

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

«Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті» коммерциялық емес акционерлік қоғамы

Автоматика және ақпараттық технологиялар институты

Автоматтандыру және басқару кафедрасы

6B07103 - Автоматтандыру және роботтандыру

Серікұлы Бекарыс

Кондитерлік фабриканың технологиялық үрдістерін автоматтандыру жүйесін зерттеу

Дипломдық жобаға  
**ТҮСІНДІРМЕ ЖАЗБА**

6B07103 - Автоматтандыру және роботтандыру

Алматы 2024

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ  
«Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті» коммерциялық  
емес акционерлік қоғамы

Автоматика және ақпараттық технологиялар институты

Автоматтандыру және басқару кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ  
Автоматика және ақпараттық  
технологиялар  
институт директорының м.ф.  
PhD докторы, қауымдастырылған  
профессор

Кальпеева Ж.Б.

«04» шілде 2024 ж.



Дипломдық жобаға  
ТҮСІНДІРМЕ ЖАЗБА

Тақырыбы: «Кондитерлік фабриканың технологиялық үрдістерін автоматтандыру жүйесін зерттеу»

6B07103 - «Автоматтандыру және роботтандыру»

Орындаған:

Серікұлы Бекарыс

Рецензент:

эл-Фараби атындағы ҚазҰУ  
“Тұрақты даму бойынша ЮНЕСКО”  
кафедрасының доцент м.а, т.ғ.к

Ғылыми жетекші:

техника ғылымының магистрі  
аға оқытушы

Умбетбеков А.Т.

«5» шілде 2024 ж

Баяндина Г.С.

«5» шілде 2024 ж.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ  
«Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті» коммерциялық  
емес акционерлік қоғамы

Автоматика және ақпараттық технологиялар институты

Автоматтандыру және басқару кафедрасы

6B07103 - Автоматтандыру және роботтандыру

**БЕКІТЕМІН**

Автоматика және ақпараттық  
технологиялар

институт директорының м.ф.

PhD докторы қауымдастырылған

профессор

Кальпеева Ж.Б.

2024 ж.



Дипломдық жобаны дайындауға  
**ТАПСЫРМА**

Білім алушы Серікұлы Бекарыс

Жобаның тақырыбы: «Кондитерлік фабрианың технологиялық үрдістерін автоматтандыру жүйесін зерттеу»

Университет проректоры Б.А.Жаутиковтың «04» желтоқсан 2023 ж. № «548-П/Ө» бұйрығымен бекітілген.

Аяқталған жұмысты тапсыру мерзімі «\_\_\_» \_\_\_\_\_ 2024 ж.

Дипломдық жобаның бастапқы деректері: кондитерлік фабрианың жұмыс жасау принципі;  
Дипломдық жобада әзірлеуге жататын мәселелер тізімі:

а) Бидай астығын ұнға қайта өңдеуге арналған аппаратуралық технологиясын зерттеу;

б) Асинхронды веб-сервер 80-порт, және ESPDash тақтасы;

в) Автоматтандырудың функционалдық схемасы

Қажет материалдар тізімі (міндетті программалық кодтарын көрсете отырып):  
функционалдық сұлба, құрылымдық сұлба

Жұмыс презентациясы \_\_\_ слайдтарда көрсетілген.

Ұсынылатын негізгі әдебиеттер:

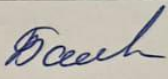

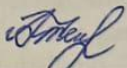
[1] Бердышев, в.ф. газдар мен суды тазартудың технологиялық процестерін автоматтандыру негіздері: дәрістер курсы / в. ф. Бердышев, К. С. Шатохин. - М.: МИСиС, 2016. - 136 с.

[2] Кангин, в. в. технологиялық процестерді автоматтандыру жүйелеріндегі өнеркәсіптік контроллерлер: оқу құралы / В. В. Кангин. - Ст. Оскол: ТНТ, 2016. - 408 с.

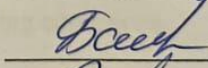
Дипломдық жобаны дайындау  
КЕСТЕСІ

Бөлімдердің атауы, зерттеп дайындалатын мәселелер тізімі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Технологиялық бөлім Кондитерлік фабриканың автоматтандырылуы	4.01.2024 ж. – 4.02.2024 ж.	орындалды
Арнайы бөлім Arduino программасымен жұмысты іске асыру	5.02.2024 ж. – 20.05.2024 ж.	орындалды

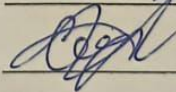
Аяқталған дипломдық жоба үшін, оған қатысты бөлімдердің жобасын көрсетумен, кеңесшілер мен норма бақылаушының қойған қолдары

Бөлімдер атауы	Кеңесшілер тегі, аты, әкесінің аты, (ғылыми дәрежесі, атағы)	Қол қойылған күні	Қолы
Технологиялық бөлім	Баяндина Г.С. техника ғылымдарының магистрі, аға оқытушы	25.05.24	
Арнайы бөлім	Баяндина Г.С. техника ғылымдарының магистрі, аға оқытушы	24.05.24	
Норма бақылаушы	Жеңіс А.Б. техника ғылымдарының магистрі, оқытушы	07.06.2024	

Ғылыми жетекшісі

 Баяндина Г.С.

Тапсырманы орындауға қабылдаған білім алушы

 Серікұлы Б.

Күні

« \_\_\_\_\_ » \_\_\_\_\_ 2024 ж.

## **АНДАТПА**

Дипломдық жобада біз кондитерлік фабриканың технологиялық процестерін автоматтандыру жүйелерін зерттеуді дамытуымыз керек. Даму кезінде біз кондитерлік фабрика туралы және кондитерлік фабриканы басқару жүйесін құрудың барлық түрлері туралы түсінік алдық. Ол үшін біз Arduino бағдарламасына негізделген wemos тақтасын алдық. Код python бағдарламалау тілінде жазылған. Біз газ датчиктерін таңдап, Arduino көмегімен осы датчиктердің жұмысын көруге болатын макет жасадық. Экономикалық пайда есептелді. Еңбек жағдайларына талдау жасады.

## **АННОТАЦИЯ**

В дипломной работе нам необходимо разработать технологические процессы кондитерской фабрики исследование систем автоматизации. При разработке получили понятие о кондитерской фабрике и о всевозможном создании системы управления для кондитерской фабрики. Для этого мы взяли плату wemos, которая программируется на основе программе Arduino. Код был написан на языке программирования python. Мы подобрали датчики газа и создали макет, на которой можно увидеть работу данных датчиков при помощи Arduino. Рассчитали экономическую выгоду. Выполнили анализ условий труда.

## **ANNOTATION**

In the thesis, we need to develop the technological processes of the confectionery factory and the study of automation systems. During the development, we got the concept of a confectionery factory and about the possible creation of a control system for a confectionery factory. To do this, we took the wemos board, which is programmed based on the Arduino program. The code was written in the python programming language. We picked up the gas sensors and created a layout on which you can see the operation of these sensors using Arduino. We calculated the economic benefits. We have performed an analysis of working conditions.

## МАЗМҰНЫ

Кіріспе.....	11
1 Аналитикалық шолу.....	8
1.1 Кондитерлік фабриканың жұмыс жасау принципі .....	8
2 Кондитерлік фабриканың автоматтандырылуы.....	16
2.1 Жалпы мәліметтер.....	16
3 Практикалық бөлім.....	20
3.1 Ардуино программасы .....	20
3.2 Жобаны іске асыру .....	23
Қорытынды.....	37
Қолданылған әдебиет тізімі.....	38
Қосымша А.....	39

## КІРІСПЕ

Дипломдық жобада біз кондитерлік фабриканың технологиялық процестерін автоматтандыру жүйелерін зерттеуді дамытуымыз керек. Даму кезінде біз кондитерлік фабрика туралы және кондитерлік фабриканы басқару жүйесін құрудың барлық түрлері туралы түсінік алдық. Ол үшін біз Arduino бағдарламасына негізделген wemos тақтасын алдық. Код python бағдарламалау тілінде жазылған. Біз газ датчиктерін таңдап, Arduino көмегімен осы датчиктердің жұмысын көруге болатын макет жасадық. Экономикалық пайда есептелді. Еңбек жағдайларына талдау жасады.

Кондитер өндірісі тамақ өндірісінің қарқынды дамыған саласы болып келеді. Қазіргі уақытта кондитерлік өнеркәсіп жоғары деңгейдегі техника, жоғары білікті мамандарды талап ететін қуатты энергетикалық шаруашылығы бар индустриялық өндіріс болып табылады. Қазақстанда кондитерлік өнімдер нарығы сан алуан деп айтуға болады, өйткені, елімізде отандық компаниялар мен әлемдік брендтер өнімі көптен ұсынылған. Сұраныс ұсыныспен толық жабылады. Шетел кондитер өнімдеріне қарағанда, еліміздің тәттілері көберек сұранысқа ие. Отандық өнімдердің ерекшелігі – кондитер өнімдеріне тек табиғи органикалық қоспаларды қоланатынында. Бұл ерекшелік шетел азаматтарының да көңілінен шығады. Бұл дипломдық жобада кондитер өнімдерінің бірі – шоколад өнімін дайындау процесі қарастырылады.

Кондитерлік өнімдерді өндірудің технологиялық процесі, жартылай фабрикаттар дайындық үшін термиялық өңдеуден өтеді және бұл кезең бүкіл технологиялық схемада ең маңызды болып табылады. Сұйық ашытқы немесе жұмсақ қысқа қамыр жасалады, кремдер, суфле, желе шайқалады. Егер қатпарлы қамыр жасау қажет болса, қамыр илейтін машина қолданылады. Кесу, қалыптау және пісіру әрі қарай, еңбек процесі кондитерлік өнімдерді кесуді және пішіндеуді қамтиды. Бұл бөлек бөлімде жасалады, онда әртүрлі қамырды кесуге арналған орындар бар. Қатпарлы және қысқа қамыр кесіліп, Тоңазытқыш үстеліне құйылады, өйткені қамырдың бұл түрлері пісіру кезінде салқындатуды қажет етеді. Қысқа, печенье, қатпарлы қамырдан жасалған қалыпталған өнімдер бірден кондитерлік парақтарға жіберіліп, пісіріледі.

## **1 Аналитикалық шолу**

### **1.1 Кондитерлік фабриканың жұмыс жасау принципі**

Кондитерлік өнімдер қандай шикізаттан жасалғанына байланысты қант пен ұн болып табылады. Технологиялық процестің сұлбасы әр жағдайда әр түрлі болады. Мармелад, карамель, шоколад, пастила, ирис, драже сияқты тағамдар қантты болып саналады, ал ұн құрамында ұн бар барлық өнімдер: вафли, печенье, пряник және т.б. Кондитерлік өнімдердің әртүрлі түрлері арнайы технологиялық схемалар бойынша шығарылады. Бірақ технологиялық процестің барлық операцияларын үш кезеңге дейін қысқартуға болады: дайындық; негізгі; қорытынды. Дайындық кезеңінде өндіріс процесін шикізатпен және қажетті компоненттермен тұрақты қамтамасыз ету маңызды рөл атқарады. Бұл кезеңде шикізат қабылданады және сақтауға дайындалады, содан кейін өндіріске дайындалады. Өндірістің негізгі кезеңінде барлық жұмыстар орындалады, оның барысында кондитерлік массалар алынады, өнімдер қалыптасады, олардың беттері өңделеді. Негізгі кезеңде оралмаған дайын өнім шығарылады. Соңғы кезеңде алынған өнім қаптамада ресімделеді. Кондитерлік цех қалай жұмыс істейді Кондитерлік цехтың жұмысы бірдей схема бойынша құрылады, ал ол тәуелсіз бөлім бола алады немесе ірі тамақ өндірісінің бөлігі бола алады. Мұнда әртүрлі аспаздық өнімдер шығарылады. Әрбір цех бөлімшелерден тұрады, олардың әрқайсысы өз функцияларын орындайды: қамыр илеуіш қамырды араластырады, ол қамырды кесу, содан кейін пісіру және әрлеу бөліміне түседі.

