

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Коммерциялық емес акционерлік қоғам
Ғұмарбек Даукеев атындағы
«АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ»
Ғарыштық инженерия және телекоммуникация институты
Электроника және Робототехника кафедрасы

«ҚОРҒАУҒА ЖІБЕРІЛДІ»

Кафедра меңгерушісі
доцент Чигамбаев Т.О.

_____ «___» _____ 2020ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Arduino Uno микроконтроллері негізінде қашықтан басқару және бақылаудың сымсыз жүйесін әзірлеу»

Мамандығы: 5B071600 – Аспап жасау

Орындаған: Арасанбай А.Т. Тобы: ПСк-16-1

Ғылыми жетекшісі: к.т.н., доцент Байкенов Б.С.

Кеңесшілер:

экономикалық бөлім бойынша:

доцент Боканова Г.Ш.

_____ «___» _____ 2020ж.
(қолы)

өміртіршілігі қауіпсіздігі бойынша:

доцент Абикенова А.А.

_____ «___» _____ 2020ж.
(қолы)

есептеу техникасын қолдану бойынша:

к.т.н., доцент Байкенов Б.С.

_____ «___» _____ 2020ж.
(қолы)

Норма бақылаушы: _____
(ғылыми дәрежесі, лауазымы, Т.А.Ж.)

Пікір беруші: _____
(ғылыми дәрежесі, лауазымы, Т.А.Ж.)

Алматы 2020

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғам
Ғұмарбек Даукеев атындағы
«АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ»
Ғарыштық техника және телекоммуникация институты
«Электроника және робототехника» кафедрасы
Мамандығы: 5В071600 – Аспап жасау

Диплом жұмысқа орындауға берілген

ТАПСЫРМА

Студент Арасанбай Алина Төлегенқызы

Жұмыстың тақырыбы: «Arduino Uno микроконтроллері негізінде қашықтан басқару және бақылаудың сымсыз жүйесін әзірлеу»

20__ ж. «__» ____ №__ университет бұйрығымен бекітілді

Аяқталған жұмысты тапсыру мерзімі ____ «__» ____
2020ж.

Жұмысқа алғашқы деректер (талап етілетін зерттеу (жұмыс) нәтижелерінің параметрлері және зерттеу нысанының алғашқы деректері):

1. Андроид – қосымшаларды пайдалану арқылы Ардуино жұмысының және басқарудың құрылымын зерттеу.

2. Софтты жасау және бағдарламалау.

3. Құрастыру, реттеу және жұмысқа қабілеттілігін тексеру.

Диплом жұмысындағы әзірленуі тиіс мәселелер тізімі немесе диплом жұмысының қысқаша мазмұны:

Құрылған Android – қосымшасы арқылы Arduino платформасының көмегімен интерактивті екі жақты басқару жүйесі бар құрылғыны жасау. Әзірленуі тиіс мәселелер:

1) радиоэлементтерді басқару жүйесін әзірлеу;

2) дербес Android-қосымшасын әзірлеу;

3) дауыстық басқаруды біріктіру.

Графикалық материалдардың (міндетті түрде дайындалатын сызбаларды көрсету) тізімі:

1. Бағдарламаның құрылымдық схемасы.

2. Бағдарламаның блок схемасы.

3. Proteus бағдарласында проекти жасап, дәлелдеу.

4. Жобаны құрастырудың схемалық диаграммасы.

Негізгі ұсынылатын әдебиеттер:

1. Белов А. В. Микроконтроллеры AVR в радиолюбительской практике. — СПб.: Наука и Техника, 2007. - 339 с.
2. Белов А. В. Конструирование устройств на микроконтроллерах. - СПб.: Наука и Техника, 2005. - 256 с.
3. Гололобов В.Н. С чего начинаются роботы? - М.: Солон-Пресс, 2011. - 189 с.
4. Голубцов М. С. Микроконтроллеры AVR от простого к сложному. - М.: Солон-Пресс, 2003. - 288 с.
5. Иго Т. Arduino, датчики и сети для связи устройств: Пер. с англ. – 2 изд. – СПб.: БХВ-Петербург, 2015. - 544 с.

Жұмыс бойынша бөлімдеріне қатысы белгіленген кеңесшілер

Бөлімдері	Кеңесшілері	Мерзімі	Қолы
Экономикалық бөлім	Доц. Боканова Г.Ш.	16.04-3.05.2020 ж.	
Өмір тіршілігінің қауіпсіздігі бөлімі	Доц. Абикенова А.А.	12.04-21.04.2020 ж.	

Дипломдық жұмысының дайындау

КЕСТЕСІ

Бөлімдердің атауы, әзірленетін мәселелердің тізімі	Ғылыми жетекшіге ұсыну	Ескерту
1. Жүйенің жұмыс жасау принциптері	2.03.2020 ж.	
2. Ардуино микроконтроллеріне шолу	7.03.2020 ж.	
3. Жабдықтарды таңдау	23.03.2020 ж.	
4. Жобаға қажет құрылғылар мен радиобөлшектер	4.04.2020 ж.	
5. Бағдарламамен қамтамасыз ету	29.04.2020 ж.	

Тапсырманың берілген уақыты «__» _____ 2020ж.

Кафедра меңгерушісі: _____ (Чигамбаев Т.О.)
(қолы)

Жұмыстың ғылыми жетекшісі: _____ (Байкенов Б.С.)
(қолы)

Орындалатын тапсырманы
қабылдаған студент: _____ (Арасанбай А.Т.)
(қолы)

Аңдатпа

Бұл дипломдық жобада Android-қосымшасы арқылы Arduino аппараттық–есептеу платформасы арқылы радиоэлементтерді басқарудың екі жақты интерактивті жүйесі ұсынылған. Бұл жұмыста Android құрылғысының көмегімен одан әрі басқару үшін Bluetooth модулінің көмегімен енгізу-шығару құрылғысының жұмыс істеу принципі, оны қосу және деректер беру көрнекі зерттелген.

Аннотация

В дипломной работе представлена цифровая система голосового управления на платформе Ардуино и смартфона с Android–приложением. Приведены программы управления в среде Ардуино для внешних устройств, включая приемопередатчик Bluetooth, подключаемой к плате Ардуино и пульта управления в виде смартфона с ОС Android.

