF
    for x in
    data:
        crc =
        crc ^
        x
        for bit in range (0, 8):
            if (crc & 0x0001) ==
                0x0001: crc >> = 1
                crc ^
                = 0xA001
            else:
                cr
            c >> = 1
        return
        crc

    @staticmethod
    def _combine_bytes
    (msb, lsb): return
    msb << 8 | lsb

    def readSensor (self):
        fd = posix.open ( "/ dev / i2c-% d"% self._i2cbus,

        posix.O_RDWR) ioctl (fd, self.I2C_SLAVE,

        self.I2C_ADDR)

        # Wake AM2320 up, goes to sleep to not warm up and affect the
        humidity sensor
        # This write will fail as AM2320 will not ACK
        this write try:
            posix.write (fd, b
            '\ 0x00') except:
                pass
            time.sleep (0.001) #Wait at least 0.8ms, at most 3ms

            # Write at addr 0x03, start reg = 0x00, num regs =
            0x04 * / posix.write (fd, b '\ x03 \ x00 \ x04')
            time.sleep (0.0016) #Wait at least 1.5ms

            for result # Read out 8 bytes of result

            data

            # Byte 0: Should be Modbus function code 0x03

```

```

# Byte 1: Should be number of registers to read
(0x04) # Byte 2: Humidity msb
# Byte 3: Humidity lsb
# Byte 4:
Temperature msb #
Byte 5:
Temperature lsb #
Byte 6: CRC lsb
byte
# Byte 7: CRC msb byte
data = bytearray (posix.read (fd, 8))

# Check data [0] and data [1]
if data [0]! = 0x03 or data [1]! = 0x04:
    raise Exception ( "First two read bytes are a mismatch")

# CRC check
if self._calc_crc16 (data [0: 6])! = self._combine_bytes (data
    [7], data [6]): raise Exception ( "CRC failed")

# Temperature resolution is 16Bit,
# Temperature highest bit (Bit15) is equal to 1
indicates a # negative temperature, the temperature
highest bit (Bit15) # is equal to 0 indicates a
positive temperature;
# Temperature in addition to the most significant bit (Bit14
~ Bit0) # indicates the temperature sensor string value.
# Temperature sensor value is a string of 10
times the # actual temperature value.
temp = self._combine_bytes (data [4],
data [5]) if temp & 0x8000:
    temp = - (temp &
0x7FFF) temp / =
10.0

humi = self._combine_bytes (data [2], data
[3]) / 10.0 return (temp, humi)