Әрбір кондитерлік цех үй-жайлар өндірістегі барлық операциялар орындалатын ретпен жүретіндей етіп салынады. Пісіру жасау кондитерлік өндірістің жұмысы тұтынушылардан алынған өтінімдер негізінде құрылады. Оларға сәйкес арнайы тоназытқыш шкафтарында дұрыс сақталатын шикізаттың қажетті мөлшері есептеледі. Бірінші кезеңде цехтағы еңбек процесі өнімді дайындаудан, жұмыртқаны өңдеуден және ұнды електен өткізуден басталады. Бұл арнайы жуу ванналарында және өндіріс үстелінде жасалады. Елегіш ұннан механикалық қоспаларды жояды, кондитерлік өнімдерді сапалы ету үшін оны бос етеді. Қамырды илеу ашытқы, ашытылмаған немесе қысқа қамырды тез және сапалы илейтін илеу машинасында жасалады.

Егер қатпарлы қамыр жасау қажет болса, қамыр илейтін машина қолданылады. Сұйық ашытқы немесе жұмсақ қысқа қамыр жасалады, кремдер, суфле, желе шайқалады. Кесу, қалыптау және пісіру Әрі қарай, еңбек процесі кондитерлік өнімдерді кесуді және пішіндеуді қамтиды. Бұл бөлек бөлімде жасалады, онда әртүрлі қамырды кесуге арналған орындар бар. Қатпарлы және қысқа қамыр кесіліп, Тоңазытқыш үстеліне құйылады, өйткені қамырдың бұл түрлері пісіру кезінде салқындатуды қажет етеді. Қысқа, печенье, қатпарлы қамырдан жасалған қалыпталған өнімдер бірден кондитерлік парақтарға жіберіліп, пісіріледі. Кондитерлік өнімдерді өндірудің технологиялық процесі, жартылай фабрикаттар дайындық үшін термиялық өңдеуден өтеді және бұл кезең бүкіл технологиялық схемада ең маңызды болып табылады.

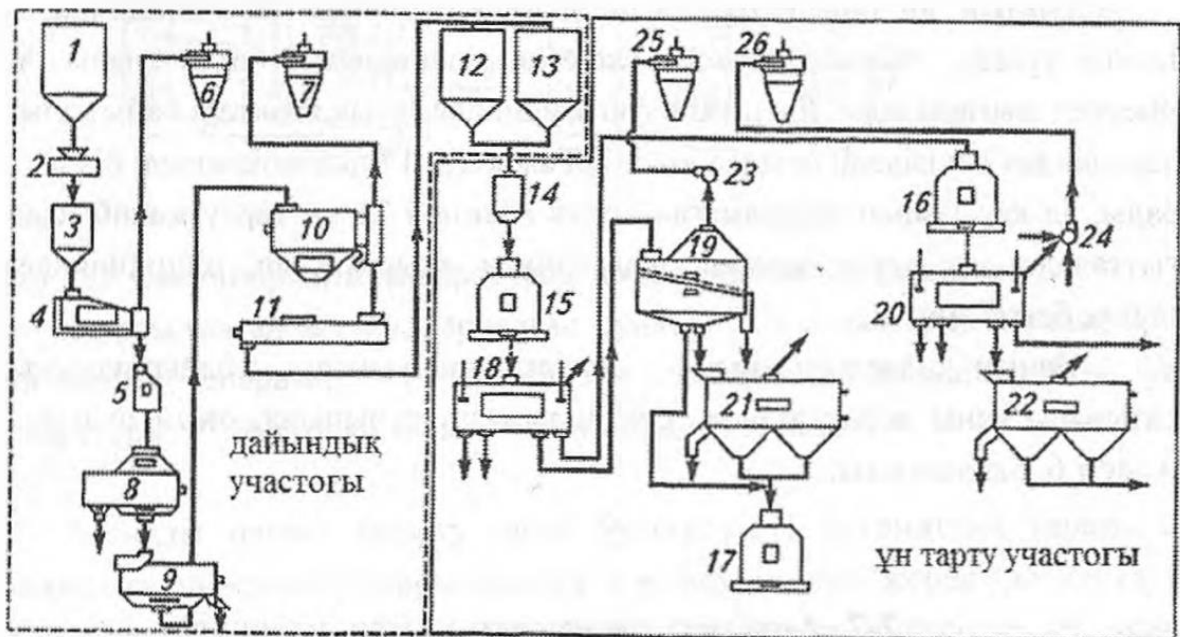
## **1.2 Кондитерлік өнімді дайындау процесі**

Кондитерлердің әр түрі белгілі бір температурада және термиялық өңдеу ұзақтығында жасалады, оны сақтау қажет. Пісіру 2-4 камерадан тұратын арнайы наубайхана шкафтарында жүзеге асырылады. Пісіру аяқталғаннан кейін дайын кондитерлік өнімдер салқындатылатын салқындату бөліміне жіберіледі. Торттар мен торттарды безендіру Технологиялық процестің схемасы кондитерлік өнімдерді жобалауды да қамтиды. Бұл торттар кесуге, сіндіруге, майлауға және безендіруге ұшырайтын жеке өндірістік учаскеде жасалады. Технологиялық схема бұл аймақты арнайы жабдықтармен және тоңазытқыш үстелдермен, сироптар мен помадаларды дайындайтын араластырғыштармен, крем дайындауды талап етеді. Алынған өнімдер сақтауға жіберіледі: кілегей мен жемістермен толтырылған өнімдер тоңазытқыш шкафтарда сақталады, онда температура 6-8 градус (1.1 суретте көрсетілген).



### 1.1 сурет – Шоколадтың өндірісі

Шоколад өндірісі кондитерлік өнімдерді өндірудің технологиялық процесі, шоколадты кондитерлік өндіріс үгітілген какао мен какао майынан жасалады. Қоспалар ретінде қант ұнтағы, сүт немесе кілегей, эмульгаторлар, жаңғақтар және әртүрлі хош иістер қолданылады. Шоколад өндірісінің кезеңдері келесідей: какао өнімдері үшін какао бұршақтары өңделеді; шоколад массасы мен қоспалар дайындалады; шоколад құйылады; шоколад оралған. Шоколад өндірісінде тауарлық какао бұршақтары арнайы жабдықта тазаланады және мөлшері бойынша сұрыпталады. Таңдалған өнімдер термиялық өңдеуге жіберіледі-бұл ылғалды кетіру және бұршақтың дәмін жақсарту үшін қажет. Тазартылған және салқындатылған бұршақтар арнайы машинада ұсақталады, ал қабық пен эмбрион бөлінеді. Алынған какао фракциялары шоколадтың әртүрлі түрлерін өндіру үшін қолданылады. Десертті шоколадтың жоғары сорттары үлкен бұршақтардан (6-8 мм) жасалғанын ескереміз.



1, 3, 12, 13 - бункерлер; 2 - автоматтық таразылар; 4 - астық тазалайтын сепаратор; 5, 14 - магниттік колонкалар; 6, 7, 25, 26 - циклондар; 8, 9 - триерлер; 10 - сыдыралау машинасы; 11 ылғалдандырғыш; 15,16,17 - білікшелі білдектер; 18, 20 – елеп-екшегіштер; 19 - елеп-екшегіш машина; 21, 22 - ұн тарту машиналары; 23, 24 - желдеткіштер

1.2 сурет - Бидай астығын ұнға қайта өңдеуге арналған аппаратуралық технологиялысының сұлбасы

Карамель өндірісі кондитерлік өнімдерді өндірудің технологиялық процесі. Кондитерлік өндіріс карамель массасынан толтырумен немесе қоспасыз жүргізіледі. Карамель өнімдерін өндіру бояғыштар, түрлі қоспалар, майлар, сүт өнімдері қосылған түйіршіктелген қант пен мелассадан жасалады. Технологиялық тұрғыдан алғанда процесс бірнеше кезеңнен тұрады: Карамель сиропы дайындалады. Оның ылғалдылығы 16% - дан аспауы керек. Карамель массасы алынады. Толтырулар дайындалады. Карамель қалыптасады және салқындатылады. Алынған өнімді орау, буып-түю және буып-түю орындалады. Сироптар үздіксіз немесе мезгіл-мезгіл дайындалады. Кез-келген нұсқада ол карамель массасының ылғалдылығы максимум 3% жеткенше қайнатылады. Бұл көрсеткіште масса аморфты күйде болады.

Танымал кондитерлік өнімдер-пастила мен мармелад. Оларды өндіру белгілі бір технологиялық схема сақталған кезде арнайы жабдықта да жүзеге асырылады. Сағыз-пастильді өнімдер жемістер мен жидектер негізінде жасалады, оларға көбік түзетін заттар мен желатинді түзетін заттар қосылады. Көптеген кондитерлік фабрикалар алма, қалып, қабат мармеладын, жеміс-жидек өнімдерін және желе түрінде шығарады. Мармелад пен пастиланы өндіруге арналған технологиялық процестер жүйесі қант пен жемістердің массасын жұмыртқаның ақуызымен құлатуға негізделген. Алма-қант қоспасына қандай масса қосылатынына байланысты пастила желім және қаймақ болып табылады.

## **2 Кондитерлік фабриканың автоматтандырылуы**

### **2.1 Жалпы мәліметтер**

Партия мен үкіметтің еңбек игілігінің күрт өсуі туралы шешімдерін ескере отырып, менде азық-түлік тауарларын өндіру өнеркәсібінің алдында кәсіпорындардың өнімділігін едәуір арттыру және өндірілетін өнімнің сапасын арттыру міндеті тұр. Бұл міндеттерді шешу өндірістік процестерді механикаландыру мен автоматтандыруды кеңінен енгізумен байланысты.

Соңғы жылдары кондитерлік өнеркәсіпте техникалық қайта жарақтандыру процесі жүріп жатыр. Өнеркәсіп жоғары өнімді машиналармен және аппараттармен жабдықталған.

Қазіргі уақытта мен карамель, печенье, мармелад және басқа да өнімдерді өндірудің үздіксіз ағындық желілерін енгізіп жатырмын. Механизатор еңбекті қажет ететін процестерге ие және еңбек мәдениеті артады.