Annotation

This diploma project presents an interactive two-way radio element control system using the Arduino hardware-computing platform via an Android application. In this paper, we have clearly studied the principle of operation of the I / o device, its connection and data transfer using a Bluetooth module for further control using an Android device.

Мазмұны

Кіріспе	7
1 Жүйенің жұмыс жасау принциптері	9
1.1 Ардуино микроконтроллеріне шолу	9
1.2 Bluetooth басқаруымен интерактивті жүйені құру	13
2 Жабдықтарды таңдау	15
2.1 Жобаға қажет құрылғылар мен радиобөлшектер	15
2.2 Arduino Mega 2560	15
2.3 Bluetooth HC-06 модулі	16
2.4 DS18B20 температура датчигі	17
2.5 Жарық диодтары 5 мм	18
3 Бағдарламамен қамтамасыз ету	19
3.1 Бағдарламаның құрылымдық сұлбасы	19
3.2 Бағдарламаның блок-схемасы	20
3.3 Ардуиномен жұмыс істеуге арналған скетч	21
4 Өмір тіршілігінің қауіпсіздігі	22
4.1 Теориялық бөлім	22
4.2 Еңбек жағдайларын талдау	22
4.3 Жабдықты және оның оңтайлы орналасуын талдау	23
4.4 Табиғи және жасанды жарықтандыруды есептеу	24
4.5 Микроклимат және көру жүктемесі	26
4.6 Кондиционерлеу жүйесін есептеу	28
4.7 Бөлімнің қорытындысы	31
5 Техникалық-экономикалық негіздеме	32
5.1 Жобаның мақсаттары мен міндеттері	32
5.2 Шығындар жоспарын анықтау	33
5.3 Аспапты әзірлеуге арналған шығындарды есептеу	33
5.3.1 Еңбекақы төлеу қоры	33
5.3.2 Әлеуметтік салық	37
5.3.3 Амортизация	37
5.3.4 Материалдар мен жинақтаушылардың шығыны	38
5.3.5 Электр энергиясына шығындар	39
5.3.6 Үйді жалға алу шығындары	40
5.4 Динамикалық көрсеткіштерді есептеу	42
5.5 Бөлім бойынша қорытынды техникалық-экономикалық негіздеме	46
Қорытынды	47
Әдебиеттер тізімі	48
А қосымшасы	49

Кіріспе

Микроэлектрониканың дамуы және оны өнеркәсіптік өндірісте, әртүрлі объектілер мен процестерді басқару құрылғылары мен жүйелерінде кеңінен қолдану қазіргі уақытта ғылыми-техникалық прогрестің негізгі бағыттарының бірі болып табылады.

Бұйымдарда микроконтроллерлерді пайдалану техникалық-экономикалық көрсеткіштердің (құны, сенімділігі, тұтынылатын қуаты, габариттік өлшемдері) артуына әкеліп қана қоймай, сонымен қатар бұйымдарды әзірлеу уақытын қысқартуға мүмкіндік береді және оларды түрлендірілген, бейімделген етеді. Басқару жүйелерінде микроконтроллерлерді пайдалану төмен құн кезінде тиімділіктің жоғары көрсеткіштеріне қол жеткізуді қамтамасыз етеді.

Arduino - екі негізгі компоненттен тұратын аппараттық есептеу платформасы: енгізу-шығару платасы және Processing/Wiring тілінде өңдеу ортасы. Arduino жаңадан келгендерге де, кәсіби мамандарға де электрондық құрылғыларды жасау үшін ыңғайлы. Бұл платформа қарапайым бағдарламалау тілі, ашық архитектурасы және бағдарламалық коды үшін бүкіл әлемде кең танымал. Бұл платформаның ерекшелігі, ол USB арқылы программаторларды пайдаланбай ол бағдарлана береді.

Arduino платасында RS-232 (Recommended Standart 232, асинхронды интерфейс үшін физикалық деңгей) сигналдар деңгейлерін TTL деңгейіне (транзисторлы-транзисторлық логика - биполярлы транзисторлар (үш электрлік жартылай өткізгіш аспап және резисторлар) негізінде құрылған цифрлық логикалық микросхемалардың түрлері) және керісінше конвертация үшін инвертелетін схемадан тұрады.

Arduino дамуының интеграцияланған ортасы - бұл код редакторы, компилятор және платадағы бағдарламаны қосып беру модулі кіретін Java-дағы кроссплатфорлы бағдарлама. Arduino үшін қолданылатын бағдарламалау тілі кейбір кітапханалармен толықтырылған СИ++ өте ұқсас. Бағдарламаны өңдеу алдыңғы процессордың көмегімен жүзеге асырылады, ал AVR-GCC көмегімен компиляцияланады.

Сіздің жобаларыңызда жиі қашықтан басқару немесе телефон гаджеттерінен деректерді беру қажеттілігі туындайды. Сондықтан осы жобада біз ең танымал және қолжетімді Bluetooth байланысын пайдаланамыз. Қосылу арнайы модульдер арқылы орнатылады.

Көптеген модульдер BC417 чиппен және Flash жады қолданылады. Чип Bluetooth 2.0 + EDR, AT командалары спецификациясын қолдайды, Master немесе Slave (HC-03/HC-5 модульдері, AT-командасы орната алады) режимінде жұмыс істей алады, алмасу жылдамдығын 2400-ден 1382400-ге дейін қолдайды.

Модульдің қоректену кернеуі 3.3 В, тұтыну тогы ~50мА. Құрылғыны басқару бағдарламасы Android studio бағдарламасы арқылы жасалған. Android Studio-Android-дегі әзірлеудің ресми ортасы. Шын мәнінде, Android Studio — плагиндермен танымал Java Ide IntelliJ IDEA. JetBrains

компаниясынан IntelliJ IDEA бағдарламалық қамтамасыз етуге негізделген Android Studio-Android қосымшаларын әзірлеудің ресми құралы

Android SDK Bluetooth құрылғысының адаптерімен жұмыс істеуге, жақын орналасқан құрылғыларды іздеуге, смартфондар арасында деректерді таратуға және т.б. қабілетті қуатты Bluetooth API-мен жұмыс істеу мүмкіндігімен жабдықталған. Бағдарлама Bluetooth-пен жұмыс істеудің әр түрлі мүмкіндіктеріне ие, олар осы функциямен жұмыс істейтін қосымшаларды жасау кезінде пайдалы болады, бұл бізге жобамен жұмыс істеу үшін қажет болды.

