

**«Astana IT University» ЖШС колледжі**

**Есеп**

**Өндірістік оқыту бойынша**

ПМ06. Микроконтроллер негізінде сандық құрылғыларды бағдарламалау

Тақырыбы: «Құрылғыларды дауыстық басқару»

Тобы: ПО2215

Студенттің аты-жөні:

Қорабай Ж.Ж.

Пән оқытушысы:

Сарбасова К.К.

Астана, 2025 ж.

## МАЗМҰНЫ

<b>Кіріспе</b>	<b>3</b>
<b>1 Voxie.AI – дыбыстық басқару жүйесінің теориялық негіздемесі</b>	<b>5</b>
1.1 Тұрмыстық техниканы басқарудың өзекті мәселелері	5
1.2 Заманауи автоматтандыру жүйелеріндегі дауыспен басқарудың артықшылықтары	7
1.3 Дауысты басқарудың қолданыстағы технологиялары мен шешімдеріне талдау	9
<b>2 Voxie.AI жүйесінің құрылымына шолу</b>	<b>14</b>
2.1 Voxie.AI жобасының концепциясы мен архитектурасы	14
2.2 Қолданылған компоненттерге сипаттама	15
2.3 Жобаны жүзеге асыруға арналған бағдарламалық қамтамасыз ету	17
2.4 Жүйені тестілеу: дауыс пәрменін тану және жауап беру дәлдігін тексеру	20
2.5 Жобаның болашақ перспективаларына шолу	20
<b>Қорытынды</b>	<b>22</b>
<b>Қолданылған әдебиеттер тізімі</b>	<b>23</b>
<b>Қосымша 1</b>	<b>24</b>
<b>Қосымша 2</b>	<b>30</b>

## **КІРІПСЕ**

Voxie.AI - жобасы смарт құрылғыны басқару технологияларын зерттеу және дауыстық командаларды өңдеу әдістерін қолдану аясында жүзеге асырылды.

Жұмыстың мақсаты дауыстық командаларды тиімді тани алатын және алынған ақпарат негізінде белгіленген операцияларды орындай алатын жүйенің прототипін жасау болды.

Жобаны әзірлеу барысында бірнеше негізгі технологиялар мен компоненттер пайдаланылды. Жүйенің негізі Arduino микроконтроллері болды, ол сыртқы құрылғыларды басқаруды қамтамасыз етті, мысалы, құлыптарды ашу және жабу үшін сервожетектер, артқы жарықтандыруға арналған жарықдиодты жолақтар және басқа да тұрмыстық элементтер. Дауыстық пәрмендерді тану үшін дыбыстық сигналдарды өңдеуге арналған қосымша кітапхана пайдаланылды, бұл жүйемен дауыстық өзара әрекеттесуді ұйымдастыруға мүмкіндік берді.

Дамудың маңызды кезеңдерінің бірі дыбыс сигналдарын электрлік сигналдарға түрлендіретін микрофонды қосу болды. Бұл жүйенің командаларды қабылдауға және оларға жауап беруге мүмкіндік берді. Жоба сонымен қатар жүйенің әртүрлі күйлерін – құрылғыны қосудан бастап әртүрлі жұмыс режимдерін белсендіруге дейін визуализациялауға мүмкіндік беретін жарық диодты басқару модулін біріктірді.

Жұмыстың маңызды бөлігі жүйені тестілеу және оңтайландыру болды. Тестілеу барысында микрофонның сезімталдығын, команданы тану дәлдігін және әртүрлі сыртқы әсерлерге жүйенің жауаптарын реттеу үшін көптеген эксперименттер жүргізілді. Әртүрлі конфигурация опциялары талданды және алгоритмдерге түзетулер енгізілді, бұл жүйенің жоғары дәлдігін қамтамасыз етті.

Сонымен қатар жұмыс барысында смарт құрылғыларды дауыспен басқару саласындағы қолданыстағы шешімдерге талдау жүргізілді. Жүргізілген зерттеулердің негізінде жүйенің өнімділігі мен тұрақтылығының қажетті деңгейіне

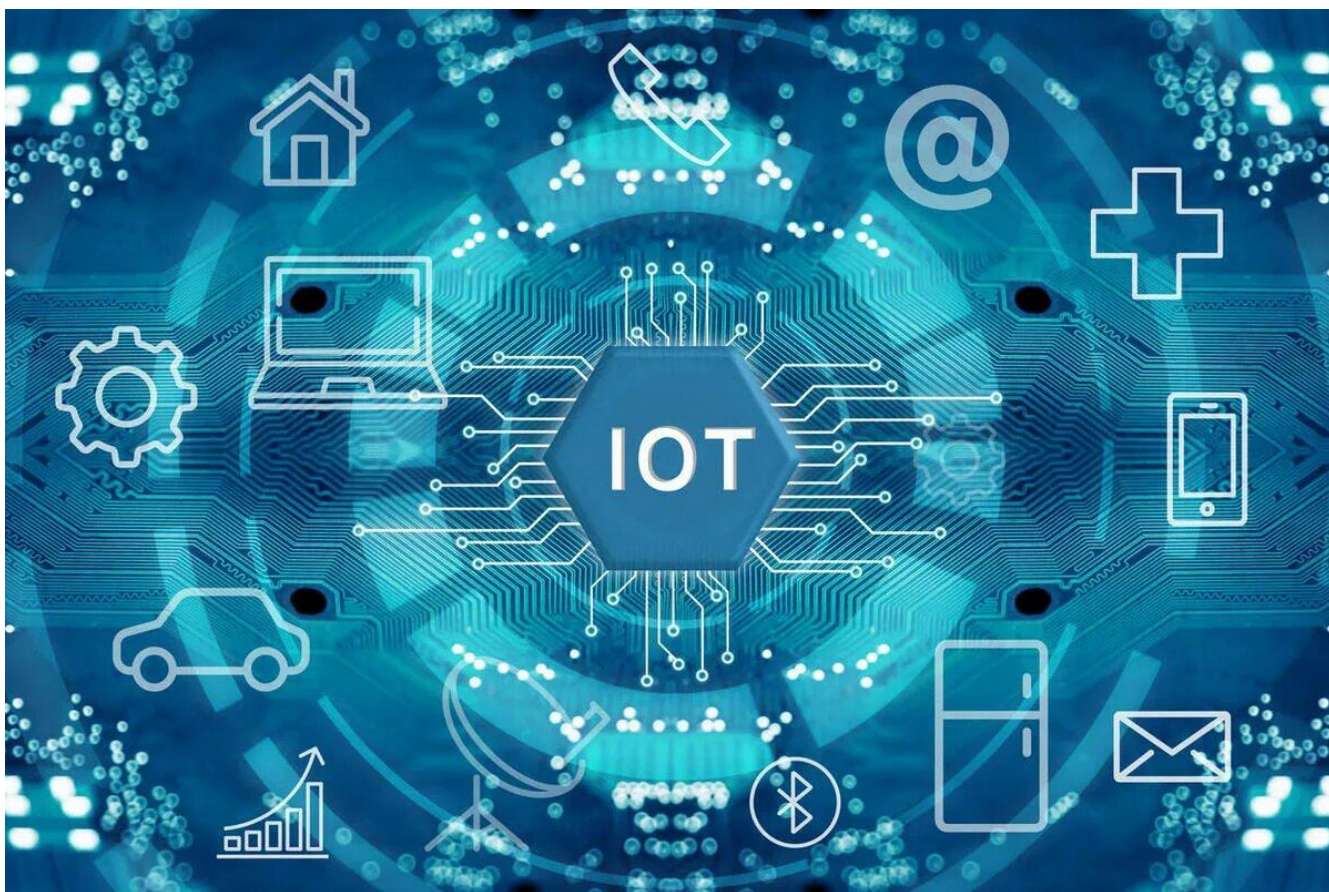
қол жеткізуге мүмкіндік беретін ең тиімді және қолжетімді әдістер таңдалды.