Бұдан әрі сапалық көрсеткіштерді жақсартуға және еңбек өнімділігі туралы ұлғайтуға жаппай шығарылатын өнімдер өндірісінің ағындық желілерінде технологиялық процестерді автоматтандыруды кеңінен енгізу арқылы қол жеткізуге болады. Мұндай өнімдердің қатарына, мысалы, карамель жатады, оның кондитерлік өнімдер өндірісіндегі үлес салмағы ~40% жетеді.

Карамель өндірісінде қолданылатын технологиялық процестер негізінен кондитерлік өнімдер өндірісіне тән. Бұл процестер туралы кейбір түсініктерді 2.1-суретте келтірілген технологиялық схема бойынша жасауға болады.

Қара-мель өндірісі мынадай негізгі технологиялық операциялардан тұрады:

- 1) карамель сиропын өндіру;
- 2) карамель массасы;
- 3) толтыру дайындау;
- 4) карамель массасын өңдеу;
- 5) карамельді қалыптау;
- 6) оны салқындату;
- 7) орау және орау.

Сиропты үздіксіз дайындау процесі араластырғышпен бумен қыздырылған секциялық еріткіште жүзеге асырылады.

Қант, меласса және су белгілі бір пропорцияда еріткішке үздіксіз жеткізіледі. Еріткіштен шығатын соңғы өнім-тұрақты ылғалдылығы бар сироп (шамамен 20%). Процестің осы кезеңінде қант, меласса және су шығындарының арақатынасын автоматты түрде сақтау қажет.

Карамель мен масса сироптан оны катушкалық вакуумдық аппаратта қайнату арқылы алынады.

Вакуумдық аппараттан шығатын карамель массасының ылғалдылығы өнімнің стандарттылығы жағдайынан және одан әрі технологиялық процестің дұрыстығын қамтамасыз етуден қатаң белгіленген шектерде сақталуы тиіс.

Үздіксіз таспа түріндегі дайын карамель массасы салқындатқыш машинаға шұңқыр арқылы түседі, оның мақсаты карамель массасының температурасын 120-130° - тан қатаң тұрақты температураға дейін төмендету турлар (^80°C). Бұл учаскедегі карамель массасының температурасын реттеу Оны тарту және катальды машиналарда өңдеу мүмкіндігі үшін қажет. Толтыру ұсынылатын катальды машинада карамель турникеті туралы, одан кейін карамель пайда болады; бұл жерде карамель массасын тұтыну мен толтыру арасындағы белгілі бір қатынасты қамтамасыз ету қиын міндет болып табылады.

Соңғы уақытқа дейін бұл учаскеде ешқандай автоматты бақылау , тіпті одан да көп реттеу жоқ. Қалыпталған карамель салқындатылып, орау машинасына түседі.

Карамель өндірісінің ағындық желісінің бастапқы сатысында әрекет ететін агрегаттардың бірі вакуумдық аппарат болып табылады. Ол бастапқы ылғалдылығы 20% болатын карамель сиропын қайнатады. Аппараттан карамель массасы түрінде шығатын дайын өнімнің ылғалдылығы стандарт талаптарына сәйкес және технологиялық себептерге байланысты 2-2,5% деңгейінде сақталуы керек. Карамель массасының қайнау температурасы ылғалдылық пен қысымға байланысты болғандықтан, ылғалдылықты реттеу вакуумдық аппараттағы сиретудің өзгеруін ескеретін температура реттегіші арқылы мүмкін болады.

Автоматты реттегіш вакуумдық аппараттан шығатын карамель массасының ылғалдылығын сақтау жүзеге асырылды. Бұл гидравликалық типтегі ылғалдылық реттегіші. Ол тіркеу құрылғысымен жабдықталған.

Атқарушы механизм ретінде өнеркәсіптік өндірістің мембраналық клапаны қолданылды.

Вакуум-аппараттың жұмыс жағдайлары бойынша карамель массасын түсіру мезгіл-мезгіл жүргізілуі тиіс. Қазіргі уақытта бұл операция ол арнайы әзірленген ав түсіру қызанағының көмегімен автоматты түрде жасалады.

Бұл құрылғыларды қазіргі уақытта МОСКИП зауыты жаппай шығарады. Осылайша, катушкалық вакуумдық аппараттағы массаны қайнатудың техникалық-логикалық процесін автоматтандыру және реттеу міндеті шешілген деп санауға болады.

Қазіргі уақытта үздіксіз жұмыс істейтін Е вакуумдық аппаратына толтыруды қайнату процесін автоматтандыру жүзеге асырылуда.

Мен салқындату машинасында карамель массасының температурасын реттеу бағытында жұмыс жүргіземін.

Толтыру тақырыптары үшін қызмет ететін температуралық машиналардың жұмысы автоматтандырылады. Сироп станциясының жұмысын автоматтандыру жүргізіледі. Технологиялық процестің әртүрлі кезеңдерінде температураны бақылау жүзеге асырылады. Печенье өндірісінің ағындық желісінің қамыр илеу станциясында технологиялық процестерді басқаруды орталықтандырумен кешенді автоматтандыру жүзеге асырылады.

Ирис массасын қайнату процесі де автоматтандырылған.

Кондитерлік өнеркәсіптегі технологиялық процестерді бақылау мен реттеудің жай-күйі кондитерлік өнімдердің қазіргі заманғы ағындық өндірісінің талаптарына жауап бермейді бұйымдар.

Соңғы уақытқа дейін шығармашылық ой ең алдымен кондитерлік өнеркәсіпте кеңінен қолданылатын қол процестерін механикаландырылған процестерге ауыстыруға бағытталған.

Механикаландыру саласында соңғы уақытта айтарлықтай жетістіктерге қол жеткізілді. Жаңа технологиялық жабдықты пайдалану максималды автоматтандыруды қажет етеді.

Кондитерлік өнеркәсіптегі технологиялық процестерді автоматтандыру саласындағы негізгі міндеттер: өнімдерді қайнату процестерін бақылау және автоматтандыру; өңделетін массалардың температурасын реттеуді бақылау және автоматтандыру және сусымалы, сұйық және тұтқыр массалардың мөлшерін бақылау және автоматтандыру.

Автоматтандыру басқа салаларда қолданылатын құралдармен жүзеге асырылуы мүмкін процестерден басқа, бірқатар процестер ерекше және автоматты реттеуді жүзеге асыру өңделетін материалдардың қасиеттеріне байланысты техникалық қиындықтарға тап болады. Кондитерлік өнеркәсіпте карамель массасы, жеміс-жидек қоспалары, сироптар сияқты тұтқыр, жабысқақ өнімдерді орау керек.

Сондықтан, мысалы, салқындатқыш машинаның үстелінде таспа түрінде қозғалатын карамель массасының температурасын өлшеу мен үшін өте қиын міндет. Тиісті сенсорды (сезгіш элементті) құру - карамель массасының қарқынын реттеу мәселесін шешуде алға жасалған қадам.

Қиын міндет-шығындар мен жылдамдықты өлшеу ди жени мен тұтқырлығы жүздеген болатын тұтқыр орталардың құбырлары арқылы және мыңдаған пуазалар. Қолданыстағы құрылғылар дл мен сұйықтық ағынының жылдамдығы мен жылдамдығын бақылау аталған мәселені шешуге мүмкіндік бермейді. Сондықтан қазіргі уақытта қолданылатын сұйық өнімдерді дозалаудың барлық әдістері, кондитерлік Өнеркәсіпте қолданылатын сорғыларды диспенсер ретінде пайдалану бір нәрсеге дейін азаяды. Бақылау әдістерін табыңыз дозалау және мөлшерлеу құралдары - жақын арада шешілуі керек маңызды міндеттердің бірі.

Басқа салалардағы сусымалы өнімдерді мөлшерлеу тәжірибесін қант, қант ұнтағы, ұн және т. б. диспенсерлерді жасау кезінде ескеру қажет.

Тұтқыр өнімдер мен сусымалы өнімдердің ылғалдылығын реттейтін автоматты құрылғыларды жасауда айтарлықтай қиындықтар туындайды.

Белгілі бір кондитерлік өнімдерді өндіруді кешенді автоматтандырудың алдында автоматты бақылау жүргізілуі керек және жекелеген кезеңдердегі технологиялық процестерді реттеу.

Осыған байланысты, құрылыс зауыттары шығаратын көптеген аппараттар мен машиналар мен кондитерлік өнеркәсіпке арналған машиналар мен ондаған жылдар бойы айтарлықтай өзгеріссіз шығарылғанын атап өткен жөн.

Әрине , мұндай жабдыққа автоматты реттегіштерді, кейде тіпті қарапайым Бақылау-өлшеу құралдарын орнату жабдықтың өзін өзгертпестен мүмкін емес. Машиналар мен аппараттарды шығаратын зауыттар олардың жұмысын автоматтандыруды қамтамасыз ету тұрғысынан олардың дизайнын қайта қарауы керек. Жоғарыда айтылғандар кондитерлік өнеркәсіптің автоматтандыру саласындағы міндеттерінің қаншалықты маңызды және маңызды екенін көрсетеді.

### **3 Практикалық бөлім**

#### **3.1 Ардуино программасы**

Arduino-автоматика және робототехника жүйелерін құруға арналған аппараттық-бағдарламалық құрал. Басты артықшылығы-платформа кәсіби емес пайдаланушыларға бағытталған. Яғни, кез-келген адам бағдарламалау біліміне және өз дағдыларына қарамастан өз роботын жасай алады.

Arduino-да жобаны құру 3 негізгі кезеңнен тұрады: кодты жазу, прототиптеу (макет) және микробағдарлама. Кодты жазып, содан кейін тақтаны

жыпылықтау үшін бізге даму ортасы қажет. Шындығында, олардың көпшілігі бар, бірақ біз түпнұсқа ортада – Arduino IDE-де бағдарламалаймыз. Біз кодты Arduino-ға бейімделген C++ тілінде жазамыз. Сіз ресми сайттан жүктей аласыз. Эскиз (эскиз) - Arduino-да жазылған бағдарлама. Кодтың құрылымын қарастырайық:

```
main(){  
void setup(){  
void loop(){}}
```

Arduino процессоры c++ тілінде міндетті түрде Main() функциясын өзі жасайтынын атап өткен жөн. Және бағдарламашы көретін нәтиже:

```
void setup(){  
void loop(){}
```

Екі міндетті функцияны қарастырайық. Setup() функциясы микроконтроллер іске қосылған кезде тек бір рет шақырылады. Ол барлық негізгі параметрлерді орнатады. Loop () функциясы — циклдік. Ол микроконтроллердің бүкіл жұмыс уақытында шексіз циклде шақырылады.

Платформаның жұмыс принципін жақсы түсіну үшін бірінші бағдарламаны жазайық. Бұл қарапайым бағдарламаны (Blink) біз екі нұсқада орындаймыз. Олардың арасындағы айырмашылық тек құрастыруда.

```
int Led = 13; // 13 pin (шығу) үшін LED айнымалысын жариялаймыз  
void setup(){  
pinMode (Led, OUTPUT); // айнымалыны анықтаймыз}  
void loop(){  
digitalWrite (Led, HIGH); // 13 пин кернеуі  
delay (1000); // 1 секунд күтіңіз  
digitalWrite (Led, LOW); // біз 13 пин кернеуін бермейміз  
delay (1000); // 1 секунд күтіңіз}
```

Бұл бағдарламаның жұмыс принципі өте қарапайым: жарық диоды 1 секундқа жанып, 1 секундқа сөнеді. Бірінші нұсқа үшін орналасуды жинаудың қажеті жоқ. Arduino платформасында кірістірілген жарық диоды 13 пинге қосылған.