1 Жүйенің жұмыс істеу принципі

1.1 Ардуино микроконтроллеріне шолу

Қазіргі уақытта сөйлеуді тануға байланысты технологияларға деген қызығушылықтың айтарлықтай өсуі байқалады. Мысалы, дауыс командалары арқылы құрылғыларды басқару міндеттері. Соңғы уақытта "ақылды үй" технологиясы ретінде кеңінен жарнамаланатын Дауыстық командалармен Андроид - құрылғылардың көмегімен үй, кеңсе техникасын басқару мүмкіндігі пайда болды.

Бұл жұмыста Bluetooth HC - 06 арқылы Android құрылғысы арқылы Ардуино дауыстық басқару бағдарламасы ұсынылады. Ардуиноға үш атқарушы механизм және температура датчигі қосылған. Тиісті командаларды орындағаннан кейін бағдарлама жұмыс нәтижесі туралы дауыстық хабарламаларды синтездейді. Бұл мысалы - "ақылды үй" жалпы тапсырмасының типтік қосымшасы. Бағдарлама қарапайым сұрақтарға жауап бере отырып, құрылғы мен пайдаланушы арасындағы қарапайым диалогты іске асырады. Сөйлеуді тану және синтездеу үшін Google құралдары қолданылады. Егер смартфон оффлайн режимінде дауыс іздеуді қолдаса, Интернетке кіру міндетті емес. Андроид үшін бағдарлама Android Studio (Java тілі) ортасында, ал Ардуино үшін - C++ (Wiring жобасы) Ардуиноның даму ортасында жазылған. LG G3 Stylus телефонында және Acer A500 планшетінде сынақтан өтті.

Arduino қарапайымдылықтың және достықтың арқасында танымал болды. Бұған мыңдаған басылымдар, оқулықтар, ғаламтордағы жазбалар және жұмыс бойынша түрлі баспа және бейне құралдар ықпал етеді.

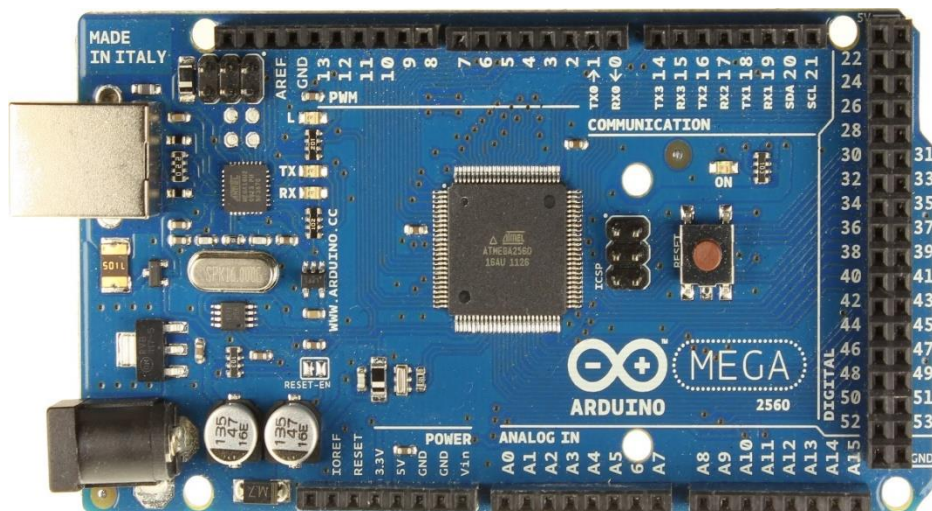
Arduino бағдарламасы контактілер енгізу/шығару басқару үшін қарапайым және түсінікті функциялармен толықтырылған қарапайым C++ түрінде жазылады. Arduino-мен жұмыс істеуге ыңғайлы болу үшін Windows, Mac OS және Linux астында жұмыс істейтін "Arduino IDE" бағдарламалаудың тегін ресми ортасы бар. Ол жаңа бағдарламаны Arduino-ны жүктеу қиын болмайды, ол үшін тек USB арқылы компьютерге баспа платасын қосу қажет.

Қажетті құрылғыларды жасау үшін арнайы жабдықтың қажеті жоқ. Толық құрылғыны арнайы макет тақтасын, маңдайшалар мен сымдарды пайдаланып жинауға болады [1].

Arduino тағы бір ерекшелігі-shields немесе жай "Шилд" деп аталатын кеңейту платаларының болуы. Бұл қосымша платалар, Arduino үстіне бутерброд сияқты қабаттарға ұқсас қойылады, ол жаңа мүмкіндіктерді береу үшін қажет. Мысалы, жергілікті желі мен интернетке қосылу үшін, GPS жерсеріктерінен координаттар мен уақыт алу үшін қуатты моторларды басқару үшін кеңейту платалары бар.

ATmega2560 микроконтроллерінде Arduino Mega (1.1-сурет) құрылған (техникалық сипаттамасы). Платада 54 сандық кіріс/шығыстары бар (оның

14-ті ШИМ шықпасы ретінде пайдаланылуы мүмкін), 16 аналогтық кіріс, 4 тізбекті UART порты, 16 МГц кварц генераторы, USB коннекторы, қуат қосқышы, ICSP қосқышы және қайта жүктеу түймесі бар. Жұмыс істеу үшін платформаны компьютерге USB кабелі арқылы қосу немесе AC/DC адаптері немесе аккумуляторлық батарея арқылы қуат беру қажет.



1.1 сурет - Arduino Mega 2560 микроконтроллері

Arduino платформасының шықпалары кіріс немесе шығыс ретінде жұмыс істей алады. Бұл бөлімде осы режимдердегі шықпалардың жұмыс істеуі сипатталады. Сонымен қатар, Arduino (Atmega) аналогтық кірістердің көпшілігі енгізу/шығару цифрлық порттары сияқты конфигурациялауға және жұмыс істей алатынына назар аудару қажет (1.2-сурет). Енгізу порты ретінде конфигурацияланған енгізу/шығару портының (pin) қасиеттері. Arduino (Atmega) тұжырымдары стандартты түрде енгізу порттары ретінде құрастырылған, осылайша pinMode () функциясына айқын декларацияны талап етілмейді.