am2320 = AM2320 (1)
(T, h) =
am2320.readSensor ()
print t
print h

```

Додаток Б

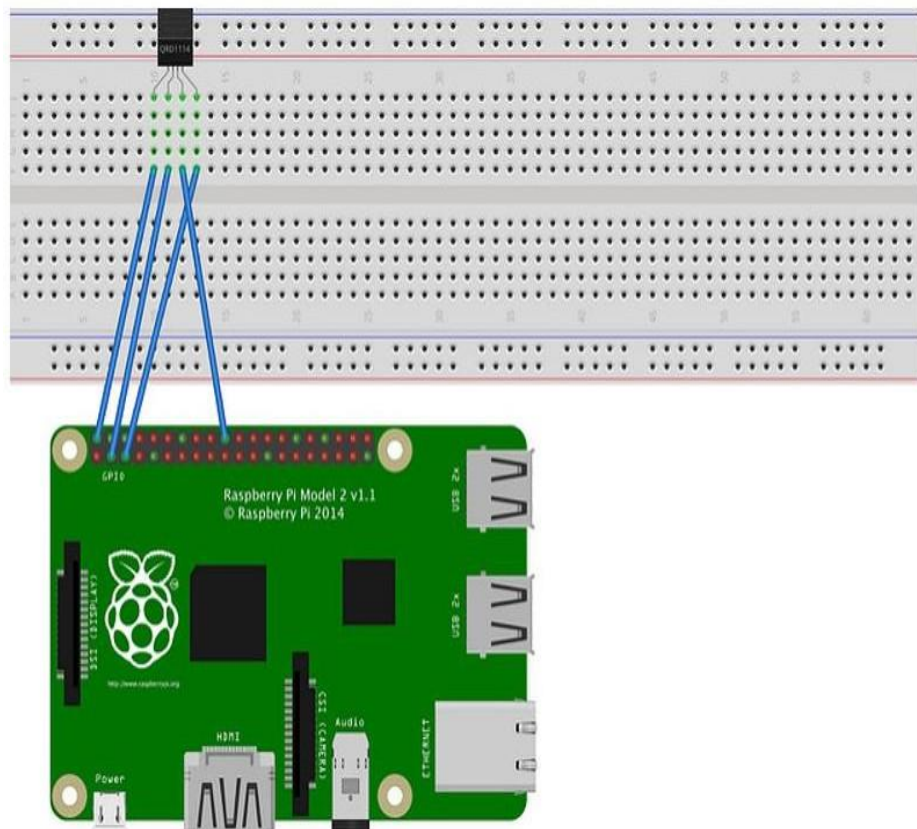
Підключення датчиків до Raspberry Pi

У цьому додатку описано підключення контактів датчиків до контактів Raspberry Pi з відповідними малюнками, на яких це наочно можна спостерігати. Всі малюнки виконані за допомогою сервісу Fritzing.

1) AM2320

Контакти датчика AM2320:

- VDD - для підключення живлення (3.3 В або 5 В)
- GND - для підключення до мінуса (GND)
- SCL і SDA - для підключення до шини I2C

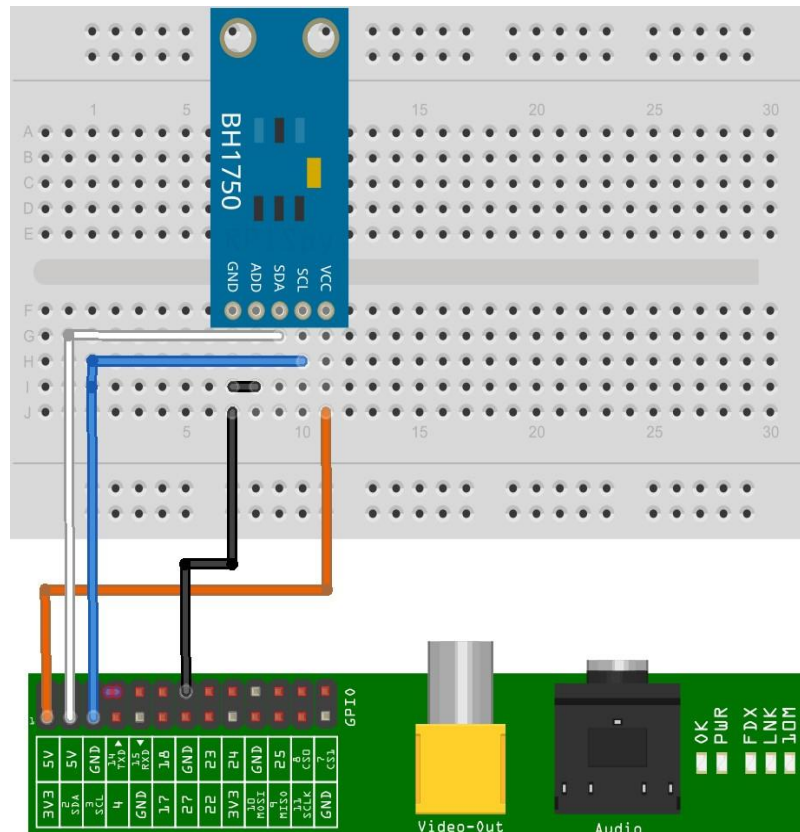


Малюнок Б.1 - Підключення AM2320 до Raspberry Pi 2)

VH1750

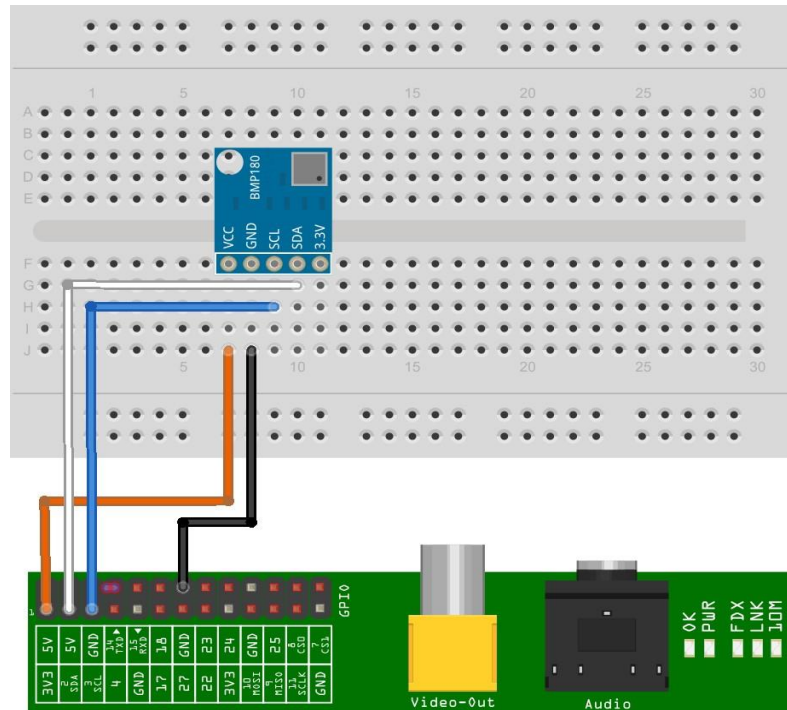
Контакти датчика VH1750:

- VCC - для підключення живлення 5 В
- GND - для підключення до мінуса (GND)
- SCL і SDA - для підключення до шини I2C



Малюнок Б.2 - Підключення VH1750 до Raspberry Pi

3) BMP180



Малюнок Б.3 - Підключення BMP180 до Raspberry Pi

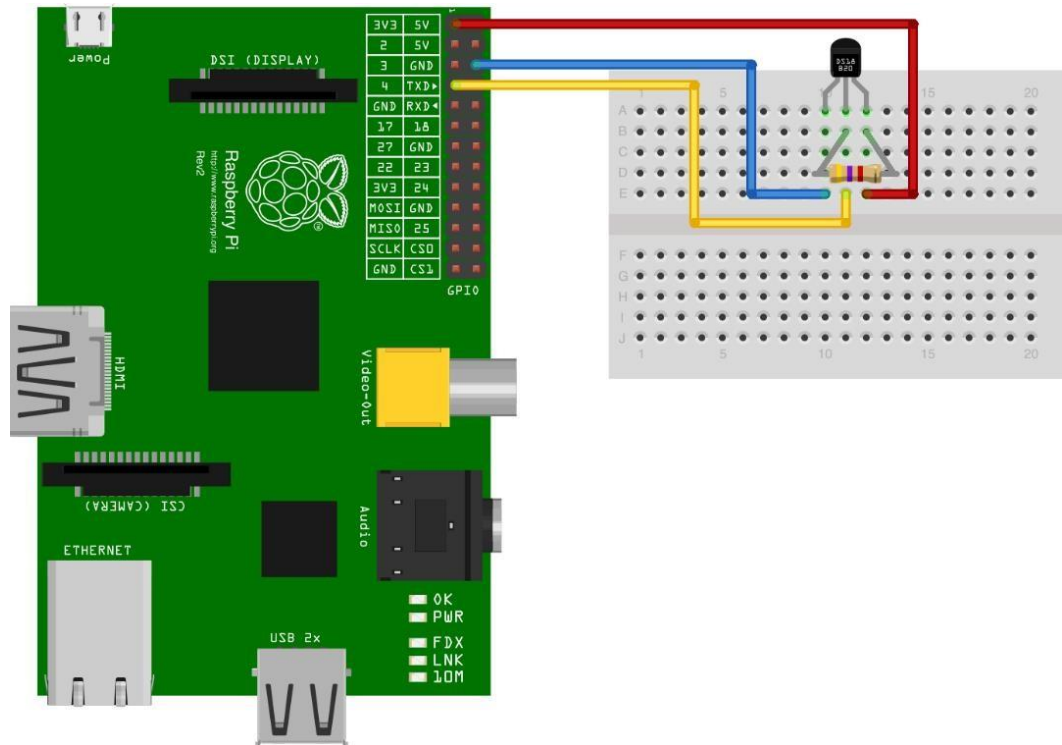
Контакти датчика BMP180:

- VCC - для підключення 5 В живлення
- GND - для підключення до мінуса (GND)
- SCL і SDA - для підключення до шини I2C
- 3.3 - для підключення 3.3 В живлення

живлення 4) BH1750

Контакти датчика BH1750:

- VDD - для підключення живлення (3,3 В або 5 В)
- GND - для підключення до мінуса (GND)
- DQ - введення / виведення даних за допомогою інтерфейсу 1-Wire



Малюнок Б.4 - Підключення DS18B20 до Raspberry Pi