Эскизді Arduino-ға толтыру үшін алдымен оны сақтау керек. Әрі қарай, жүктеу кезінде проблемаларды болдырмас үшін бағдарламашының параметрлерін тексеру керек. Ол үшін жоғарғы панельде "Құралдар" қойындысын таңдаймын. "Төлем" бөлімінде төлемді таңдаймыз. Бұл Arduino Uno, Arduino Nano, Arduino Mega, Arduino Leonardo немесе басқалары болуы мүмкін. Сондай-ақ, "Порт" бөлімінде сіз қосылу портын таңдауыңыз керек (платформаңызды қосқан порт). Осы әрекеттерден кейін эскизді жүктеуге болады. Мұны істеу үшін көрсеткіні таңдаймыз немесе "эскиз" қойындысында "жүктеу" бастырмасын басу керек (сонымен қатар "Ctrl + U" пернелер тіркесімін пайдалануға болады). Микробағдарлама сәтті аяқталды.

Орналасуды құрастыру үшін бізге келесі элементтер қажет: жарық диоды, резистор, сымдар (секіргіштер), орналасу тақтасы (Breadboard). Ештеңені күйдіріп алмау үшін және бәрі сәтті жұмыс істеуі үшін жарық диодымен күресу

керек. Оның екі "аяғы" бар. Қысқа-минус, ұзақ-плюс. Қысқа, біз қосу "жер" (GND) және резистор (бұған азайтуға ток күші түседі).

Arduino жобасы әрдайым электронды схеманың, кейбір аппараттық және механикалық құрылғылардың, қуат жүйесінің және осы хаосты басқаратын бағдарламалық жасақтаманың үйлесімі болып табылады.

Егер бұл оқу жобасы туралы болмаса, онда сіз міндетті түрде осындай міндеттерді жүзеге асырудың келесі кезеңдеріне тап боласыз.

Басқаларға пайдалы және (немесе) қызықты болатын нәрсені ойлап табыңыз. Тіпті қарапайым жобаның да пайдасы бар-кем дегенде, ол жаңа технологияларды үйренуге көмектеседі.

Схеманы жинаймыз, модульдерді бір-біріне және контроллерге қосамыз.

Эскизді (бағдарламаны) арнайы ортада жазыңыз және оны контроллерге жүктеңіз.

Барлығы қалай жұмыс істейтінін тексеріп, қателерді түзетемін.

Тестілеуден кейін-дайын құрылғыны жасауға дайындаламын. Бұл құрылғыны пайдалануға болатын корпуста жинау керек, қуат жүйесін, қоршаған ортамен байланысты қамтамасыз ету керек дегенді білдіреді.

Егер сіз өзіңіз жасаған құрылғыларды таратқыңыз келсе, сізге дизайнды, тасымалдау жүйесін жасау керек, оқытылмаған пайдаланушылардың пайдалану қауіпсіздігі туралы ойлану керек және сол пайдаланушыларды оқыту керек.

Егер біздің құрылғымыз жұмыс істеп тұрса, және басқа шешімдерге қарағанда қандай да бір артықшылықтарға ие болса, онда инженерлік жобаныңдан бизнес-жоба жасап, инвестиция тартуға тырысуға болады.

Arduino nano дегеніміз не: ерекшеліктері мен қолдану аясы.

Жобаны құрудың осы кезеңдерінің әрқайсысы жеке мақалаға лайық. Бірақ біз электронды схемаларды (электроника негіздері) құрастыру және контроллерді бағдарламалау кезеңдеріне назар аударамыз.

Электрондық тізбектер, әдетте, элементтерді бір-біріне дәнекерлеусіз және бұрамай бекітетін орналасу тақталарын қолдана отырып жиналады. Модульдер мен қосылу схемалары қалай жұмыс істейтіні туралы біздің веб-сайттан білуге болады. Әдетте жобаның сипаттамасында бөлшектерді орнату әдістері көрсетілген. Бірақ көптеген танымал модульдер үшін интернетте ондаған дайын схемалар мен мысалдар бар.

Эскиздерді жасау және микробағдарламалау арнайы бағдарламада – бағдарламалау ортасында жүзеге асырылады. Мұндай ортаның ең танымал нұсқасы-Arduino IDE. Біздің сайтта сіз осы бағдарламаны жүктеу, орнату және конфигурациялау туралы ақпаратты таба аласыз.

Көптеген электрониктер өз жобаларын біз бірнеше рет жазған Arduino Uno микроконтроллері негізінде құруды жөн көреді.

Бастау үшін, тұр танысуға функционалы микропроцессордың Ардуино сжқ, онда құрылады жобалардың көпшілігі, сондай-ақ қарауға таңдау себептерін, осы керек-жарақтар. Төменде бастаушы өнертапқыш Arduino uno-да тоқтауы керек факторлар сипатталған.

Интерфейсті пайдалану өте оңай. Байланыс қай жерде және байланыс сымдары қайда бекітілетіні анық.

Тақтадағы чип тікелей USB портына қосылады. Бұл қондырғының артықшылығы-сериялық байланыс-бұл өте қарапайым протокол, ол уақытпен тексерілген және USB заманауи компьютерлерге қосылуды өте ыңғайлы етеді.

Atmega328 чипі болып табылатын микроконтроллердің орталық бөлігін табу оңай. Ол таймерлер, сыртқы және ішкі үзілістер, PWM түйреуіштері және бірнеше күту режимдері сияқты көптеген аппараттық мүмкіндіктерге ие.

Ашық бастапқы құрылғы, сондықтан көптеген радиоэуесқойлар бағдарламалық жасақтамадағы қателер мен мәселелерді шеше алады. Бұл жобаларды жөндеуді жеңілдетеді.

Сағат жылдамдығы 16 МГц, бұл көптеген қосымшалар үшін тез және микроконтроллерді тездетпейді.

Оның ішіндегі қуатты басқару өте ыңғайлы және кернеуді реттеу функциясы бар. Сондай-ақ, микроконтроллерді сыртқы қуат көзі жоқ USB портынан ажыратуға болады. Сыртқы қуат көзін 12 В дейін қосуға болады, сонымен қатар микропроцессордың өзі қажетті кернеуді анықтайды.

13 сандық және 6 аналогтық контактілердің болуы. Бұл түйреуіштер жабдықты Arduino uno тақтасына үшінші тарап медиасынан қосуға мүмкіндік береді. Контактілер Arduino uno есептеу қабілетін нақты әлемде кеңейту үшін кілт ретінде қолданылады. Электрондық құрылғылар мен сенсорларды осы контактілердің әрқайсысына сәйкес келетін коннекторларға қосыңыз.

USB портын айналып өтуге және Arduino-мен тікелей сериялық құрылғы ретінде жұптауға арналған ICSP қосқышы бар. Бұл порт чипті қалпына келтіру үшін қажет, егер ол зақымдалған болса және оны енді компьютерде пайдалану мүмкін болмаса.

Әзірлеуші кодын сақтау үшін 32 КБ флэш-жады бар.

Кодты тез түзету және процесті жеңілдету үшін тақтадағы жарық диоды 13 сандық контактіге қосылады.

Соңында, чипте бағдарламаны қалпына келтіруге арналған түйме бар.

Arduino 2005 жылы екі итальяндық инженер – Дэвид Куартиллес пен Массимо Банзи студенттерге Arduino UNO микроконтроллерін бағдарламалауды және электроника дағдыларын жетілдіруді және оларды нақты әлемде қолдануды үйрену мақсатында құрылды.

Arduino uno қоршаған ортаны әртүрлі Сенсорлардан қабылдау арқылы қабылдай алады және қоршаған ортаға жарық, қозғалтқыштар және басқа да жетектерді басқару арқылы әсер ете алады. Микроконтроллер Arduino бағдарламалау тілін (сым негізінде) және Arduino даму ортасын (өңдеу негізінде) қолдана отырып бағдарламаланған.

### **3.2 Жобаны іске асыру**

Дипломдық жұмыста мен IoT қолданған кезде қазандықты басқаруды автоматтандыруды жасауым керек. Біз wemos D1 mini arduino nano, MT4, mq 8, dht11 микроконтроллерін қосу арқылы макет жасадым.

Жақын арада 2014 жылы WiFi стандартына сәйкес ESP8266EX сымсыз чипінің шығарылуымен қатар, үйде жасалған құрылғыларды жинауды ұнататындар арасында өте танымал болды, WEMOS Electronics компаниясы Arduino үйлесімді платформаларын дамытудың жаңа векторын орнатып, wemos брендімен өзінің модульдік электронды тақталарының үлкен желісін ұсынды. Espressif Systems компаниясының микроконтроллерін негізге ала отырып, әзірлеушілер негізгі модульдердің көлемін азайтуға ғана емес, оларды мүмкіндігінше ықшам етуге ғана емес, сонымен қатар кеңейтімдердің кең спектрін шығаруға да көп күш жұмсады, пайдаланушыларға әртүрлі пайдалы өнертабыстарды жобалау мен жобалауда әртүрлі күрделілік деңгейіндегі тапсырмаларды орындауға көмектесетін бай функционалдылықты ұсынды.

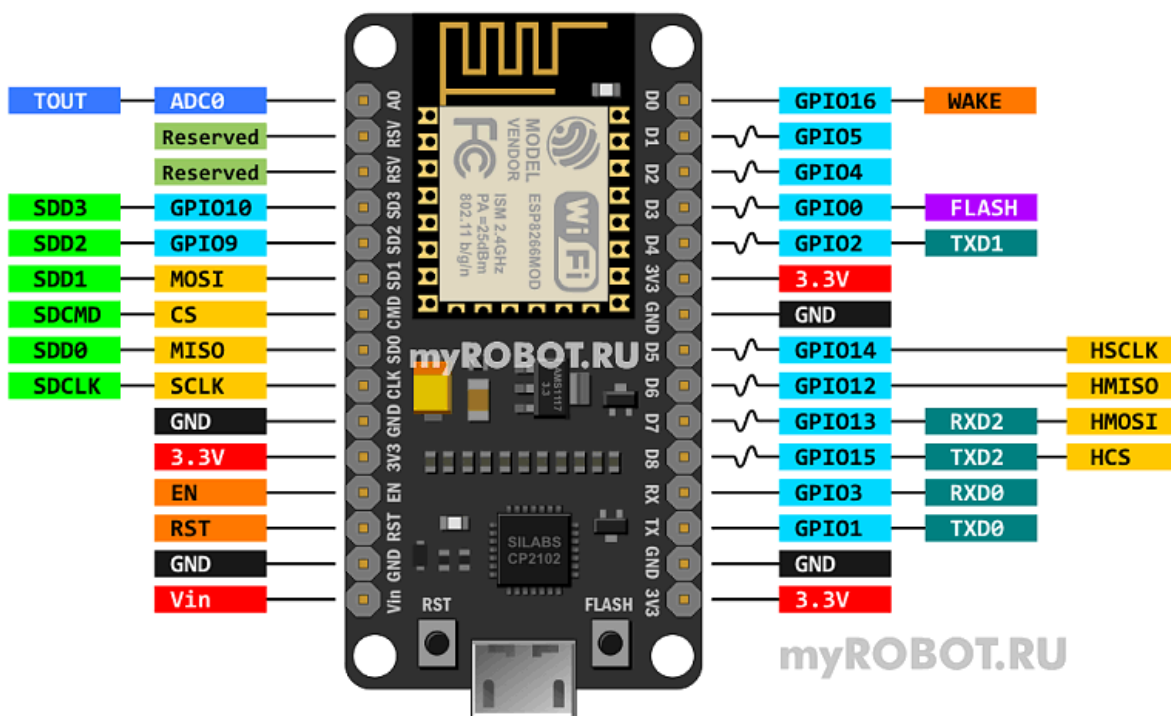
Қазіргі заманғы WeMos электронды өнімдері тиімді төмен құнды шешімдер санатына кіреді және оларды жасаушылар WiFi технологиясы бойынша сымсыз байланыс негізінде көп мақсатты құрылғыларды тәуелсіз дамытуға арналған ашық платформа ретінде орналастырады. Бұл ретте негізгі назар "Заттар интернеті" (ағылш. - Internet of Things, IoT), бұлтты есептеу және машинааралық өзара әрекеттесудің қазіргі мүмкіндіктері белсенді ықпал етеді.