Егер енгізу портына сигнал келіп түспесе, онда бұл жағдайда портқа белгілі күйді орнату ұсынылады. Бұл +5 В кіріске (резисторлардың қоректендіруге тартатын) немесе жерге (жерге резистор тартатын) қосатын 10 кОм тартқыш резисторларды қосу арқылы жасалады. Atmega микроконтроллерінде қоректендіруге тартатын бағдарланған құрылған 20 кОм резисторы бар. Резисторлардың деректерін бағдарламалау келесі түрде жүзеге асырылады.

```
pinmode (pin, INPUT); // шығысқа енгізу портын белгілеу (pin, HIGH); //  
тартқыш резисторды қосу [2].
```



1.2 сурет - Ардуино MEGA платасының шығыс диаграммасы.

Тартқыш резистор кіріс порты ретінде жұмыс істейтін шығысқа қосылған жарық диодты сәл жарық түсіру үшін жеткілікті ток өткізе алады. Сондай-ақ, сәл ғана сәулесі бар жарық диодты бағдарламалау кезінде қорытынды pinMode() функциясына шығыс порты ретінде орнатылмаған. Тартқыш резисторлар (микроконтроллер жадының ішкі мекен-жайлары) бұл шығыс күйлерін басқаратын: HIGH немесе LOW, регистрлермен басқарылады. Сондықтан, микросхеманың шығысындағы шығыс портының pinMode() функция конфигурациясы HIGH – ке мәнді береді. Бұл процедура кері бағытта да жұмыс істейді, яғни егер шықпа HIGH мәні болса, онда pinMode() функциясы енгізу порты ретінде микросхеманың шығыс конфигурациясы қоректендіруге тартқыш резисторды қосады [3].

Ескерту: жарық диоды мен резисторлардың микросхеманың 13 шықпасына енгізу порты ретінде қосылғандықтан пайдалану қиын.

20 кОм резисторға қорек көзінің тартыруының қосылуы кезінде кірісте 1,7 В болады 5В орынына, өйткені тізбекті қосылған резистор мен жарық диодында кернеу түсуі болады. Микросхеманың 13 шықпасын сандық енгізу порты ретінде шықпалар мен тартылатын сыртқы резистордың жері

арасында қосу қажет. Енгізу/шығару портының қызметі, шығыс порты ретінде конфигурацияланған. Шығыстар, шығыс порты ретінде конфигурацияланған, олар төменгі импенданс күйінде болады. Шығыстардың мәлеметтері өзі арқылы жеткілікті үлкен ток өткізе алады. Atmega микросхемасының шықпалары басқа құрылғылар үшін 40 мА дейін тогы бар қабылдағыш (теріс) немесе көзі (оң) ретінде бола алады. Осындай токтың мәні жарық диодты, датчиктер (жүйелі түрде қосылған резистор) міндетті қосу үшін жеткілікті, бірақ көптеген реле, соленоидтар және қозғалтқыштар үшін жеткіліксіз.

Arduino шықпаларының қысқа тұйықталуы немесе энергияны қажет ететін құрылғыларды қосу әрекеттері шығыс транзисторларын немесе Atmega микроконтроллерін зақымдауы мүмкін. Көп жағдайда бұл әрекеттер микроконтроллерде шығысты ажыратуға әкеледі, бірақ схеманың қалған бөлігі бағдарламаға сәйкес жұмыс істейтін болады. Егер құрылғы жұмыс істеу үшін үлкен ток қажет болмаса, құрылғыны 470 Ом немесе 1 кОм резисторлары арқылы платформа шығыстарына қосу ұсынылады.

Аналогты-сандық түрлендіргіш. Arduino - да қолданылатын Atmega микроконтроллерлерінде алты арналы аналогты-сандық түрлендіргіш (АСТ) бар. Түрлендіргіш ажыратымдылығы 10 битті құрайды, бұл 0-ден 1023-ге дейінгі мәндерді шығыста алуға мүмкіндік береді. Arduino платформаларының көпшілігінің аналогтық кірістерінің негізгі қолдану аналогтық датчиктерді оқу болып табылады, бірақ сонымен қатар олар кең қолданылатын (GPIO) (0 - 13 енгізу/шығару сандық порттары сияқты) енгізу/шығару функционалдығы бар.

Mega жаңа кодты жазар алдында қайта жүктеу платформадағы батырманы баспай ақ, бағдарламаның өзі жүзеге асырылатындай етіп жасалған. Деректер (DTR) ағынымен басқаратын ATmega8U2 желілерінің бірі 100 нФ конденсатор арқылы ATmega2560 микроконтроллердің жүктеме шығысына қосылған. Бұл желіні іске қосу, яғни төмен деңгейдегі сигнал беру, микроконтроллерді қайта жүктейді. Arduino бағдарламасы осы функцияны пайдалана отырып, бағдарламалау ортасында Upload батырмасын бір рет басу арқылы кодты жүктейді. DTR желісі бойынша төмен деңгейдегі сигнал беру код жазбасын бастай отырып, жүктеушінің таймаутын қысқартады.

Функцияның тағы бір қолданылуы бар. Arduino бағдарламасына Mac X немесе Linux ОЖ бар компьютеріне (USB арқылы) қосылған сайын Mega2560-ны қайта жүктеу жүзеге асырылады. Қайта жүктелгеннен кейін келесі жарты секундтарда жүктеуші жұмыс істейді. Бағдарламалау кезінде платформа қате деректерді (жаңа бағдарлама кодынан басқа, барлығы) алуды болдырмау үшін кодтың бірнеше алғашқы байттары кешіктіріледі. Егер платформаға жазылған скетчты бір реттік жөндеу немесе бірінші рет іске қосу кезінде басқа да деректерді енгізу жүргізілсе, бағдарлама деректерді беру алдында бір секунд ішінде компьютерде күтілетініне көз жеткізу қажет.