Жоба тұрмыстық құрылғыларды басқару үшін дауыстық интерфейсті пайдаланудың жоғары тиімділігін көрсетті, үйдегі процестерді автоматтандырудың жаңа мүмкіндіктерін ашты. Жобаның одан әрі дамуы басқа смарт құрылғылармен интеграцияны және қосымшалардың одан да кең ауқымын қамтамасыз ету үшін жүйенің функционалдығын кеңейтуді қамтыды.

# 1. VOXIE.AI – ДЫБЫСТЫҚ БАСҚАРУ ЖҮЙЕСІНІҢ ТЕОРИЯЛЫҚ НЕГІЗДЕМЕСІ

## 1.1 Тұрмыстық техниканы басқарудың өзекті мәселелері

Соңғы жылдары Интернет заттары (IoT) технологиялары мен смарт құрылғылардың дамуымен тұрмыстық техниканы басқару айтарлықтай өзгерістерге ұшырады (Сурет 1.1) [1].



Сурет 1.1 – Интернет заттары (IoT)

Дәстүрлі тұрғыда қарастырылса, тұрмыстық техника механикалық түймелермен немесе қарапайым қашықтан басқару пультімен басқарылатын құрылғылар болды. Дегенмен, Amazon Alexa, Google Assistant және Apple Siri сияқты ақылды үй технологиялары мен дауыс көмекшілерінің пайда болуымен тұрмыстық құрылғыларды басқару мәселесі әлдеқайда күрделі болды. Дыбыстық

басқарудың танымал болуына әкелген негізгі факторлардың бірі - олардың ыңғайлылығы. Пайдаланушыға құрылғыдағы қашықтан басқару құралын немесе түймені іздеуден гөрі дауыспен пәрмен беру әлдеқайда оңай. Бұл дәстүрлі құралдарды пайдалануда қиындықтарға тап болуы мүмкін мүмкіндігі шектеулі адамдар үшін өте маңызды. Дауыс интерфейстері технологиямен интуитивті өзара әрекеттесуге, пайдаланушы тәжірибесін жақсартуға және технологияның қолжетімділігін арттыруға мүмкіндік береді.

Тұрмыстық техниканы басқарудың ыңғайлылығы мен қолжетімділігі мәселесі қауіпсіздік пен энергия тиімділігінің маңызды аспектілерін де қозғайды. Ақылды басқару жүйелері әртүрлі құрылғыларды бір желіге біріктіруге және оларды бір интерфейстен басқаруға мүмкіндік береді. Бұл күнделікті тапсырмаларды автоматтандыруға ғана емес, сонымен қатар шығындарды азайтуға және қоршаған ортаны жақсартуға көмектесетін энергия тұтынуды бақылауға мүмкіндік береді.

Дегенмен, барлық артықшылықтарға қарамастан, дауыстық басқару жүйесін енгізу бірқатар мәселелердің пайда болуына әкелді. Ең маңыздыларының бірі - шу және көптеген қоршаған дыбыстар жағдайында команданы тану дәлдігі.

Дауысты басқарудың тиімділігін арттыру үшін бөгде шуды сүзгілеу және сөйлеуді тану дәлдігін арттыру мәселелерін шешу қажет. Бұл сонымен қатар тұрақты алгоритм жаңартуларын және аппараттық құралдарды жақсартуды талап етеді.

Сонымен қатар деректерді беру қауіпсіздігі маңызды аспект болып табылады. Смарт құрылғыларды басқару көбінесе пайдаланушылардың жеке ақпаратын жинауды және өңдеуді, деректерді қорғау мен құпиялылыққа қатысты алаңдаушылықты тудырады. Осыған байланысты ақпараттық қауіпсіздік мәселелері, сондай-ақ дауыстық интерфейстердің сенімділігі мен тұрақтылығы осындай жүйелерді дамытудың маңызды аспектілеріне айналады.

Осылайша, дауысты басқару саласында қол жеткізілген табыстарға

карамастан, одан әрі назар аударуды және дамытуды талап ететін шешілмеген мәселелер бар. Бұл мәселелер техникалық аспектілерге де, деректер қауіпсіздігін қамтамасыз етуге және пайдаланушы тәжірибесін жақсартуға да қатысты.

## **1.2 Заманауи автоматтандыру жүйелеріндегі дауыспен басқарудың артықшылықтары**

Заманауи автоматтандыру жүйелеріндегі дауыспен басқару күнделікті өмірді айтарлықтай өзгертетін және технологиямен өзара әрекеттесуді табиғи және интуитивті ететін интеллектуалды технологиялардың маңызды бөлігі болып табылады (Сурет 1.2) [2].



Сурет 1.2 – Тұрмыстық дыбыстық басқару

Дауыстық интерфейстермен жабдықталған автоматтандыру жүйелері смарт үйлерде, кеңселерде, автомобильдерде және пайдаланушының ең аз күш-жігерімен процестерді автоматтандыруды қажет ететін басқа жерлерде әртүрлі

құрылғыларды басқару үшін кеңінен қолданылады. Дауыс жүйелерінің негізгі артықшылықтарының бірі олардың қолмен басқаруды қамтамасыз ету мүмкіндігі болып табылады, бұл пайдаланушыға бір уақытта бірнеше әрекеттерді орындау қажет болған көп тапсырмалы жағдайларда өте маңызды. Мысалы, смарт үйде дауысты жарықтандыруды, бөлме температурасын, қауіпсіздікті, тұрмыстық техниканы және басқа құрылғыларды басқару үшін пайдаланылады, бұл өмір сапасын айтарлықтай жақсартыды және күнделікті тапсырмаларды жеңілдетеді. Сол сияқты, автомобиль жүйелерінде дауыспен басқару жүргізушіге жолдан көзін алмай-ақ навигацияны, музыка параметрлерін және қоңырауларды басқаруға мүмкіндік береді.

Мұндай жүйелердің негізгі компоненттерінің бірі сөйлеуді тану технологиясы болып табылады. Бұл жүйеге пайдаланушы пәрмендерін, тіпті олар қате айтылса да немесе шулы ортада да дәл түсіндіруге мүмкіндік береді. Соңғы жылдары машиналық оқыту және жасанды интеллект алгоритмдері сөйлеуді тану дәлдігі мен контекстік түсінуді айтарлықтай жақсартты. Заманауи дауыс жүйелері командалардың кең ауқымын тануға және пайдаланушының айтылуына, акцентіне және тіліне бейімделуге қабілетті. Бұл құрылғылармен өзара әрекеттесуді одан да ыңғайлы және жекелендірілген етеді.