Тұтынушы тақталардың физикалық үйлесімділігіне араласпауы үшін әр серияның өзіндік ерекшеліктері бар. Мысалы, ESP8266EX чипіндегі WiFi контроллері бар модульге жататын WeMos D1 Mini сериясында 3v3 таңбасының жанында текстолит бетінің көк түсі және кішкене дөңгелек бұрыштық ойық бар.

Wemos техникалық сипаттамалары:

- үлгі: 3.0.0;
- контроллер: esp8266ex serial wifi, tensilica 1106 негізіндегі бір чипті жүйе;
- ультра төмен қуат, 32 биттік бит;
- сағат жиілігі: 80 мгц (160 мгц дейін);
- жұмыс кернеуі: 3.3 в;
- қуат кернеуі: 5 в;
- тұтынылатын ток: 300 ма дейін;
- флэш жады: 32 мбит / 4 мбит;
- жиілік диапазоны: 2.4 ггц-2.5 ггц (2412м-2484м);
- wifi режимдері: клиент, бағдарламалық жасақтама кіру нүктесі, клиент+бағдарламалық жасақтама кіру нүктесі (station, softap, station+softap);
- қорғау: wpa-psk, wpa2-psk;
- шифрлау: wep, tkip, aes;
- wifi протоколдары: 802.11 b / g / n;
- жалпы мақсаттағы тұжырымдар (кіру/шығу, gpio): 11;
- аналогтық кіріс: 1, 10 биттік бит;

- жалпы мақсаттағы контактіге максималды ток: 12 ма, ұсынылған 6 ма;
- stbc технологиясы, 1x1 mimo, 2x1 mimo;
- 802.11 b режиміндегі шығу қуаты: +20dbm;
- интерфейстер: gpio, uart, i2c, hspi, pwm;
- деректерді беру жылдамдығы: 300-4608000 бод, әдепкі 115200 бод;
- кірістірілген кабылдау/беру қосқышы, жоғары жиілікті трансформатор, қуат күшейткіші;
- кіріктірілген блоктар: желіні үйлестіру, жиілікті фазалық реттеу, қуатты басқару, реттеу блоктары;
- wifi антеннасы: пхд, жол түрінде тақтада ажырасқан;
- arduino, nodemcu, micropyton қолдау;
- spiffs файлдық жүйесін қолдау;
- andriod және ios құрылғыларына арналған smart link мүмкіндіктерін қолдау;
- ат-командаларды, бұлтты серверді және әзірлеу жиынтықтарын (sdk) қолдау, микробағдарламаны жаңарту;
- жарық диодты индикатор: d4 шығысы (gpio2);
- контактілер арасындағы қадам: 2.54 мм;
- жұмыс температурасы: -20°...+85°;
- өлшемдері: 34.2 x 25.6 x 5 мм;



3.1 сурет – Платаның кірістері

Кішкентай көрінетін wemos D1 Mini тақтасы әуесқойларға немесе кәсіби әзірлеушілерге өздерінің жобаларын "нөлден" құруға мүмкіндік беретін әсерлі

мүмкіндіктерге ие. WeMos микроконтроллерлік платформасы ескірмейтін Ардуиноның толық алмастырушысы бола алады, ал ол өнімділік пен функционалдылықтағы тікелей әріптесінен әлдеқайда жоғары. Қуатты 32 биттік процессор сізге есептеулерді немесе деректерді өңдеуді тезірек орындауға мүмкіндік береді, ал WiFi сымсыз байланысы арқылы басқа электрондық құрылғылармен байланысу мүмкіндігі қарапайым немесе күрделі ақылды құрылғыларды жобалау мен жобалауда кең мүмкіндіктер ашады.

Wemos платформасы-пайдаланушыға белгілі бір тапсырмалар үшін "қайралған" схемаларды жасауға мүмкіндік беретін модульдік платформа. Ол OLED дисплейлерінен, қосқыштардан, температура, қысым және жарық сенсорларынан, жарықдиодты шамдардан және серво-мотор контроллерлерінен үйлесімді кеңейтімдердің үлкен жиынтығын қолдайды, оларды бір-бірінің үстіне қою арқылы өз қалауыңыз бойынша құрастыруға болады.

Атаудағы "мини" сөзі кездейсоқ пайда болған жоқ. ESPRESSIF Systems стандартты non-OS микробағдарламаларында сөйлесу мүмкіндігімен қатар, ат командалары арқылы байланысты басқарудың ыңғайлы әдісін қолдана отырып, ESP8266EX чипі NodeMCU деп аталатын Lua тілінің кіріктірілген аудармашысымен микробағдарламаны жақсы түсінеді. Әрине, түпнұсқа толық өлшемді NodeMCU тақталары қол жетімді түйреуіштер санына байланысты әлдеқайда функционалды, бірақ олардың көпшілігінде әлдеқайда үлкен. Бос орынды сақтау үшін WeMos D1 Mini контроллерінде бірнеше контактілер жоқ, сонымен қатар түйреуіштер дәнекерленген емес. Әйтпесе, ол "мини" моделі болмас еді.

Байланысты ұйымдастыру және ESP8266EX чипін бағдарламалау UART интерфейсінің негізгі шинасы арқылы жүзеге асырылады. WeMos D1 Mini компьютерден USB портына келетін сигналдарды түсінуді үйренуі үшін модуль схемасында CH340G чипіндегі USB-TTL сигнал түрлендіргіші біріктірілген. бірінші қосқан кезде компьютердің операциялық жүйесі, егер тиісті драйвер анық орнатылмаған болса, оны анықтай алмайды. Ch340 драйверін орнатамын, содан кейін тақтаны USB кабелімен компьютерге қосамын. Wemos D1 Mini контроллері жұмыс істеуге дайын.

Wemos D1 Mini микроконтроллеріне қуат беру екі жолмен мүмкін.

MicroUSB 2.0 порты арқылы 5 вольт кіріктірілген реттегішке түседі және 3.3 вольт номиналды кернеуге дейін төмендейді. Бастапқы және түрлендірілген кернеудің екі түрі де тиісті таңбамен тақтаның тиісті контактілеріне шығарылады.

5V және GND түйреуіштері арқылы (мысалы, кернеу түрлендіргіш модулімен немесе резервтік қуат модулімен біріктірілген жағдайда). Кіріс кернеуі 5 вольт реттегіш арқылы өтеді және жұмыс деңгейіне дейін төмендейді.

ME6211 реттегіш чипі 500 миллиамперден аспайтын ток өткізеді. Реттегішті қорғау үшін модульде полимерлі өзін-өзі емдейтін сақтандырғыш орнатылған. Esp8266ex контроллері шыңында 300 миллиамперге дейін тұтынады, сондықтан қосымша кеңейтімдерді немесе сенсорларды қосу үшін шамамен 200 миллиампер қалады.

WeMos D1 Mini-ді 3v3 шығысы арқылы қуаттандыру ұсынылмайды. Кіріс кернеуі ESP8266 чипіне тікелей түседі, ол барлық кернеу мен қысқа тұйықталудан қорғайды.

Сіз әрқашан кішкентайдан бастауыңыз керек. Бағдарламалау әдістерімен танысудың ең алғашқы және қарапайым мысалы әрқашан эскиздер болып табылады, ол тақтаға орнатылған жарықдиодты жыпылықтайды. Төменде келтірілген бағдарламаның мәтіні ғана емес, зажигаєт және гасит светодиод, ал бұл барынша қалыпты.

```
// кіріктірілген жарықдиодты анықтау
const int ledPin = BUILTIN_LED;
// жарық диоды айнымалы (0 = ең жарқын,
// 512 = жарықтылықтың жартысы , 1023 = толығымен өшірілген)
int brightness = 0;
//өшу / тұтану қадамы бар айнымалы
int fadeAmount = 5;
// әрбір циклде шағын аралық
const int delayMillis = 10;
void setup() {
  // шығару үшін жарықдиодты инициализациялау
  pinMode(ledPin, OUTPUT);
}
void loop() {
  // жарық диодының жарықтығын орнату
  analogWrite(ledPin, brightness);
  //келесі цикл үшін жарықтылықты қосу / азайту
  brightness = brightness + fadeAmount;
  // жарықтылық шегін тексеру (10-бит, 0-1023 мәндері)
  if (brightness < 0) brightness = 0;
  if (brightness > 1023) brightness = 1023;
  // егер шекке жетсе, өшу/тұтану арасындағы бағытты өзгерту
  if (brightness == 0 || brightness == 1023) {
    fadeAmount = -fadeAmount;
  }
  / жарықтық деңгейін қарастыру үшін кідірту
  delay(delayMillis);
}
```

WeMos D1 Mini ішкі жады, SPIFFS файлдық жүйесі

Wemos D1 Mini модулі 4 мегабайт (32 мегабит) болатын жад чипімен жабдықталған. Бұл өлшем негізгі микробағдарламаны, жеке кодты сақтауға және қажет болған жағдайда spiffs жеңілдетілген файлдық жүйесін құруға жеткілікті. NodeMCU платформасымен үйлесімділікке бейімделген Arduino IDE ортасында жад әрқашан белгілі бір пропорцияда бөлінеді. Әдепкі бойынша, микробағдарлама мен орындалатын бағдарламаны жазу үшін 1 Мегабайт бөлінеді, Сіз бұл өлшемді көбейте алмайсыз. Қалған 3 мегабайт бос қалады

немесе деректерді сақтау үшін белгіленеді. Микробағдарламаны жаңарту кезінде пайдаланушы файлдық жүйенің қажеттілігін және оның өлшемін таңдайды. Онда нобай деректерін, конфигурация файлдарын немесе веб-сервер мазмұнын сақтауға болады.