Mega 2560 - да тиісті желінің үзілуімен автоматты қайта жүктеу желісін өшіру мүмкіндігі бар. Желінің екі шетінен микросхемалардың контактілері содан қалпына келтіру мақсатында кейін қосылуы мүмкіндігі бар. Желі "RESET-EN" маркіленген. Автоматты қайта жүктеуді ажырату, сондай-ақ осы желі мен 5 В қорек көзі арасында 110 Ом резисторын қосу арқылы мүмкіндік бар. USB жалғағышының ток қорғанысы.

Arduino Mega2560-да компьютердің USB портын қысқа тұйықталу токтарынан және жоғары токтардан қорғайтын қайта жүктелетін балқымалы ендіrmесі орнатылған. Дегенмен, барлық компьютерлердің осындай қорғауы болса да, бұл сақтандырғыш қосымша кедергіні қамтамасыз етеді. Сақтандырғыш USB порты арқылы 500 мА астам ток өткен кезде деректер алмасуын автоматты түрде үзеді.

Осылайша, қосымша енгізу/шығару порттарын қолдану қажет болған жағдайда пайдаланылмайтын аналогтық кірістерді конфигурациялау мүмкіндігі бар [4].

1.2 Bluetooth басқарылатын интерактивті жүйені құру

Бүгінгі күні сөйлеуді тануға байланысты технологияларға деген қызығушылықтың айтарлықтай өсуі байқалады. Мысалы, дауыс командалары арқылы құрылғыларды басқару міндеттері. Соңғы уақытта "ақылды үй" технологиясы ретінде кеңінен жарнамаланатын дауыстық командалармен Андроид - құрылғылардың көмегімен үй, кеңсе техникасын басқару мүмкіндігі пайда болды.

Мұнда Андроид - құрылғының көмегімен Ардуино дауыстық басқару бағдарламасы ұсынылған Bluetooth HC-06 арқылы Ардуиноға үш орындаушы механизм және температура датчигі қосылған. Тиісті командаларды орындағаннан кейін бағдарлама жұмыс нәтижесі туралы дауыстық хабарламаларды синтездейді. Бұл мысал ретінде - "Ақылды үй" жалпы тапсырмасының типтік қосымшасы. Бағдарлама қарапайым сұрақтарға жауап бере отырып, құрылғы мен пайдаланушы арасындағы қарапайым диалогты іске асырады. Сөйлеуді тану және синтездеу Google құралдары қолданылады. Егер смартфон оффлайн режимінде дауыс іздеуді қолдаса, онда Интернетке кіру міндетті емес. Андроид үшін бағдарлама Android Studio (Java тілі) ортасында, ал Ардуино үшін - C++ (Wiring жобасы) Ардуиноның ортасында жазылған.

Android бағдарламасы екі үлкен блоктан тұрады-бұл (I) қосымшаның жұмысын бастау блогы және "Диалогты бастау үшін басу" (II) батырмасын басқаннан кейін қосымшаның жұмыс блогы.

Бағдарлама жұмысының келесі ерекшеліктеріне тоқтау қажет:

1) Сөйлеу синтезаторын іске қосар алдында оны инициализациялау қажет. Бұл үшін MainActivity негізгі класы TextToSpeech.OnInitListener интерфейсімен жасалады. Инициализация кезінде onInit(int status) әдісі қолданылады.

2) `utteranceprogresslistener` класындағы `onDone` әдісін қолдану үшін, сөйлеу синтезінің дыбыс динамигімен генерацияның аяқталуын қадағалайды, кілт `params`-та сақтауға арналған хеш картасын сипаттау қажет. Синтезатордың инициализациясы кезінде кілт беруді `params.put (TextToSpeech.Engine.KEY_PARAM_utterance_id, "hello")` әдісі арқылы орындау қажет. сондай-ақ `tts.setOnUtteranceProgressListener (new utteranceProgressListener())` сөйлеу синтезінің тыңдаушысын орнату қажет.

3) Ардуинодан температураны оқу `Timer` және `TimerTask` класстарының көмегімен бір секундта бір рет орындалады, олар болашақта белгілі бір уақыт аралығында тапсырмаларды іске қосуды орындайды (бағдарлама бойынша сәйкесінше 500 және 1000 миллисекунд).

`Thread` негізгі ағынында кейіннен оқып алуы бар температураны сұрастыруды ұйымдастыру әрекеті, ағымдағы температура мәнін алу үшін оны қайта сұратуға сұраныс қажет. Яғни, температураның дұрыс ағымдағы мәнін алу үшін екі рет айту керек: "температура" деп. `Handler` класты `Thread` жеке ағынын пайдаланған кезде дұрыс жұмыс істейді, бірақ кириллицаның қате жіберілуіне әкеледі. Латынша дұрыс беріледі, сондықтан оны бір уақытта дауыс синтезаторына кез келген сәтте қол жетімді `readMessage` айнымалы секунд сайын Android сұраумен бірге температураны оқу әдісін қолданамыз. "Температура" командасы кезінде оның мәні `readMessage` строкасынан алынады және синтезатормен синтезделеді [5].