Дегенмен, барлық артықшылықтарға қарамастан, автоматтандыру жүйелеріндегі дауысты басқару бірқатар техникалық мәселелерге тап болады. Олардың бірі - шуға сезімталдық. Қалалық ортада немесе жұмыс істеп тұрған тұрмыстық құрылғылар сияқты басқа шу көздері болған кезде сөйлеуді тану айтарлықтай азайып, пәрмендерді түсіндіруде қателіктер туындауы мүмкін. Бұл мәселені шешу үшін шуды азайту технологиялары мен жетілдірілген сөйлеуді тану алгоритмдері белсенді түрде әзірленуде, бұл дауыс жүйелерінің сапасын жақсартуға көмектеседі.

Тағы бір маңызды мәселе - қауіпсіздік пен құпиялылық. Дауыстық көмекшілер пайдаланушы деректерін жинайды және өңдейді, ақпараттың ағып



кетуіне және рұқсатсыз кіруге қатысты алаңдаушылық тудырады. Қауіпсіздіктің жоғары деңгейін қамтамасыз ету үшін дауыс жүйесін әзірлеушілер әртүрлі деректерді шифрлау әдістерін пайдаланады, сонымен қатар екі факторлы аутентификация және дауысты тану сияқты қосымша қауіпсіздік тетіктерін біріктіреді.

Заманауи автоматтандыру жүйелеріндегі дауысты басқару пайдаланушының технологиямен өзара әрекеттесуін айтарлықтай жақсартатын және жайлылық пен қауіпсіздік деңгейін арттыратын қуатты құрал болып табылады. Сонымен бірге оның дамуы сөзді тану технологияларын, сондай-ақ пайдаланушылар үшін одан да ыңғайлы және қауіпсіз жүйелерді құруға мүмкіндік беретін шу және деректерді қорғау мәселелерін шешуді үнемі жетілдіруді талап етеді.

### **1.3 Дауысты басқарудың қолданыстағы технологиялары мен шешімдеріне талдау**

Соңғы жылдары дауыспен басқару көптеген заманауи технологиялар мен автоматтандыру шешімдерінің ажырамас бөлігіне айналды. Сөйлеуді тану технологияларының, жасанды интеллекттің және машиналық оқытудың дамуы пайдаланушының дауыстық командалары негізінде әртүрлі тапсырмаларды орындауға қабілетті құрылғылар мен жүйелерді жасауға мүмкіндік берді. Қолданыстағы дауыспен басқару шешімдері өмірдің әртүрлі салаларында, смарт үйлерден автомобильдерге және жеке көмекшілерге дейін қолданылады, бұл технологиямен өзара әрекеттесуді айтарлықтай жеңілдетеді.

Ең танымал және кеңінен қолданылатын шешімдердің бірі - Amazon Alexa, Google Assistant және Apple Siri сияқты дауыстық көмекші (Сурет 1.3) [3]. Бұл жүйелер пайдаланушыларға үйіндегі смарт құрылғыларды басқаруға, интернетте ақпарат іздеуге, музыканы басқаруға, қоңырау шалуға немесе хабарлама жіберуге және басқа да көптеген тапсырмаларды орындауға мүмкіндік береді.

Дауыстық пәрмендерді қолдану арқылы пайдаланушылар физикалық түймелерді немесе сенсорлық экрандарды пайдаланбай-ақ жарықтандыруды, бөлме температурасын, тұрмыстық техниканы және т.б. оңай басқара алады.



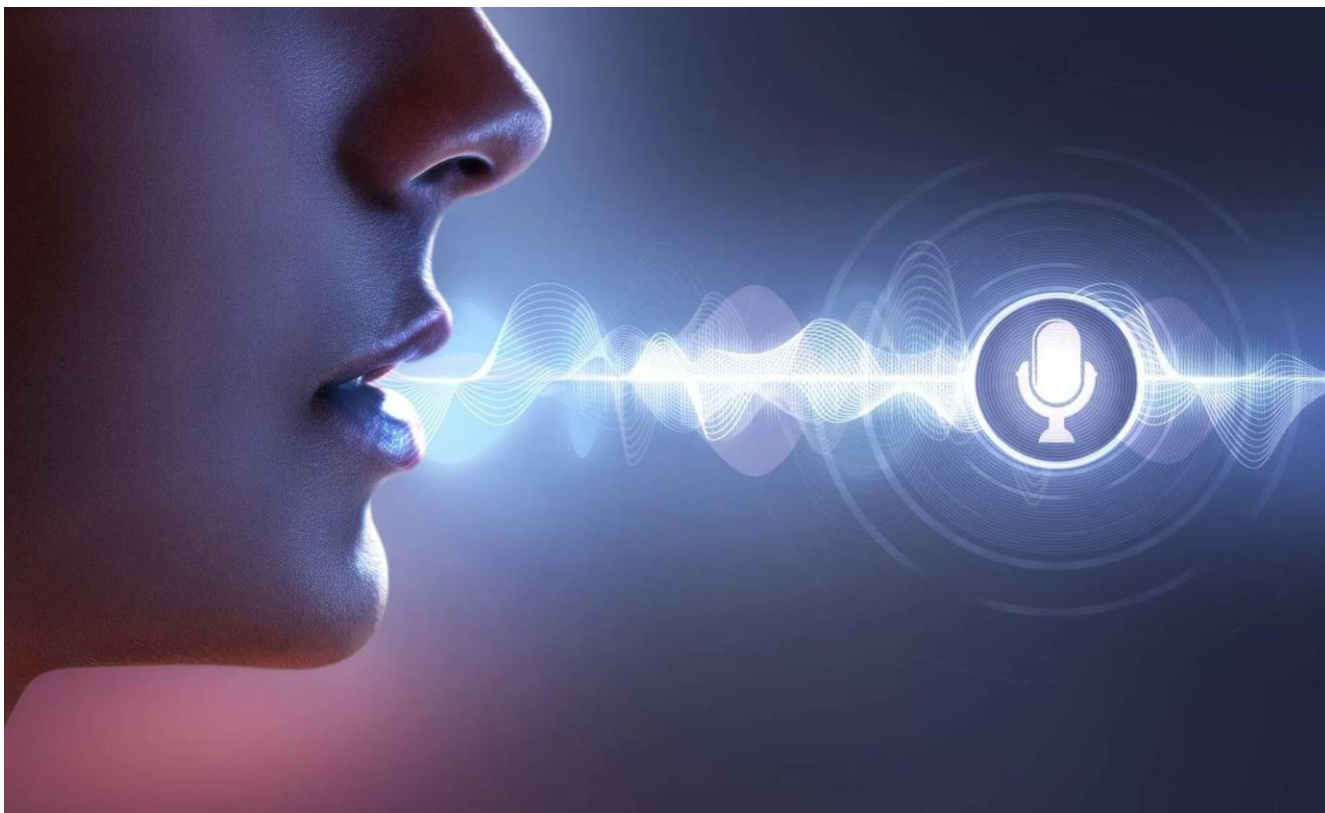
Сурет 1.3 – Танымал заманауи дауыспен басқару құралдары

Бұл жүйелерде қолданылатын негізгі технологиялардың бірі - сөйлеуді тану. Дауыс көмекшілерінің тиімді жұмысы сөйлеуді танудың жоғары дәлдігін және контекстік талдауды орындау мүмкіндігін талап етеді, бұл жүйеге тіпті шулы жағдайларда немесе әртүрлі екпіндерді ескере отырып, сұрауларды дұрыс түсіндіруге мүмкіндік береді.