Егер кодтың көлемді мәтінін жазу және орындау туралы айтатын болсақ, пайдаланушы Lua тілінің аудармашысымен (NONOS-SDK негізінде) модификацияланған NodeMCU микробағдарламасын қолдана отырып, оның құрамын NodeMCU SDK әзірлеушісінің жиынтықтары арқылы өзгерту арқылы жадты үнемдеуге тырысады. Lua тіліндегі бағдарламалардың барлық орындалатын файлдары SPIFFS аймағында сақталатындығын ескере отырып, модульдің микробағдарламасын NodeMCU-да жаңарту жадыны біршама басқаша таратады, микробағдарламаның өзі бос емес қалған бос кеңістіктен файлдық жүйені құрайды.

NRF24101-2,4 ГГц диапазоны үшін 2мбит/с жоғары интеграцияланған қуат тұтыну чипі (ULP). Модульдің көмегімен мен радиоарна арқылы деректерді беру үшін бірнеше құрылғыны байланыстыра аламыз. Біз жеті құрылғыны 2,4 ГГц жиілікте бір жалпы радио желісіне біріктіре аламыз, модульдердің бірі жетекші, қалғандары жетекші болады. Nrf24101 радио модулі арзан, сондықтан оны әртүрлі жобалардан табуға болады – ақылды үйден бастап әртүрлі роботтарға дейін.

Nrf24101 сипаттамалары:

- төмен энергия шығындары;
- shockburst аппараттық протоколының жетілдірілген үдеткіші бар;
- ism операциялық жүйесі;
- деректерді беру жылдамдығы 250 кбит/с, 1 мбит/с және 2 мбит / с;
- барлық стандартты nrf241 nordic сериясымен, сондай-ақ nrf24e және nrf240 сериясымен толық үйлесімділік;
- қуат кернеуі 3,3 в;
- жұмыс температурасы-40с-тан 85с-қа дейін, сақтау температурасы-40с-тан 125с-қа дейін;
- байланыс қашықтығы 100 м дейін.

Модульдің негізі Nordic Semiconductor компаниясы шығарған nRF24L01+ болып табылады. Чипте барлық қажетті элементтер мен коннектор ашасы орналасқан. SPI интерфейсі арқылы протоколды теңшеуге, шығыс қуатын орнатуға және деректер алмасу арналарын орнатуға болады.

IoT жобаларының маңызды компоненттерінің бірі-байланыс құралдары. nrf24101 келесі салаларда сәтті қолданылуы мүмкін:

- мобильді электроника;
- компьютерлер;
- автоматтандырылған жүйелер;
- "ақылды үйдің" әртүрлі элементтері-дабыл, температураны бақылау және басқа да функциялар;
- ойындар;
- тұрмыстық электроника.

Nrf24101 + тақтасына жиілік синтезаторы, демодулятор, күшейткіштер және басқа компоненттер кіреді. Модульдің жұмыс жиілігі арна нөмірімен анықталады, байланыс жүретін жиілік диапазоны, 2,4 – 2,483 ГГц. Арналар 1 МГц арқылы орналасады, яғни 2,4 ГГц жиілігі нөлге, 83 – 2,483 ГГц арнасына сәйкес келеді.

Модульде 4 Жұмыс режимі бар-өшіру (Power Down), ұйқы режимі (Standby), деректерді қабылдау (RX mode), деректерді беру (TX Mode). RX деректерін қабылдау режимінде ток шығыны TX деректерін беру режиміне қарағанда жоғары.

Тұрақты және сенімді деректерді беру және қабылдау үшін Enhanced ShockBurst протоколы жауап береді. Қабылдаушы құрылғы деректерді қабылдау туралы жауап беруі керек, осылайша кері байланысты растайды.

MQ-4 сенсоры жартылай өткізгіш құрылғыларға жатады. Сенсордың жұмыс принципі анықталған газдың молекулаларымен байланысқан кезде жұқа қабатты SnO<sub>2</sub> қалайы диоксиді қабатының кедергісін өзгертуге негізделген. Сенсордың сезімтал элементі Al<sub>2</sub>O<sub>3</sub> қапталған керамикалық түтіктен және оған қолданылатын қалайы диоксидінің сезімтал қабатынан тұрады. Түтіктің ішінде қыздыру элементі өтеді, ол сезімтал қабатты анықталған газға жауап бере бастайтын температураға дейін қыздырады. Әр түрлі газдарға сезімталдыққа сезімтал қабаттағы қоспалардың құрамын өзгерту арқылы қол жеткізіледі.

Сенсор MQ-8 жартылай өткізгіш газ анализаторының негізінде жасалған. Сенсор сутегі құрамына пропорционалды аналогтық сигналды логикалық шығысқа шығарады.

Біз сенсордың қыздыру элементін жеке контактілер тобы арқылы басқаруға мүмкіндік бердік.

Сенсор логикасы мен жылытқыштың жұмысы үшін әртүрлі кернеуді қолдануға болады. Ал тағы — бағдарламалық қосуға және өшіруге болады қыздырғыш айтарлықтай ұзартады автономды жұмыс уақыты құрылғылар.

Газ анализаторына химиялық реакция үшін қажет қыздыру элементі салынған: жұмыс кезінде сенсор ыстық болады — бұл қалыпты жағдай.

Сенсор екі бөліктен тұрады – сыйымдылық температура сенсоры және гигрометр. Біріншісі температураны өлшеу үшін, екіншісі ауаның ылғалдылығы үшін қолданылады. Ішіндегі чип Аналогты-сандық түрлендірулерді орындай алады және микроконтроллер арқылы оқылатын сандық сигнал бере алады.

DHT11 температура мен ылғалдылық сенсоры.

Көп жағдайда DHT11 немесе DHT22 екі нұсқада қол жетімді: металл контактілері бар пластикалық корпус түрінде жеке сенсор ретінде немесе сенсор мен дәнекерленген байланыстырушы элементтері бар дайын модуль ретінде. Екінші нұсқа нақты жобаларда қолдану әлдеқайда оңай және жаңадан бастаушыларға өте ұсынылады. 3.2 суретте Arduino интерфейсі көрсетілген.

```

VT.ino
1 #include <ESPDash.h> // Visualization
2 #include <ESP8266WiFi.h>
3 #include <ESPAsyncTCP.h>
4 #include <ESPAsyncWebServer.h> // Server
5
6 #define AMPER A0
7
8 uint8_t relay_1 = D1;
9 uint8_t relay_2 = D2;
10 uint8_t relay_3 = D3;
11 uint8_t relay_4 = D4;
12 uint8_t relay_5 = D5;
13 uint8_t relay_6 = D6;
14 uint8_t relay_7 = D7;
15 uint8_t relay_8 = D8;
16
17 const char *ssid = "iPhone 13 (2)"; // replace with your wifi ssid and wpa2 key
18 const char *pass = "qwerty123";
19
20 AsyncWebServer server(80);
21 ESPDash dashboard(&server);
22
23 Card sens1(&dashboard, TEMPERATURE_CARD, "Ток күйі: ", "A");
24 Card amperBar(&dashboard, BAR_CARD, "Ампер"); // Bar card for Amper
25
26 Card relay1(&dashboard, BUTTON_CARD, "Relay 1");
27 Card relay2(&dashboard, BUTTON_CARD, "Relay 2");
28 Card relay3(&dashboard, BUTTON_CARD, "Relay 3");
29 Card relay4(&dashboard, BUTTON_CARD, "Relay 4");
30 Card relay5(&dashboard, BUTTON_CARD, "Relay 5");
31 Card relay6(&dashboard, BUTTON_CARD, "Relay 6");
32 Card relay7(&dashboard, BUTTON_CARD, "Relay 7");
33 Card relay8(&dashboard, BUTTON_CARD, "Relay 8");
34
35 // Initialization
Output

```

### 3.2 сурет – Arduino интерфейсі

Бұл код ESP8266 микроконтроллерін қолдану арқылы веб-сервер мен басқару панелін жасау үшін қолданылады. Кодта бірнеше кітапханалар қосылады, веб-сервер орнатылады және ESPDash кітапханасы арқылы визуализация тақтасы жасалады. Енді әрбір бөлікті толық түсіндірейін:

```

#include <ESPDash.h> // Визуализация
#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h> // Сервер

```

Бұл жолдарда қажетті кітапханалар қосылады. ESPDash.h визуализация жасау үшін, ESP8266WiFi.h WiFi байланысын орнату үшін, ESPAsyncTCP.h және ESPAsyncWebServer.h асинхронды веб-сервер жасау үшін қолданылады.

```

#define AMPER A0
uint8_t relay_1 = D1;
uint8_t relay_2 = D2;
uint8_t relay_3 = D3;
uint8_t relay_4 = D4;
uint8_t relay_5 = D5;
uint8_t relay_6 = D6;
uint8_t relay_7 = D7;
uint8_t relay_8 = D8;
const char *ssid = "iPhone 13 (2)";
const char *pass = "qwerty123";

```

Мұнда WiFi желісінің атауы (ssid) және құпия сөз (pass) белгіленеді.

```

AsyncWebServer server(80);
ESPDash dashboard(&server);

```

Бұл жолдарда асинхронды веб-сервер 80-портта орнатылады, және ESPDash тақтасы осы серверге байланысады.

```
Card sens1(&dashboard, TEMPERATURE_CARD, "Тоқ күші: ", "A");
Card amperBar(&dashboard, BAR_CARD, "Ампер"); // Бар картасы Ампер үшін
```

```
Card relay1(&dashboard, BUTTON_CARD, "Relay 1");
Card relay2(&dashboard, BUTTON_CARD, "Relay 2");
Card relay3(&dashboard, BUTTON_CARD, "Relay 3");
Card relay4(&dashboard, BUTTON_CARD, "Relay 4");
Card relay5(&dashboard, BUTTON_CARD, "Relay 5");
Card relay6(&dashboard, BUTTON_CARD, "Relay 6");
Card relay7(&dashboard, BUTTON_CARD, "Relay 7");
Card relay8(&dashboard, BUTTON_CARD, "Relay 8");
```

Бұл жерде әртүрлі виджеттер (карточкалар) жасалады. sens1 виджеті температураны көрсетеді, amperBar виджеті ампер мәнін көрсететін бар графигі болып табылады. Қалған виджеттер релелерді (қосқыштарды) басқару үшін батырмалар болып табылады.

**Инициализация:** Кодтың қалған бөлігі виджеттердің жұмысын және олардың функционалдығын толық анықтайды, бірақ бұл бөлікте толық код көрсетілмеген.

Бұл жалпы түсініктеме біздің кодтың негізгі құрылымын және оның мақсатын түсіндіреді.