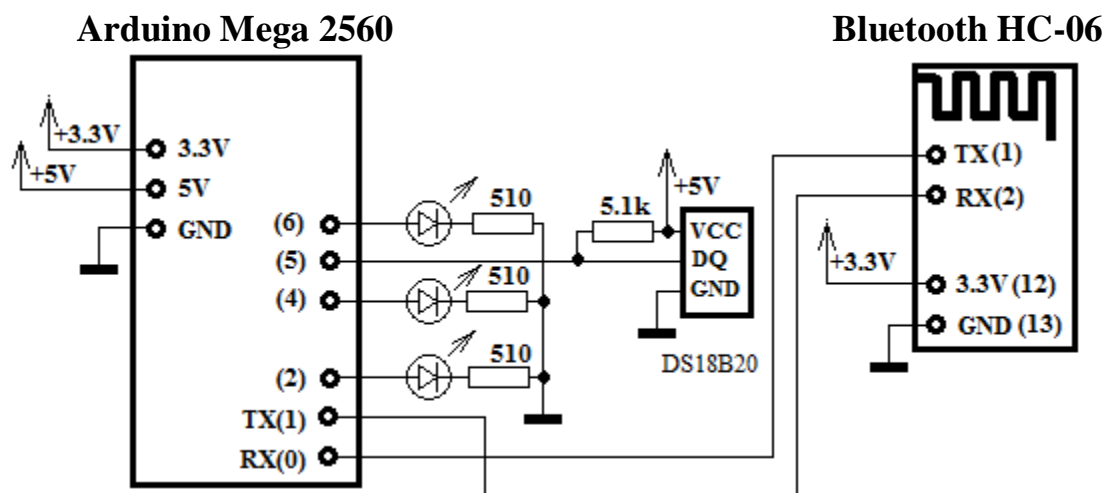
2 Жабдықты таңдау

2.1 Жобаға қажет құрылғылар мен радиобөлшектер

Жобаны құру үшін келесі құрылғылар мен радиобөлшектер қажет болды:

- Arduino Mega 2560 микроконтроллері (Sunfounder өндірісі)
- Bluetooth –HC-06 модулі
- DS18B20 датчик температурасы
- Жарық диодтар – 5 мм
- 510 Ом резисторлары (жарық диодтар үшін тоқты шектеу үшін)
- 5,1 кОм резисторы (датчик температурасы үшін тоқ шектегіші)
- Микроконтроллер үшін қуат көзі (7 – 12 В беруші үшін қалыпты жұмыс)
- ПАПА-МАМА типті сымдары

Алдымен, жобаның барлық бөлшектерінің дұрыс жұмыс жасауына көз жеткіземіз, бүтіндігіне көз жеткіземіз, қуат көзіндегі кернеуді өлшейміз. Кейін, схемаға сәйкес барлық бөлшектерді микроконтроллерге қосамыз. (2.1 сурет).



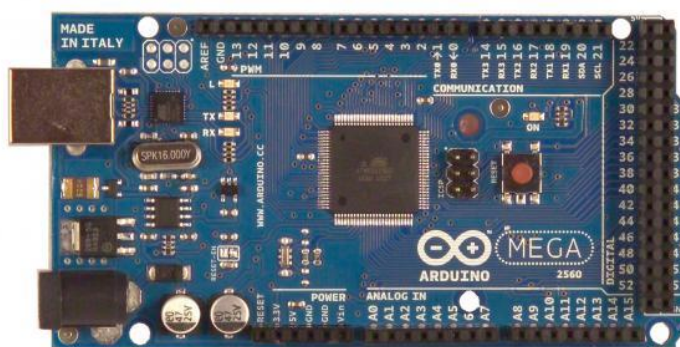
2.1 сурет - Болашақ жобаны құрастырудың принципіалды схемасы

2.2 Arduino Mega 2560

Arduino Mega ATmega2560 микроконтроллерінде (техникалық сипаттамасы) құрылған. Платада 54 сандық кіріс/шығысы бар (оның 14 ШИМ шығысы ретінде пайдаланылуы мүмкін), 16 аналогтық кірістері, 4 тізбекті UART порты, 16 МГц кварц генераторы, USB коннекторы, қуат қосқышы, ICSP қосқышы және қайта жүктеу батырмасы бар [6]. Жұмыс істеу үшін платформаны компьютерге USB кабелі арқылы қосу немесе AC/DC адаптері немесе аккумуляторлық батарея арқылы қуат беру қажет. Arduino Mega 2560 Uno немесе Duemilanove платформалары үшін әзірленген барлық кеңейту платаларымен үйлесімді.

Қысқаша сипаттамалары:

- Микроконтроллер ATmega328;
- Жұмыс кернеуі 5 В;
- Кіріс кернеуі (ұсынылатын) 7-12 В;
- Кіріс кернеуі (шектеу) 6-20 В;
- Сандық кіріс / Шығу 14 (олардың 6 ШИМ шығу ретінде пайдаланылуы мүмкін);
- Аналогтық кіріс 6;
- Кіріс/шығыс арқылы тұрақты ток 40 мА;
- 3.3 В шығару үшін тұрақты ток 50 мА;
- Флеш-жады 32 Кб (ATmega328) оның 0.5 Кб жүктеуші үшін қолданылады;
- ОЗУ 2 Кб (ATmega328);
- EEPROM 1 Кб (ATmega328);
- Тактикалық жиілігі 16 МГц.



2.2 сурет - ATmega2560

2.3 Bluetooth HC-06 модулі

Bluetooth HC-06 модулі Arduino-ны басқа құрылғыларына қосылу үшін bluetooth бойынша орындалады. Модуль пассивті режимде жұмыс істейді, яғни басқару құрылғысында (Master) іздеуді орнату керек (ноутбук, телефон), құрылғыны табу (әдетте оның атауы linvor), содан кейін Мастер- құрылғыда тізбекті порт пайда болады, оның бәрі Arduino-да пайда болады, және керісінше, Arduino не жіберседе бәрі сіздің компьютеріңізде қабылданады. AT командалар көмегімен модульдің параметрлерін құруға болады [7].

Пиндеу:

- STATE-бұл жерге орнатылған жарық диодтың сигналы қайталанады, егер модуль қосылғанда жарық диоды жыпылықтайды, байланыс орнатылғанда-жанады;
- RXD-осы пинде модуль деректерді қабылдайды (яғни, сіздің скетчте осында деректерді жіберу керек);
- TXD-мұнда модуль деректерді жібереді;
- GND-жер;

- VCC-5В қуат көзі;
- EN - қос/өшір, егер мұнда логикалық бірлікті (немесе жай логикалық бірлікті) берсе, онда модуль өшіп қалады, егер логикалық нөл болса (немесе осы пинді қосыпаса) жұмыс істейді.

Сипаттамалары:

- Қорек кернеуі 3,3-6 В;
- 5 В логикалық бірліктің максималды кіріс кернеуі;
- 3,3 В логикалық бірліктің шығу кернеуі;
- Ең жоғары тұтыну тогы 45 А;
- Деректерді беру жылдамдығы 1200-1382400 бод;
- Тікелей көрініс кезіндегі байланыс қашықтығы 30 м.