Заманауи дауыстық интерфейстер деректердің үлкен көлемі бойынша оқытылады, бұл олардың айтылу сипаттамалары мен пайдаланушылардың қалауларына бейімделуге мүмкіндік береді, жүйенің ыңғайлылығы мен дәлдігін арттырады.

Сонымен қатар қазіргі дауыспен басқару технологиялары кездесетін мәселелер бар. Осындай проблемалардың бірі - шуға сезімталдық (Сурет 1.4) [4].

Жоғары фондық шу немесе көп тапсырма жағдайында командаларды тану қиын болуы мүмкін, бұл қателер мен түсінбеушіліктерге әкеледі.



Сурет 1.4 – Дауыспен басқару құрылғысының дыбысқа сезімталдығы

Бұл мәселені шешу үшін соңғы жылдары дыбысты тану сапасын жақсартатын және дауыстық командалардың дәлдігін арттыратын шуды азайту технологиялары белсенді түрде әзірленді. Деректер қауіпсіздігі мен құпиялылық дауыстық жүйелерді әзірлеу кезінде назар аударуды қажет ететін маңызды аспектілер болып табылады. Дауыстық көмекшілер мен құрылғылар жиі пайдаланушылардан жеке деректерді жинайды, бұл құпия ақпараттың ағып кету қаупін тудырады. Деректерді қорғау үшін әртүрлі шифрлау және аутентификация жүйелері, соның ішінде қауіпсіздік деңгейін арттыруға көмектесетін дауысты тану сияқты биометриялық әдістер әзірленуде (Сурет 1.5) [5].

Үлкен коммерциялық шешімдерден басқа, әртүрлі салаларға арналған көптеген мамандандырылған технологиялар мен жүйелер бар. Мысалы,

медицинада дыбыстық басқару жүйелері дәрігерлерге диагностика және емдеу процесінде көмек көрсету үшін, ал өнеркәсіпте - өндірісте жабдықты басқару үшін қолданылады. Бұл шешімдер өнімділікті арттырып қана қоймайды, сонымен қатар ыңғайлылық пен қауіпсіздікті қамтамасыз етеді.



Сурет 1.5 – Дыбыстық басқарудағы биометрия

Дауысты басқару саласындағы қолданыстағы технологиялар мен шешімдерді талдау бұл саланың үнемі дамып келе жатқанын, пайдаланушыларға технологиямен өзара әрекеттесу үшін жаңа мүмкіндіктерді ұсынатынын көрсетеді. Шу сезімталдығы және деректер қауіпсіздігі сияқты кейбір қиындықтарға қарамастан, технология жақсаруды жалғастыруда, бұл өмірдің әртүрлі салаларында дауыстық интерфейстердің одан әрі өсуін және таралуын күтуге мүмкіндік береді.

Бұл тұрғыда Voxie.AI жобасы тұрмыстық техниканы дауыспен басқару жүйесін құруға бағытталған перспективалы шешім болып табылады.

Жетілдірілген сөйлеуді тану және жасанды интеллект алгоритмдерін пайдалана отырып, Voxie.AI шамдар, құлыптар және басқа смарт үй элементтері сияқты құрылғыларды дәл және ыңғайлы басқаруды қамтамасыз етеді. Жүйе пайдаланушыға дауыстық пәрмендерді пайдаланып жабдықпен өзара әрекеттесу мүмкіндігін береді, бұл тұрмыстық құрылғыларды пайдалануды жеңілдетеді және ыңғайлы етеді. Voxie.AI жобасы күнделікті қолдану үшін қауіпсіз және тиімді шешім жасай отырып, дербестендірілген автоматтандыру жүйелері мен смарт технологияларды дамытудың жаңа мүмкіндіктерін ашады.

## **2. VOXIE.AI ЖҮЙЕСІНІҢ ҚҰРЫЛЫМЫНА ШОЛУ**

### **2.1 Voxie.AI жобасының концепциясы мен архитектурасы**

Voxie.AI жобасы тұрмыстық техниканы дауыспен басқару жүйесін құру мақсатында әзірленген. Жобаның негізгі концепциясы дауыстық командаларды қолдану арқылы құрылғыларды басқаруға инновациялық тәсілді енгізу болып табылады. Voxie.AI жүйесі пайдаланушыға әртүрлі тұрмыстық құрылғыларды олармен физикалық өзара әрекеттесусіз басқару мүмкіндігін беруге бағытталған, бұл жайлылықты айтарлықтай арттырады және күнделікті тапсырмаларды жеңілдетеді.

Voxie.AI жүйесінің архитектурасы тұрақты және тиімді жұмысты қамтамасыз ету үшін бірге тығыз жұмыс істейтін бірнеше негізгі құрамдастарды пайдалануға негізделген. Архитектураның маңызды элементі жүйеге әртүрлі сценарийлерге бейімделуге және командаларды тану дәлдігін жақсартуға мүмкіндік беретін бұлттық қызметтермен және мәліметтер базасымен интеграция болып табылады.

Жобаның тұжырымдамасын жүзеге асыру үшін сөзді танудың жоғары дәлдігін қамтамасыз ететін жетілдірілген машиналық оқыту және жасанды интеллект алгоритмдері қолданылды. Осы алгоритмдердің негізінде жүйе қабылданған дауыстық командаларды талдауға, оларды түсіндіруге және сәйкес нұсқауларды құрылғыларға жіберуге қабілетті. Маңызды аспект - пайдаланушының қажеттіліктеріне байланысты жүйені теңшеу және жекелеңдіру мүмкіндігі. Мысалы, жүйе уақыт өте келе үйреніп, иесінің нақты пәрмендері мен қалауларына бейімделе алады.

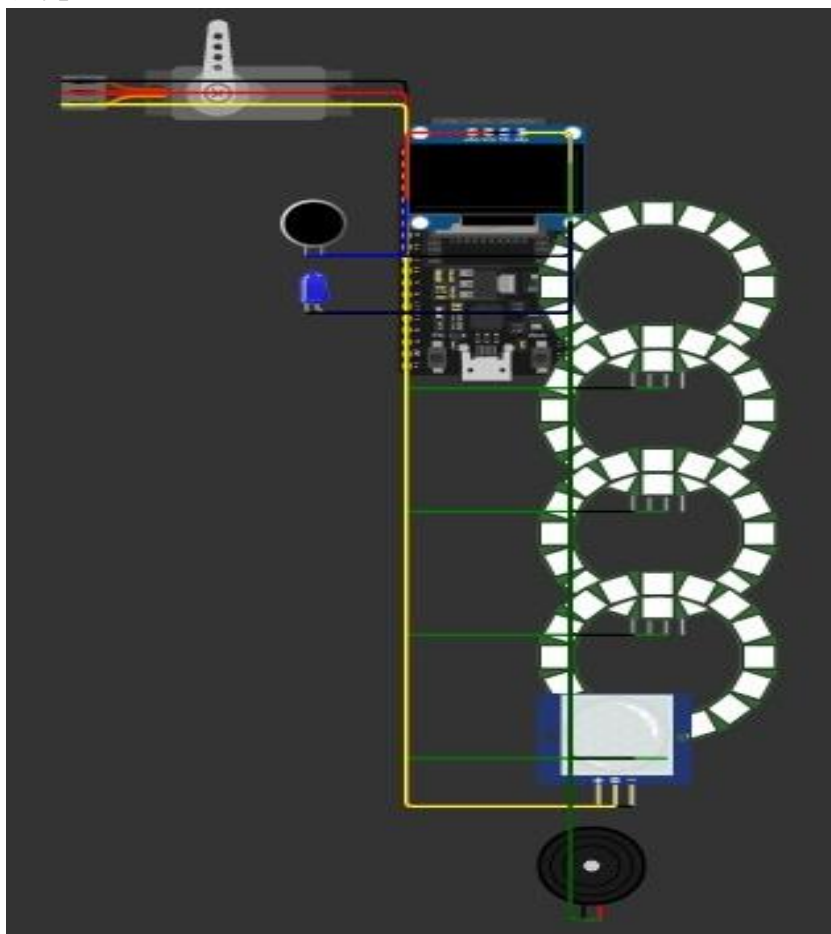
Voxie.AI архитектурасының негізгі мақсаты - пайдаланушыға қарапайым және интуитивті интерфейсті, сондай-ақ сенімділік пен қауіпсіздіктің жоғары дәрежесін қамтамасыз ету. Жоба сонымен қатар жүйені қауіпсіздік талаптарының жоғарылауы жағдайында пайдалануға мүмкіндік беретін деректерді қорғауды