Бұл кодты түсіндіру үшін, біз кодтың екі негізгі бөлігін талдаймыз: setup() функциясы және loop() функциясы.

setup() функциясы

Бұл функция микроконтроллер іске қосылған кезде бір рет орындалады. Мұнда негізгі инициализация параметрлері орнатылады.

```
void setup() {
  Serial.begin(115200); // Сериялық байланыс жылдамдығын 115200 бит/с орнату
  delay(2000); // 2 секунд күту
  WiFi.begin(ssid, pass); // WiFi байланысын бастау
  if (WiFi.waitForConnectResult() != WL_CONNECTED) { // WiFi байланысы тексеріледі
    Serial.printf("WiFi Failed!\n"); // Егер байланыс болмаған жағдайда, қате хабарламасы басылады
    return;
  }
  Serial.print("IP Address: "); // IP мекен-жайын басып шығару
  Serial.println(WiFi.localIP()); // Жергілікті IP мекен-жайын басып шығару
  server.begin(); // Веб-серверді бастау
  // Релелердің барлық пиндерін шығыс режиміне орнату
  pinMode(relay_1, OUTPUT);
  pinMode(relay_2, OUTPUT);
  pinMode(relay_3, OUTPUT);
  pinMode(relay_4, OUTPUT);
}
```

```

pinMode(relay_5, OUTPUT);
pinMode(relay_6, OUTPUT);
pinMode(relay_7, OUTPUT);
pinMode(relay_8, OUTPUT);
/*
// Релелерді жоғары күйге қою (қаласак, бұл жолдарды қосуға болады)
digitalWrite(relay_1, HIGH);
digitalWrite(relay_2, HIGH);
digitalWrite(relay_3, HIGH);
digitalWrite(relay_4, HIGH);
digitalWrite(relay_5, HIGH);
digitalWrite(relay_6, HIGH);
digitalWrite(relay_7, HIGH);
digitalWrite(relay_8, HIGH);
*/
}

```

loop() функциясы

Бұл функция микроконтроллер іске қосылған кезде қайта-қайта орындалады. Мұнда негізгі басқару логикасы жүзеге асырылады.

```

void loop() {
float amper = analogRead(AMPER); // Ампер мәнін оқу (аналогтық оқу)
Serial.print("Ampper: "); // Ампер мәнін басып шығару
Serial.println(amper);
sens1.update(amper, "Ампер"); // Ампер мәнін сенсор картасына жаңарту
amperBar.update(amper); // Ампер мәнін бар графигіне жаңарту
// Реле 1 батырмасының басқару функциясы
relay1.attachCallback([&](bool value){
Serial.println("[relay1] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_1, (value));
relay1.update(value);
dashboard.sendUpdates();
});
// Реле 2 батырмасының басқару функциясы
relay2.attachCallback([&](bool value){
Serial.println("[relay2] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_2, (value));
relay2.update(value);
dashboard.sendUpdates();
});
// Реле 3 батырмасының басқару функциясы
relay3.attachCallback([&](bool value){
Serial.println("[relay3] Button Callback Triggered:

```

```

"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_3, (value));
relay3.update(value);
dashboard.sendUpdates();
});
// Реле 4 батырмасының басқару функциясы
relay4.attachCallback([&](bool value){
Serial.println("[relay4] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_4, (value));
relay4.update(value);
dashboard.sendUpdates();
});
// Реле 5 батырмасының басқару функциясы
relay5.attachCallback([&](bool value){
Serial.println("[relay5] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_5, (value));
relay5.update(value);
dashboard.sendUpdates();
});
// Реле 6 батырмасының басқару функциясы
relay6.attachCallback([&](bool value){
Serial.println("[relay6] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_6, (value));
relay6.update(value);
dashboard.sendUpdates();
});
// Реле 7 батырмасының басқару функциясы
relay7.attachCallback([&](bool value){
Serial.println("[relay7] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_7, (value));
relay7.update(value);
dashboard.sendUpdates();
});
// Реле 8 батырмасының басқару функциясы
relay8.attachCallback([&](bool value){
Serial.println("[relay8] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_8, (value));
relay8.update(value);
dashboard.sendUpdates();

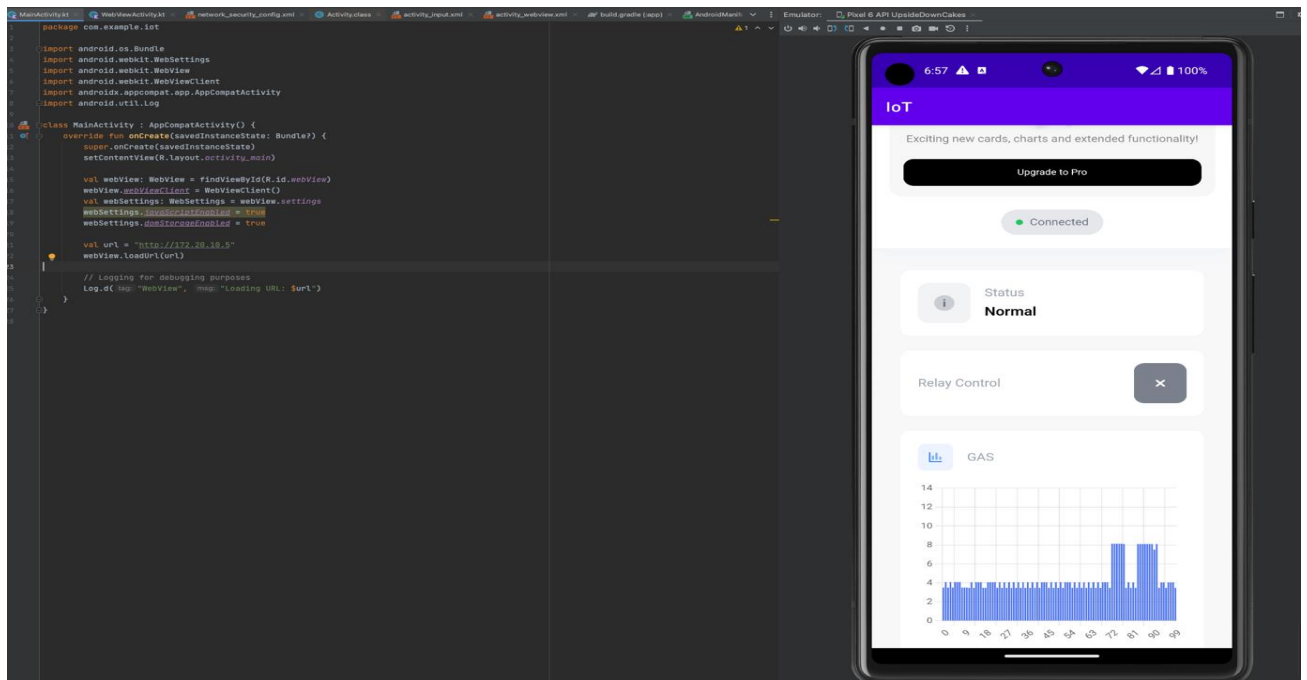
```

```
});
// Жаңартуларды веб-серверге жіберу
dashboard.sendUpdates();
delay(1000); // 1 секунд күту
}
```

Бұл функцияның негізгі қызметі:

- ампер мәнін аналогтық пиннен оқу және оны жаңарту;
- реле батырмаларына арналған кері шақырулар (callbacks) орнату, олар басылған кезде реле күйін өзгертіп, жаңартуларды веб-серверге жібереді;
- жаңартуларды үнемі веб-серверге жіберу;

Осылайша, бұл код ESP8266 негізінде жасалған веб-басқарылатын реле жүйесін іске қосады, ол интернет арқылы басқарылып, көрсеткіштерді визуализациялайды.



3.3 сурет – IoT қосымшасының терезесі

3.3 суретте Android-те жұмыс істейтін IoT қосымшасының терезесі көрсетілген. Міне, қосымша интерфейсінің сипаттамасы.

Қосымша тақырыбы:

Экранның жоғарғы жағында фиолет түсті фонда "IoT" қосымшасының атауы көрсетілген.

Оң жақ жоғарғы бұрышта мәзірге көшу түймесі бар (тік үш нүкте).

Қосылу күйі:

Тақырыптың астында қосылу күйінің индикаторы көрсетіледі, бұл жағдайда құрылғы "Connected" (қосылған) деп көрсетілген.

Күй карточкасы:

Төменде күй карточкасы бар, ол бұл жағдайда "Status: Normal" (Күйі: Қалыпты) деп көрсетіледі және жасыл белгі қалыпты жағдайды білдіреді.

Реледі басқару:

Әрі қарай, "Relay Control" (Реледі басқару) түймесі бар карточка орналасқан. Түймеде крестик белгісі бар, бұл белсенді емес күйді немесе жабу мүмкіндігін білдіруі мүмкін.

Өлшемдер графигі:

Экранның төменгі жағында "GAS" (Газ) деп жазылған график көрсетіледі, ол датчиктен алынған аналогтық сигнал мәндерін көрсетеді. График көк бағандар түрінде уақыт бойынша газ деңгейін көрсетеді.

Негізгі бағдарлама коды

WiFi баптаулары:

Бағдарлама берілген SSID және парольмен WiFi желісіне қосылады.

Веб-сервер мен басқару тақтасын инициализациялау:

80-портта веб-сервер құрылады.

ESP-DASH басқару тақтасы веб-серверге қосылады.

Аналогтық пиннен деректерді оқу:

Деректер A0 аналогтық пиннен оқылады, мұнда газ датчигі қосылған.

Релені басқару:

D2 пині релені басқару үшін қолданылады. Реле басқару түймесінің күйіне және газ датчигінің сигнал деңгейіне байланысты қосылады және ажыратылады.

Интерфейсті жаңарту:

Интерфейс ағымдағы мәндерді және жүйе күйін көрсетіп, нақты уақытта жаңартылады.

Бұл интерфейс пайдаланушыға газ датчигінің күйін бақылауға және құрылғыны қосу немесе өшіру үшін релені қолмен басқаруға мүмкіндік береді.

Суреттегі құрылғы туралы түсініктеме.

Суретте газ датчигі бар ESP8266 негізінде жасалған IoT құрылғының жинағы көрсетілген. Бұл құрылғы газ деңгейін өлшеуге және релені басқаруға арналған. Төменде әрбір компоненттің түсіндірмесі берілген:

ESP8266 Модулі:

Суреттің төменгі жағында орналасқан.

Бұл модуль Wi-Fi қосылымын қамтамасыз етеді және микроконтроллердің рөлін атқарады.

Ол аналогтық сигналдарды оқып, релені басқару үшін пайдаланылатын негізгі процессор.

Газ датчигі (MQ-2):

Ортада орналасқан, ESP8266 модулінің үстінде.

Бұл датчик газдың деңгейін өлшейді, мысалы, түтін немесе басқа жанғыш газдар.

Датчиктен алынған деректер аналогтық сигнал ретінде ESP8266 модуліне беріледі.

Реле модулі:

Сол жақ жоғарғы бұрышта орналасқан.

Реле модулі жоғары ток немесе кернеу жүктемелерін басқаруға арналған.

Бұл модуль суды шашу жүйесін басқару үшін пайдаланылуы мүмкін, мысалы, өрт анықталған кезде.

Қуат көзі:

Оң жақта орналасқан.

Бұл модуль құрылғының барлық компоненттеріне қажетті тұрақты кернеуді қамтамасыз етеді.

Қуат көзі 220В айнымалы токты 5В тұрақты токқа түрлендіреді.

Қосқыш және сымдар:

Компоненттер бір-біріне сымдар арқылы қосылған.

Әрбір сым белгілі бір компоненттер арасында сигналдар мен қуатты тасымалдауға арналған.

Құрылғының жұмысы

Бұл құрылғы газдың деңгейін өлшеуге және релені басқаруға арналған IoT жүйесі ретінде жұмыс істейді. ESP8266 модулі Wi-Fi арқылы интернетке қосылып, деректерді қашықтықтан бақылауға мүмкіндік береді. Мысалы, газдың деңгейі белгілі бір шектен асып кетсе, реле арқылы су шашу жүйесі автоматты түрде іске қосылады.