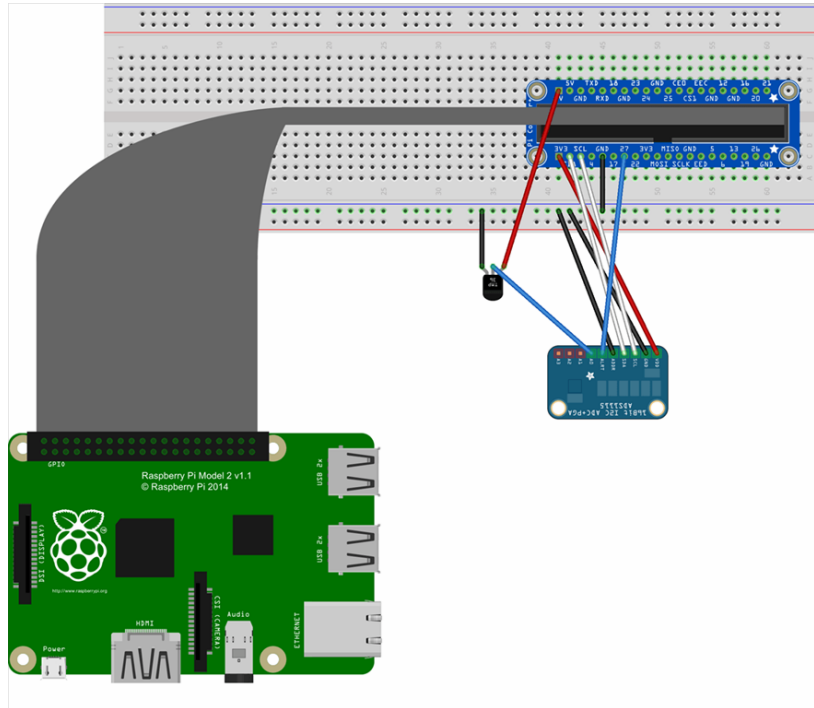
5) Аналогові датчики MQ-2 і Moisture Sensor і АЦП

ADS1115 Контакти АЦП ADS1115:

- VDD - для підключення живлення (3,3 В або 5 В)
 - GND - для підключення до мінуса (GND)
 - SCL і SDA - для підключення до шини I2C
- Контакти датчика MQ-2:
- DOUT - цифровий вихід
 - AOUT - аналоговий вихід (підключається до аналогового входу АЦП)
 - GND - для підключення до мінуса (GND)
 - VCC — для підключення к напрузі (3.3 В або 5 В)

Контакти датчика Moisture Sensor:

- AOUT – аналоговий вихід (підключається к аналоговому входу АЦП)
- GND — для підключений к мінусу (GND)
- VCC — для підключення к напрузі (5 В)



Малюнок Б.5 – Підключення MQ-2 и ADS1115 к Raspberry Pi