қамтамасыз ететін компоненттерді қамтиды. Voxie.AI жобасы болашақтың смарт үйінің ажырамас бөлігіне айналатын әмбебап және қолжетімді дауыспен басқару жүйесін құруға бағытталған.

## 2.2 Қолданылған компоненттерге сипаттама

Voxie.AI жобасында дауыстық басқару жүйесінің тиімді жұмыс істеуін қамтамасыз ететін әртүрлі аппараттық және бағдарламалық құралдарды пайдаланылды (Сурет 2.1) (Қосымша 1).



Сурет 2.1 – Voxie.AI құрылғысы

Бұл компоненттер бірнеше негізгі топтарға бөлінеді: аппараттық, бағдарламалық және көмекші модульдер.

### Аппараттық құрамдас бөліктер:

1. Arduino микроконтроллері жүйенің негізі болып табылады, құрылғыларды

басқару және деректерді өңдеу үшін қолданылады. Микроконтроллер дауыстық командаларды қабылдайтын, оларды тиісті жетектерге жіберетін және құлыптар, шамдар, кондиционерлер және басқа смарт үй элементтері сияқты қосылған құрылғыларды басқаратын орталық контроллер рөлін атқарады.

2. Микрофон модулі - пайдаланушының дауыстық пәрмендерін түсіруге жауапты. Бұл жүйенің маңызды элементі, кейіннен тану үшін жүйеге дыбысты берудің дәлдігі мен сапасын қамтамасыз етеді. Микрофон әртүрлі жағдайларда пәрмендерді дәл түсіруге мүмкіндік беретін төмен шу деңгейі мен жоғары сезімталдығы үшін таңдалды.
3. Сервожетектер - есіктер, құлыптар, жарықтандыру және басқа элементтер сияқты физикалық құрылғыларды басқару үшін қолданылады. Бұл механизмдер жүйеден келген командаларға жауап береді және есіктерді ашу немесе жабу, жарықтандыру деңгейін өзгерту және т.б. сияқты белгілі әрекеттерді орындайды.
4. Дисплей және пайдаланушы интерфейсі – жүйенің ағымдағы жағдайы туралы ақпаратты көрсету, сонымен қатар жүйенің пайдаланушыға кері байланысын визуализациялау үшін. Дисплейді пайдалану пайдаланушыға команданың сәтті қабылданғаны немесе құрылғының іске қосылғаны сияқты жүйе күйі туралы ақпаратты көруге мүмкіндік береді.

#### **Бағдарламалық қамтамасыз ету компоненттері:**

1. Сөйлеуді тану алгоритмдері дауыстық командаларды өңдеуді қамтамасыз ететін жүйенің маңызды элементі болып табылады. Бұл алгоритмдер дыбыстық сигналдарды мәтінге түрлендіреді, содан кейін оны жүйе талдайды. Voxie.AI жобасы сөзді тану дәлдігін және жүйені пайдаланушының дауыс сипаттамаларына бейімдеу мүмкіндігін қамтамасыз ететін заманауи машиналық оқыту әдістерін пайдаланды.
2. Командаларды өңдеу және құрылғы интерфейсі - сөзді тану жүйесінен



алынған мәтіндік командаларды талдайтын бағдарламалық құрал компоненті. Бұл модуль командаларды интерпретациялауға және сигналдарды сәйкес құрылғыларға жіберуге, пайдаланушының қалауына сәйкес олардың күйін басқаруға жауап береді.

3. Желілік құрамдас бөліктер және бұлттық қызметтер - жүйенің үздіксіз жұмысын және қашықтан басқару мүмкіндігін қамтамасыз ету үшін бұлттық қызметтер пайдаланылды, оның көмегімен пайдаланушы Интернет арқылы құрылғыларды басқара алады. Бұл компоненттер жүйеге жаңа құрылғылар мен жаңартуларға бейімделуге және деректердің қауіпсіз сақталуын қамтамасыз етуге мүмкіндік береді.

#### **Көмекші модульдер:**

1. Қуат және энергиямен қамтамасыз ету - Жүйенің тұрақты жұмысын қамтамасыз ету үшін барлық компоненттерді сенімді электрмен жабдықтауды қамтамасыз ету маңызды. Voxie.AI жобасы жүйенің барлық элементтерінің қауіпсіз жұмысын қамтамасыз ететін асқын кернеу мен қызып кетуден қорғайтын сенімді қуат көздерін пайдаланды.
2. Қауіпсіздік жүйелері - жобада шифрлау және пайдаланушының аутентификациясы сияқты деректерді қорғауды қамтамасыз ететін модульдер де бар. Бұл құрамдас бөліктер рұқсатсыз кіруді болдырмау және пайдаланушы деректерін қорғау үшін жасалған.

Voxie.AI жобасы тұрмыстық техника үшін жоғары тиімді және сенімді дауыспен басқару жүйесін құруға мүмкіндік беретін әртүрлі аппараттық және бағдарламалық қамтамасыз ету компоненттерінің комбинациясын пайдаланады.

### **2.3 Жобаны жүзеге асыруға арналған бағдарламалық қамтамасыз ету**

Voxie.AI жобасын сәтті жүзеге асыру үшін дауыстық басқару жүйесінің толық жұмыс істеуін қамтамасыз ететін бірнеше негізгі бағдарламалық шешімдер таңдалды. Бұл компоненттер микроконтроллерлер үшін төмен деңгейлі

бағдарламалық жасақтаманы әзірлеуді де, сөйлеуді тану мен құрылғыны басқаруға арналған жоғары деңгейлі қолданбаны да қамтиды. Барлық бағдарламалық қамтамасыз ету жүйенің өнімділігіне, сенімділігіне және қауіпсіздігіне қойылатын талаптарды ескере отырып әзірленді.