Бағдарлама кодының түсіндірмесі.

Бағдарлама кодының негізгі бөліктері:

Wi-Fi-ға қосылу:

ESP8266 Wi-Fi желісіне қосылып, интернетке шығады.

Аналогтық пиннен деректерді оқу:

Газ датчигінен аналогтық сигналды оқып, оның мәнін анықтайды.

Релені басқару:

Газ деңгейі белгілі бір шектен асқан кезде релені қосып, газ деңгейі қалыпты жағдайға түскенде релені өшіреді.

ESP-DASH арқылы деректерді бақылау:

Веб-интерфейс арқылы құрылғының күйін бақылауға және релені қолмен басқаруға мүмкіндік береді.

Бұл құрылғы газ деңгейін қадағалауға және қауіпті жағдайларда релені қосып/өшіруге мүмкіндік беретін толыққанды IoT жүйесі болып табылады.

## ҚОРЫТЫНДЫ

Дипломдық жобада кондитерлік фабриканың технологиялық процестерін автоматтандыру жүйелерін зерттеуді дамытуымыз керек. Жұмыстың құрамында өнімнің техникалық мүмкіндіктерін толық пайдалануды қамтамасыз ету үшін қажетті техникалық мәліметтер мен басқа да мәліметтер бар. Ол үшін біз Arduino бағдарламасына негізделген wemos тақтасын алдық. Код python бағдарламалау тілінде жазылған.

Arduino-да жобаны құру 3 негізгі кезеңнен тұрады: кодты жазу, прототиптеу және микробағдарлама. Дипломдық жұмыста мен IoT қолданған кезде қазандықты басқаруды автоматтандыруды жасауым керек. Ол үшін мен wemos D1 mini arduino nano, MT4, mq 8, dht11 микроконтроллерін қосу арқылы макет жасадым.

Кішкентай көрінетін wemos D1 Mini тақтасы әуесқойларға немесе кәсіби әзірлеушілерге өздерінің жобаларын "нөлден" құруға мүмкіндік беретін әсерлі мүмкіндіктерге ие. ESP8266 микроконтроллерлік платформасы ескірмейтін Ардуиноның толық алмастырушысы бола алады, ал ол өнімділік пен функционалдылықтағы тікелей әріптесінен әлдеқайда жоғары. Қуатты 32 биттік процессор сізге есептеулерді немесе деректерді өңдеуді тезірек орындауға мүмкіндік береді, ал WiFi сымсыз байланысы арқылы басқа электрондық құрылғылармен байланысу мүмкіндігі қарапайым немесе күрделі ақылды құрылғыларды жобалау мен жобалауда кең мүмкіндіктер ашады.

Жобада Arduino жабдықтауды және пайтон жабдықтау тілін қолданып, макет ұсынылды. Макет стандарттарға сай және жұмыс жасайды.

### Қолданылған әдебиет тізімі

1. Бердышев, в.ф. газдар мен суды тазартудың технологиялық процестерін автоматтандыру негіздері: дәрістер курсы / в. ф. Бердышев, К. С. Шатохин. - М.: МИСиС, 2016. - 136 с.
2. Бородин, и. Ф. технологиялық процестерді автоматтандыру және автоматты басқару жүйелері: орта кәсіптік білімге арналған оқулық / и. Ф. Бородин, С.А. Андреев. — 2-ші басылым., испр. және қосымша-Мәскеу: Юрайт баспасы, 2018. — 386 Б.
3. Дастин, Э. бағдарламалық жасақтаманы тестілеу. Енгізу, басқару және автоматтандыру / Э. Дастин, Д. Рэшка, Д. Пол; ағылшын тілінен ауд. М. Павлов. - М.: Лори, 2019. - 567 с.
4. Евтушенко, С.и. құрылысты Автоматтандыру және роботтандыру: оқу құралы / С. и. Евтушенко, А. Г. Булгаков, В. А. Воробьев, Д. я. Паршин. -М.: ИЦ РИОР, ФЗО ИНФРА-М, 2017. - 452 с.
5. Кангин, в. в. технологиялық процестерді автоматтандыру жүйелеріндегі өнеркәсіптік контроллерлер: оқу құралы / В. В. Кангин. - Ст. Оскол: ТНТ, 2016. - 408 с.
6. Пантелеев, В. Н. өндірісті автоматтандыру негіздері: бастауыш кәсіптік білім беру мекемелеріне арналған оқулық / В.Н. Пантелеев, В. М. Прошин. - М.: ИЦ Академия, 2018. - 208 с.

7. Шаян тәрізділер, М. Ю. өндірісті автоматтандыру: орта кәсіптік білімге арналған оқулық / М. Ю. шаян тәрізділер. — 2-ші басылым., испр. және қосымша-Мәскеу: Юрайт баспасы, 2018. — 180 б.

8. Скворцов, а.в. өнімнің өмірлік циклін басқаруды автоматтандыру: жоғары кәсіптік білім беру мекемелерінің студенттеріне арналған оқулық / А. В. Скворцов, А. Г. Ширтладзе, Д. А. Чмырь. - М.: ИЦ Академия, 2016. - 320 с.

9. Машина жасаудағы өндірістік процестерді автоматтандыру: оқулық / А. Г. Ширтладзе, В.Н. Воронов, в. п. Борискин. - Ст. Оскол: ТНТ, 2017. - 600 с.

10. Технологиялық процестерді автоматтандыру: Оқу құралы / А. Г. Ширтладзе, С. В.Бочкарев, а. Н. Лыков. - Ст. Оскол: ТНТ, 2018. - 524 с.

11. Фельдштейн, е. е. Машина жасаудағы өндірістік процестерді автоматтандыру / Е.Э. Фельдштейн, М. А. Корниевич. - Вологда: Инфра-Инженерия, 2019. - 264 с.

12. Храменков, в.г. мұнай-газ ұңғымаларын бұрғылаудың технологиялық процестерін басқаруды автоматтандыру: академиялық бакалавриатқа арналған оқу құралы / В. Г. Храменков. - Люберцы: Юрайт, 2020. - 415 с.

13. Скворцов, а.в. өнімнің өмірлік циклін басқаруды автоматтандыру: жоғары кәсіптік білім беру мекемелерінің студенттеріне арналған оқулық / А. В. Скворцов, А. Г. Ширтладзе, Д. А. Чмырь. - М.: ИЦ Академия, 2021. - 320 с.

## **А ҚОСЫМШАСЫ** (Программаның листингі)

```
#include <ESPDash.h> // Визуализация
#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h> // Сервер
#define AMPER A0
uint8_t relay_1 = D1;
uint8_t relay_2 = D2;
uint8_t relay_3 = D3;
uint8_t relay_4 = D4;
uint8_t relay_5 = D5;
uint8_t relay_6 = D6;
uint8_t relay_7 = D7;
uint8_t relay_8 = D8;
const char *ssid = "iPhone 13 (2)";
const char *pass = "qwerty123";
Мұнда WiFi желісінің атауы (ssid) және құпия сөз (pass) белгіленеді.
AsyncWebServer server(80);
ESPDash dashboard(&server);
Card sens1(&dashboard, TEMPERATURE_CARD, "Тоқ күші: ", "A");
```

```

Card amperBar(&dashboard, BAR_CARD, "Ампер"); // Бар картасы Ампер
үшін
Card relay1(&dashboard, BUTTON_CARD, "Relay 1");
Card relay2(&dashboard, BUTTON_CARD, "Relay 2");
Card relay3(&dashboard, BUTTON_CARD, "Relay 3");
Card relay4(&dashboard, BUTTON_CARD, "Relay 4");
Card relay5(&dashboard, BUTTON_CARD, "Relay 5");
Card relay6(&dashboard, BUTTON_CARD, "Relay 6");
Card relay7(&dashboard, BUTTON_CARD, "Relay 7");
Card relay8(&dashboard, BUTTON_CARD, "Relay 8");
void setup() {
Serial.begin(115200); // Сериялық байланыс жылдамдығын 115200 бит/с
орнату
delay(2000); // 2 секунд күту
WiFi.begin(ssid, pass); // WiFi байланысын бастау
if (WiFi.waitForConnectResult() != WL_CONNECTED) { // WiFi байланысы
тексеріледі
Serial.printf("WiFi Failed!\n"); // Егер байланыс болмаған жағдайда, қате
хабарламасы басылады

```

#### А Қосымшасының жалғасы

```

return;
}
Serial.print("IP Address: "); // IP мекен-жайын басып шығару
Serial.println(WiFi.localIP()); // Жергілікті IP мекен-жайын басып шығару
server.begin(); // Веб-серверді бастау
// Релелердің барлық пиндерін шығыс режиміне орнату
pinMode(relay_1, OUTPUT);
pinMode(relay_2, OUTPUT);
pinMode(relay_3, OUTPUT);
pinMode(relay_4, OUTPUT);
pinMode(relay_5, OUTPUT);
pinMode(relay_6, OUTPUT);
pinMode(relay_7, OUTPUT);
pinMode(relay_8, OUTPUT);
/*
digitalWrite(relay_1, HIGH);
digitalWrite(relay_2, HIGH);
digitalWrite(relay_3, HIGH);
digitalWrite(relay_4, HIGH);
digitalWrite(relay_5, HIGH);

```

```

digitalWrite(relay_6, HIGH);
digitalWrite(relay_7, HIGH);
digitalWrite(relay_8, HIGH);
*/
}
relay1.attachCallback([&](bool value){
Serial.println("[relay1] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_1, (value));
relay1.update(value);
dashboard.sendUpdates();
});
relay2.attachCallback([&](bool value){
Serial.println("[relay2] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_2, (value));
relay2.update(value);
dashboard.sendUpdates();
});

```

#### А ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```

relay3.attachCallback([&](bool value){
Serial.println("[relay3] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_3, (value));
relay3.update(value);
dashboard.sendUpdates();
});
relay4.attachCallback([&](bool value){
Serial.println("[relay4] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_4, (value));
relay4.update(value);
dashboard.sendUpdates();
});
relay5.attachCallback([&](bool value){
Serial.println("[relay5] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_5, (value));
relay5.update(value);
dashboard.sendUpdates();
});

```

```

});
// Реле 6 батырмасының басқару функциясы
relay6.attachCallback([&](bool value){
Serial.println("[relay6] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_6, (value));
relay6.update(value);
dashboard.sendUpdates();
});
// Реле 7 батырмасының басқару функциясы
relay7.attachCallback([&](bool value){
Serial.println("[relay7] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_7, (value));
relay7.update(value);
dashboard.sendUpdates();
});
// Реле 8 батырмасының басқару функциясы
relay8.attachCallback([&](bool value){

```

#### А Қосымшасының жалғасы

```

Serial.println("[relay8] Button Callback Triggered:
"+String((value)?"HIGH":"LOW"));
digitalWrite(relay_8, (value));
relay8.update(value);
dashboard.sendUpdates();
});
// Жаңартуларды веб-серверге жіберу
dashboard.sendUpdates();
delay(1000); // 1 секунд күту
}

```