2.3 сурет – Bluetooth-модуль HC-06

2.4 DS18B20 температура датчигі

DS18B20 микросхемасы негізіндегі герметикалық датчикті екі әдіспен қосуға болады:

Үш сым бойынша: қуат (қызыл), жер (қара) және сигнал (ақ).

Екі сым бойынша: жер және сигнал. Бұл жағдайда датчик кейде теріс көрсеткіштер бере алады, ол сүзгілеудің соңғы нәтижесінен алып тастауға болады.

Қосылу әдісіне қарамастан, сигналдық сымын 4,7 кОм резистор арқылы қоректендірумен қосу қажет. Тек бір датчикті қосқан кезде, 10 кОм резистор да жарамды [8].

Сенсорды Arduino немесе макеттік платаға қосу үшін қысқыш клеммниктерін пайдалану ыңғайлы болады.

1-Wire құрылғыларын Arduino-ға қосу үшін дайын кітапхана бар.

Сипаттамалары:

- Өлшенетін температура диапазоны: -55...+125 °C;
- Дәлдік: $\pm 0,5^{\circ}\text{C}$ (-10...+85 °C);
- Деректерді алу уақыты: 12 биттік рұқсат кезінде 750 мс; 9 биттік рұқсат кезінде 94 мс;
- Қуат кернеуі: 3-5, 5 В;
- Әрекетсіздік кезінде тұтынылатын ток: 750А;
- Сұрастыру кезінде қоректенетін ток: 1 мА.



2.4 сурет – DS18B20 температура датчигі

2.5 Жарық диодтары 5 мм

5 мм дөңгелек жарық диодтары - жұмыстық кернеуі 1,9 В-дан 3,4 В-ға дейінгі және 20мА ток күші бар жартылай өткізгішті жарық көздері. Жарық күші номиналға байланысты 30 мкд-дан 30 000 мкд-ға дейін өзгереді.

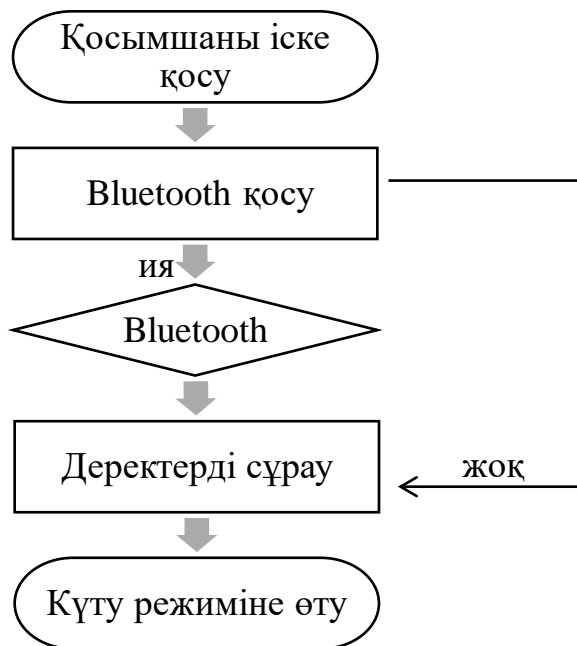
2.1 кесте – 5мм дөңгелек жарық диодтар

Сипаттамалар:	Қызыл	Жасыл	Көк
Макс. Тура кернеуі (В)	2,3	2,5	2,5
Ток күші (мА)	20	20	20

3 Бағдарламалық қамтамасыз ету

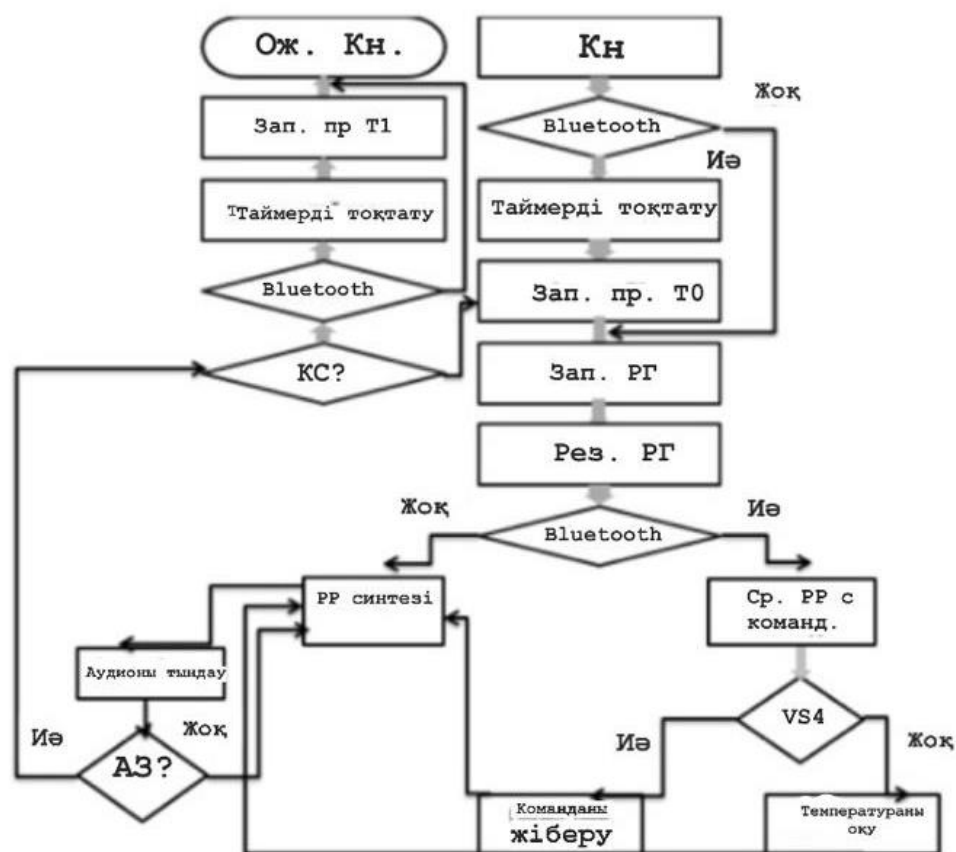
Бағдарламалық қамтамасыз ету жобаның ең маңызды бөлігі болып табылады, өйткені дұрыс бағдарламалық жасақтамаңыз, тіпті ең қарапайым құрылғы жұмыс істемейді. Android жасау – Java қосымшасымен сәйкес (Android Studio 1.0.1)