## **1. Arduino микроконтроллері үшін микробағдарламаны әзірлеу**

Құрылғыларды басқару және сигналдарды өңдеу үшін Arduino микроконтроллері қолданылды. Ол өзінің қарапайымдылығы және әзірлеушілер қауымдастығы тарапынан кең қолдау үшін таңдалды. Arduino жүйенің негізгі логикасын іске асыру үшін пайдаланылды, соның ішінде жетектерді басқару, Wi-Fi желісіне қосылу және дауыстық интерфейстен командаларды өңдеу. Серволар, сенсорлар және микрофон сияқты сыртқы құрылғылармен өзара әрекеттесу үшін тиісті микробағдарлама әзірленді (Қосымша 2). Бұл микробағдарлама мыналарды қамтиды:

- Қозғалыс сенсорлары мен микрофонның кіріс сигналдарын өңдеу.
- Жетектерді басқару (құлыптар, жарықтандыру және басқа құрылғылар).
- Жүйені қашықтан басқару үшін бұлттық қызметтермен деректер алмасу.
- Жүйенің біркелкі жұмысын қамтамасыз ету үшін Wi-Fi қосу және конфигурациялау.

## **2. Сөйлеуді тану алгоритмдер**

Дауыстық командаларды тану үшін машиналық оқытуға негізделген алгоритм пайдаланылды. Жүйе микрофоннан келетін дыбыстық сигналдарды талдайды, оларды сөйлеуді өңдеу алгоритмдері арқылы мәтінге түрлендіреді, содан кейін қабылданған командаларды түсіндіреді

Осы мақсатта нақты уақыттағы аудио өңдеуді және Google Speech API сияқты сөйлеуді тану алгоритмдерімен немесе тұрақты интернет қосылымын қажет етпей командаларды тануды қолдайтын басқа жергілікті шешімдермен интеграцияны қолдайтын кітапхана пайдаланылды. Маңызды аспект - жүйенің дәлдігі мен өнімділігін арттыратын белгілі бір пайдаланушы үшін тану алгоритмін

дәл баптау мүмкіндігі

### **3. Пайдаланушы интерфейсі және бұлттық қызметтермен әрекеттесу**

Пайдаланушымен өзара әрекеттесу және жүйе күйі туралы ақпаратты көрсету, хабарламалар мен күй деректерін көрсету үшін дисплей пайдаланылды. Бағдарлама сонымен қатар жүйені Интернет арқылы басқаруға арналған веб-интерфейсті қамтиды, бұл пайдаланушыға үйден тыс уақытта құрылғыларды басқаруға мүмкіндік береді.

Қашықтан басқаруды жүзеге асыру үшін пәрмендерді беру және жүйеден қауіпсіз арналар арқылы деректерді алу мүмкіндігін қамтамасыз ететін бұлттық қызметтер пайдаланылды. Бұл жоғары қолжетімділікті ғана емес, сонымен қатар құрылғылар арасында деректерді тасымалдау қауіпсіздігін қамтамасыз етті.

### **4. Қауіпсіздік бағдарламалық қамтамасыз ету**

Жобаның маңызды мақсаттарының бірі деректерді қорғау және рұқсатсыз кіруді болдырмау болды. Бұған қол жеткізу үшін жобада әртүрлі қауіпсіздік шаралары жүзеге асырылды, мысалы:

- микроконтроллер мен сервер арасында тасымалдау кезінде деректерді шифрлау.
- Жүйені рұқсатсыз кіруден қорғау үшін аутентификация протоколдарын пайдалану.
- Ықтимал қауіптерді уақтылы анықтау үшін жүйе күйін тіркеу және бақылау.

Қауіпсіздік жүйесін бағдарламалық қамтамасыз етуді енгізу үшін SSL/TLS протоколдарын және басқа деректерді қорғау әдістерін қолдайтын стандартты қауіпсіздік кітапханалары пайдаланылды.

### **5. Басқа құрылғылармен әрекеттесу**

Voxie.AI жүйесін басқа смарт үй құрылғыларымен біріктіруге болады. Осы мақсатта жүйені кеңейтуге және қажет болған жағдайда жаңа құрылғыларды қосуға мүмкіндік беретін Zigbee, Z-Wave және Bluetooth сияқты әртүрлі стандарттармен және протоколдармен өзара әрекеттесу үшін модульдер әзірленді.

Voxie.AI жобасын жүзеге асыруға арналған бағдарламалық қамтамасыз ету микроконтроллермен жұмыс істеуге, дауыстық командаларды өңдеуге, пайдаланушының өзара әрекеттесуіне және тұтастай жүйенің функционалдығы мен сенімділігін қамтамасыз ететін қауіпсіздікке арналған шешімдерді біріктіреді.

## **2.4 Жүйені тестілеу: дауыс пәрменін тану және жауап беру дәлдігін тексеру**

Voxie.AI жүйесінің жұмысын бағалау үшін дауыс пәрменін тану және жауап беру дәлдігі сыналды. Тестілеу процесі бірнеше кезеңді қамтыды. Бірінші кезеңде жүйенің дыбыстық пәрмендерді әртүрлі жағдайларда, соның ішінде шудың болуы мен микрофоннан қашықтығында дұрыс қабылдау қабілеті тексерілді. Осы мақсатта «жарықты қосу», «есікті ашу», «жүйені қаруландыру» және т.б. сияқты алдын ала дайындалған бірнеше командалар қолданылды.

Екінші кезеңде команданың орындалу дәлдігі тексерілді. Әрбір команданың күтілетін нәтижеге сәйкестігі, сондай-ақ түсініксіз немесе қате командалар болған жағдайда жүйенің жауабы тексерілді. Жүйенің жауап беру жылдамдығы да бағаланды. Тестілеу нәтижесінде жүйе аз кедергілермен және қалыпты шу жағдайында қанағаттанарлық жұмысымен жоғары тану дәлдігін көрсететіні анықталды. Тану дәлдігін жақсарту үшін түзетулер мен бейімделулер жасалғанын атап өту маңызды, бұл пайдаланушының тұрақты тәжірибесіне мүмкіндік берді.

## **2.5 Жобаның болашақ перспективаларына шолу**

Voxie.AI жобасының одан әрі дамыту және жақсарту үшін айтарлықтай әлеуеті бар. Болашақта бұл жүйенің функционалдығын тек тұрмыстық техниканы ғана емес, сонымен қатар қауіпсіздік жүйелері, климаттық бақылау, тіпті Интернет заттары (IoT) құрылғылары сияқты басқа да құрылғыларды басқаруға мүмкіндік беру арқылы кеңейту жоспарлануда.

Сондай-ақ әртүрлі шу жағдайларында жауап берудің дәлдігі мен

жылдамдығын арттыру үшін дауыстық командаларды тану алгоритмдерін жетілдіру жоспарлануда. Жобаның келешегі құрылғылардың кең ауқымымен интеграцияны, көп тапсырманы орындауға арналған интерфейсті әзірлеуді және басқа операциялық жүйелермен өзара әрекеттесуді қамтиды. Болашақта пайдаланушының әдеттерін үйренуге және бейімдеуге қабілетті, басқаруды одан да ыңғайлы және интуитивті ететін жасанды интеллектпен жүйені енгізуге болады.

## ҚОРЫТЫНДЫ

Voxie.AI жобасы дауыстық командаларды тану үшін озық технологияларды пайдалана отырып, тұрмыстық құрылғыларды дауыспен басқарудың жоғары тиімділігін көрсетті. Жүйені әзірлеу барысында күнделікті өмірде пайдаланушылардың ыңғайлылығын айтарлықтай жақсартуға мүмкіндік беретін әртүрлі құрылғылармен біріктіру негізі салынды. Енгізілген басқару жүйесі қарапайым және интуитивті интерфейсті қамтамасыз етеді, ол техникалық білімі жоқ адамдар үшін де пайдаланудың қарапайымдылығын қамтамасыз етеді.

Қолданылатын компоненттер мен технологиялар дауыстық интерфейстер үшін маңызды аспект болып табылатын жауап берудің минималды кідірісімен жүйенің тұрақты жұмысын қамтамасыз етті. Өткізілген сынақтар командаларды тану және әрекеттерді уақтылы орындаудың жоғары дәлдігін растады, бұл смарт үйлер мен автоматтандыру саласындағы сәтті шешімдерді енгізудің маңызды факторы болып табылады.

Жобаны одан әрі дамыту, жаңа құрылғылармен біріктіруді және жүйенің дәлдігі мен бейімделгіштігін арттыру үшін алгоритмдерді жетілдіруді қамтиды. Voxie.AI әртүрлі салаларда, соның ішінде үй және жұмыс процестерін автоматтандыруда, сондай-ақ заттар интернетінде ақылды және жоғары технологиялық басқару жүйелерін құрудың жаңа мүмкіндіктерін ұсынатын маңызды әлеуетке ие.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Timeweb. Голосовое управление в умном доме: возможности и технологии.  
Қолжетімді сілтеме: <https://habr.com/ru/companies/timeweb/articles/691610/>
2. Intelvision. Смарт-голосовое управление: технологии и перспективы.  
Қолжетімді сілтеме: <https://www.intelvision.ru/blog/smart-voice-control?ysclid=m6dkh28xf9155852659>
3. Maxxstore. Сравнение голосовых помощников: Siri, Google Assistant и Amazon Alexa — функции и интеграция с приложениями. Қолжетімді сілтеме: <https://maxxstore.ru/article/sravnenie-golosovyh-pomoschnikov:-siri,-google-assistant-i-amazon-alexa-%E2%80%94-funkcii-i-integraciya-s-prilozheniyami-670?ysclid=m6dky6nn1o192281253>
4. Научкор. Современные подходы к разработке голосовых интерфейсов.  
Қолжетімді сілтеме: <https://nauchkor.ru/uploads/documents/5b887daf7966e1073081b4e8.pdf?ysclid=m6dl6wnsff471854071>
5. Callcenterevent. Голосовые технологии для бизнеса: от теории к практике.  
Қолжетімді сілтеме: <https://callcenterevent.ru/eccs/files/2021/06/DF1440.pdf?ysclid=m6dlwsoo3d361548909>

## Бағдарлама коды:

```

#include <Arduino.h>
#include <WiFi.h>
#include <WebServer.h>
#include <ArduinoJson.h>
#include <ESP32Servo.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_NeoPixel.h>
#define BUZZER_PIN 23
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define SCREEN_ADDRESS 0x3C
#define LED_PIN1 27
#define LED_PIN2 26
#define LED_PIN3 25
#define LED_PIN4 33
#define NUM_LEDS 16
#define LED_BRIGHTNESS 50
#define PIR_PIN 32
#define SERVO_PIN 13
#define FRAME_DELAY (42)
#define FRAME_WIDTH (64)
#define FRAME_HEIGHT (64)
#define FRAME_COUNT (sizeof(wifi) / sizeof(wifi[0]))
#define MICROPHONE_PIN 34
#define TALK_LED_PIN 19
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
Adafruit_NeoPixel ring1(NUM_LEDS, LED_PIN1, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel ring2(NUM_LEDS, LED_PIN2, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel ring3(NUM_LEDS, LED_PIN3, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel ring4(NUM_LEDS, LED_PIN4, NEO_GRB + NEO_KHZ800);
WebServer server(serverPort);
Servo myservo;

bool discoMode = false;
bool securityEnabled = false;
bool motionDetected = false;
unsigned long discoStartTime = 0;
const unsigned long discoDuration = 5000;

uint32_t Wheel(byte WheelPos) {
    WheelPos = 255 - WheelPos;
    if(WheelPos < 85) {
        return ring1.Color(255 - WheelPos * 3, 0, WheelPos * 3);
    }
    if(WheelPos < 170) {
        WheelPos -= 85;
        return ring1.Color(0, WheelPos * 3, 255 - WheelPos * 3);
    }
    WheelPos -= 170;
    return ring1.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}

void runDiscoEffect() {
    static uint8_t phase = 0;
    static uint8_t position = 0;

    if (millis() - discoStartTime > discoDuration) {
        phase = (phase + 1) % 3;
        discoStartTime = millis();
    }

    switch(phase) {
        case 0:
            for (int r = 1; r <= 4; r++) {

```



```

        Adafruit_NeoPixel* ring = (r == 1) ? &ring1 : (r == 2) ? &ring2 : (r == 3) ? &ring3 : &ring4;
        uint32_t color = (millis() % 100 < 50) ? ring->Color(255, 255, 255) : 0;
        ring->fill(color);
        ring->show();
    }
    break;

case 1:
    position = (position + 1) % 256;
    for(int i = 0; i < NUM_LEDS; i++) {
        uint32_t color = Wheel(((i * 256 / NUM_LEDS) + position) & 255);
        ring1.setPixelColor(i, color);
        ring2.setPixelColor(i, color);
        ring3.setPixelColor(i, color);
        ring4.setPixelColor(i, color);
    }
    ring1.show();
    ring2.show();
    ring3.show();
    ring4.show();
    break;

case 2:
    position = (position + 1) % NUM_LEDS;
    for (int r = 1; r <= 4; r++) {
        Adafruit_NeoPixel* ring = (r == 1) ? &ring1 : (r == 2) ? &ring2 : (r == 3) ? &ring3 : &ring4;
        ring->clear();
        ring->setPixelColor(position, ring->Color(255, 0, 0));
        ring->setPixelColor((position + NUM_LEDS/2) % NUM_LEDS, ring->Color(0, 0, 255));
        ring->show();
    }
    break;
}
}

void drawSmile(bool eyesOpen = true) {
    display.clearDisplay();

    if (eyesOpen) {
        display.fillCircle(52, 22, 4, SSD1306_WHITE);
        display.fillCircle(76, 22, 4, SSD1306_WHITE);
    } else {

        display.drawLine(48, 22, 56, 22, SSD1306_WHITE);
        display.drawLine(72, 22, 80, 22, SSD1306_WHITE);
    }

    for (int x = -15; x <= 15; x++) {
        int y = sqrt(225 - (x * x)) / 2;
        display.drawPixel(64 - x, 45 + y, SSD1306_WHITE);
        display.drawPixel(64 - x, 44 + y, SSD1306_WHITE);
    }
    display.display();
}

void showMessage(const char* message) {
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);

    String text = String(message);
    int16_t x1, y1;
    uint16_t w, h;
    display.getTextBounds(text, 0, 0, &x1, &y1, &w, &h);
    int x = (SCREEN_WIDTH - w) / 2;
    int y = (SCREEN_HEIGHT - h) / 2;

    display.setCursor(x, y);
    display.println(text);
    display.display();
}

```

```

    delay(2000);
    drawSmile();
}
void showBootLogo() {

    for (int i = 0; i < 20; i++) {
        display.clearDisplay();
        display.drawCircle(64, 32, i * 2, SSD1306_WHITE);
        display.fillCircle(64, 32, i, SSD1306_WHITE);
        display.display();
        delay(30);
    }

    for (int i = 0; i < 360; i += 15) {
        display.clearDisplay();
        float x = 64 + cos(i * PI / 180) * i/10;
        float y = 32 + sin(i * PI / 180) * i/10;
        display.drawCircle(x, y, 5, SSD1306_WHITE);
        display.display();
        delay(20);
    }

    String text = "VOXIE.AI";

    for (int i = 0; i <= text.length(); i++) {
        display.clearDisplay();
        display.setTextSize(2);
        display.setTextColor(SSD1306_WHITE);

        int16_t x1, y1;
        uint16_t w, h;
        String partialText = text.substring(0, i);
        display.getTextBounds(partialText, 0, 0, &x1, &y1, &w, &h);
        int x = (SCREEN_WIDTH - w) / 2;
        int y = (SCREEN_HEIGHT - h) / 2;

        display.setCursor(x, y);
        display.println(partialText);
        display.display();
        delay(150);
    }

    display.clearDisplay();
    display.setTextSize(2);
    int16_t x1, y1;
    uint16_t w, h;
    display.getTextBounds(text, 0, 0, &x1, &y1, &w, &h);
    int x = (SCREEN_WIDTH - w) / 2;
    int y = (SCREEN_HEIGHT - h) / 2;
    display.setCursor(x, y);
    display.println(text);
    display.display();
    delay(1000);
}

void triggerSecurityAlert() {
    showMessage("MOTION DETECTED!");
    for (int j = 0; j < 5; j++) {
        for (int i = 0; i < NUM_LEDS; i++) {
            uint32_t color = ring1.Color(255, 0, 0);
            ring1.setBrightness(255);
            ring1.setPixelColor(i, color);
            ring2.setBrightness(255);
            ring2.setPixelColor(i, color);
            ring3.setBrightness(255);
            ring3.setPixelColor(i, color);
            ring4.setBrightness(255);
            ring4.setPixelColor(i, color);
        }
        ring1.show();
        ring2.show();
        ring3.show();
    }
}

```

```

    ring4.show();
    delay(100);
    tone(BUZZER_PIN, 1000, 100);
    tone(BUZZER_PIN, 2000, 100);
    for (int i = 0; i < NUM_LEDS; i++) {
        ring1.setPixelColor(i, 0);
        ring2.setPixelColor(i, 0);
        ring3.setPixelColor(i, 0);
        ring4.setPixelColor(i, 0);
    }
    ring1.show();
    ring2.show();
    ring3.show();
    ring4.show();
    delay(100);
    digitalWrite(BUZZER_PIN, LOW);
}
}

void showWiFiAnimation() {
    int frame = 0;
    while (WiFi.status() != WL_CONNECTED) {
        display.clearDisplay();
        display.drawBitmap(32, 0, wifi[frame], FRAME_WIDTH, FRAME_HEIGHT, 1);
        display.display();
        frame = (frame + 1) % FRAME_COUNT;
        delay(FRAME_DELAY);
    }

    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(10, 32);
    display.print("WiFi Connected!");
    display.display();
    delay(1000);
}

void handlePost() {
    if (server.hasArg("plain")) {
        String body = server.arg("plain");
        DynamicJsonDocument doc(1024);
        deserializeJson(doc, body);

        if (doc.containsKey("command")) {
            String command = doc["command"].as<String>();

            if (command == "disco") {
                discoMode = !discoMode;
                if (discoMode) {
                    discoStartTime = millis();
                    showMessage("DISCO MODE ON");
                    server.send(200, "application/json", "{\"status\":\"success\",\"state\":\"disco_started\"}");
                } else {
                    showMessage("DISCO MODE OFF");
                    ring1.clear(); ring2.clear(); ring3.clear(); ring4.clear();
                    ring1.show(); ring2.show(); ring3.show(); ring4.show();
                    server.send(200, "application/json", "{\"status\":\"success\",\"state\":\"disco_stopped\"}");
                }
            }

            else if (command == "security") {
                securityEnabled = !securityEnabled;
                if (securityEnabled) {
                    showMessage("SECURITY ENABLED");
                    server.send(200, "application/json", "{\"status\":\"success\",\"state\":\"security_on\"}");
                } else {
                    showMessage("SECURITY DISABLED");
                    server.send(200, "application/json", "{\"status\":\"success\",\"state\":\"security_off\"}");
                }
            }
        }
    }
}

```

```

else if (command == "light") {
  if (doc.containsKey("ring") && doc.containsKey("color")) {
    int ringNumber = doc["ring"].as<int>();
    uint32_t color = doc["color"].as<uint32_t>();
    int brightness = doc.containsKey("brightness") ? doc["brightness"].as<int>() : LED_BRIGHTNESS;
    brightness = constrain(brightness, 0, 255);

    Adafruit_NeoPixel* targetRing;
    switch(ringNumber) {
      case 1: targetRing = &ring1; break;
      case 2: targetRing = &ring2; break;
      case 3: targetRing = &ring3; break;
      case 4: targetRing = &ring4; break;
      default:
        server.send(400, "application/json", "{\"status\":\"error\",\"message\":\"Invalid ring number\"}");
        return;
    }

    targetRing->setBrightness(brightness);
    for(int i = 0; i < NUM_LEDS; i++) {
      targetRing->setPixelColor(i, color);
    }
    targetRing->show();
    showMessage("LIGHTS UPDATED");

    String response = "{\"status\":\"success\",\"ring\":\"" + String(ringNumber) + "\",\"brightness\":\"" +
String(brightness) + "\"}";
    server.send(200, "application/json", response);
  }
}

else if (command == "lock") {
  myservo.write(0);
  showMessage("DOOR LOCKED");
  server.send(200, "application/json", "{\"status\":\"success\",\"state\":\"locked\"}");
}

else if (command == "unlock") {
  myservo.write(90);
  showMessage("DOOR UNLOCKED");
  server.send(200, "application/json", "{\"status\":\"success\",\"state\":\"unlocked\"}");
}
}
}

void setup() {
  Serial.begin(115200);
  Wire.begin(21, 22);

  ring1.begin(); ring2.begin(); ring3.begin(); ring4.begin();
  ring1.setBrightness(LED_BRIGHTNESS);
  ring2.setBrightness(LED_BRIGHTNESS);
  ring3.setBrightness(LED_BRIGHTNESS);
  ring4.setBrightness(LED_BRIGHTNESS);
  ring1.show(); ring2.show(); ring3.show(); ring4.show();

  if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println("SSD1306 allocation failed");
    while (true);
  }

  WiFi.begin(ssid, password);
  showWiFiAnimation();
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }

  myservo.attach(SERVO_PIN);
  myservo.write(0);

```

```

pinMode(PIR_PIN, INPUT);
pinMode(MICROPHONE_PIN, INPUT);
pinMode(BUZZER_PIN, OUTPUT);
pinMode(TALK_LED_PIN, OUTPUT);

server.on("/control", HTTP_POST, handlePost);
server.begin();

showBootLogo();
}

void loop() {
  server.handleClient();
  int sample = analogRead(MICROPHONE_PIN);
  Serial.println(sample);
  digitalWrite(TALK_LED_PIN, abs(sample - 2048) > 200);

  if (discoMode) {
    runDiscoEffect();
    delay(20);
  }

  if (securityEnabled) {
    motionDetected = digitalRead(PIR_PIN);
    if (motionDetected) {
      triggerSecurityAlert();
    }
  }

  if (random(500) < 2) {
    drawSmile(false);
    delay(200);
    drawSmile(true);
    delay(2000);
  }
}

```

## ҚОСЫМША 2