3.1 Бағдарламаның құрылымдық схемасы



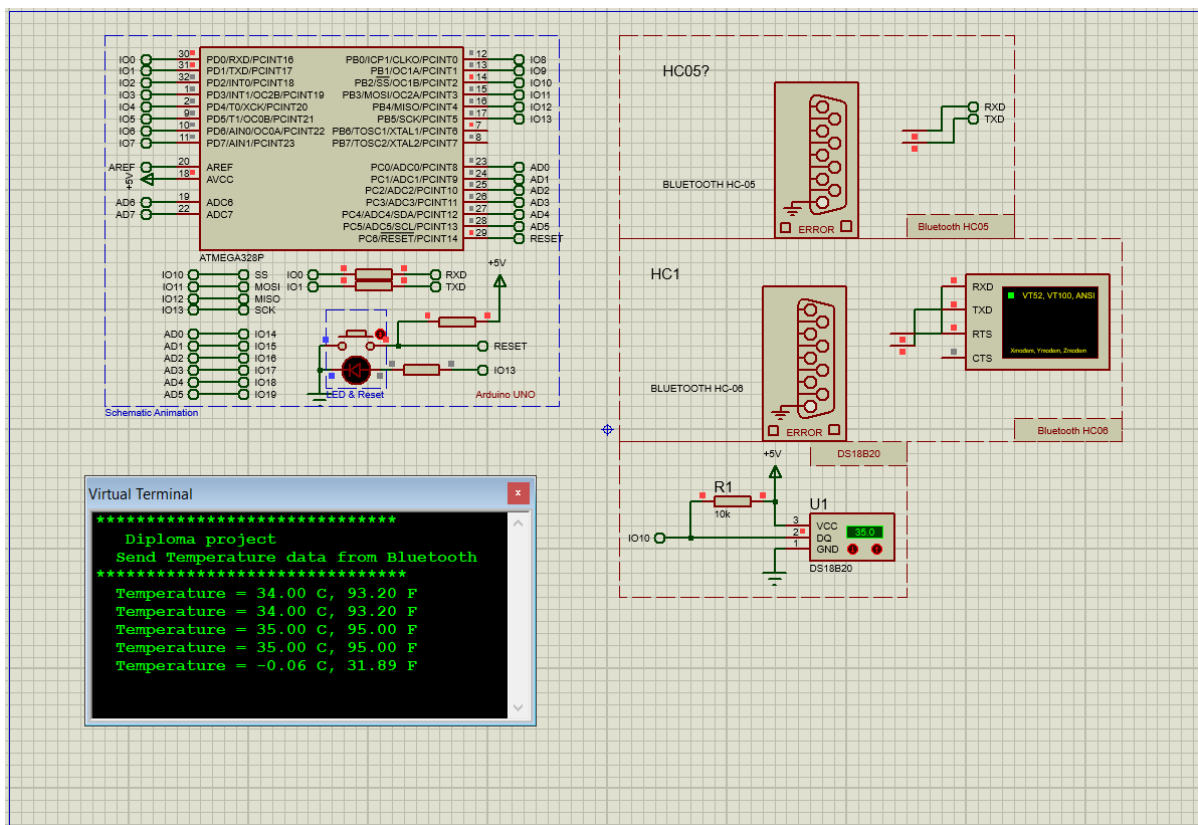
3.1 сурет – Бағдарламаның құрылымдық схемасы

3.2 Бағдарламаның блок-схемасы



3.2 сурет – Бағдарламаның блок-схемасы

Дауыстық басқару процесін Proteus-те іске асыру процесін 3.3 суретте көруге болады.



3.3 сурет – Proteus-тегі дауыстық басқару схеманысының үлгісі

4 Өмір тіршілігінің қауіпсіздігі

4.1 Теориялық бөлім

Еңбекті қорғау-заңнамалық актілерді, экономикалық, әлеуметтік, техникалық, ұйымдастырушылық және емдеу-алдын алу іс-шаралары мен адамның барынша қауіпсіздігін, денсаулығын және еңбекке қабілеттілігін уақытында қолдауды қамтамасыз ететін құралдарды жинау.

Еңбекті қорғаудың негізгі міндеті-қызметкердің жарақатынан немесе ауруынан болатын қауіп-қатерді барынша азайту, бұл ретте жайлылық пен максималды өнімділікті қамтамасыз ету болып табылады. Қауіпті фактор – бұл өндірістік фактор-әсер етуі қызметкердің белгілі бір жағдайда жарақаттануға әкелуі мүмкін немесе басқа внезапному денсаулығының нашарлауына әкеп соғады. Зиянды фактор - ауруға, жұмысқа қабілеттіліктің төмендеуіне және тіпті өлімге әкеп соқтыратын себеп болып табылатын өндірістік фактор. Қауіп деңгейіне және әсер ету ұзақтығына байланысты зиянды өндірістік фактор қауіпті болуы мүмкін. Кез келген өндірістік процестер, мысалы, ЭЕМ-мен жұмыс істеу немесе басқалар қауіпті және зиянды факторлардың пайда болуымен байланысты. Бұл ретте адам үшін қауіпсіз кернеу-бұл кернеу в 40 В, яғни ПЭВМ-де жұмыс істеу кезінде соққы, электр тогының күйеу қауіпті фактор болып табылады. Зиянды факторларды талдаудан қорғау қажет [12].

4.2 Еңбек жағдайларын талдау

Барлық қажетті жабдықтарды орналастыру үшін ауданы 40 м² болатын үй-жай қажет:

- а) бөлменің өлшемдері - 10x4x3 м;
- б) терезе саны – 3;
- в) есік саны – 2;
- г) жұмыс орындарының саны – 1.

Интерьер:

а) еден бетонды, қалыңдығы 8 мм қоңыр түсті ламинатпен қапталған қоймасы бар үй-жай;

б) еден бетонды, сұр түсті кафельмен жабылған дәнекерлеу станциясы және компьютер;

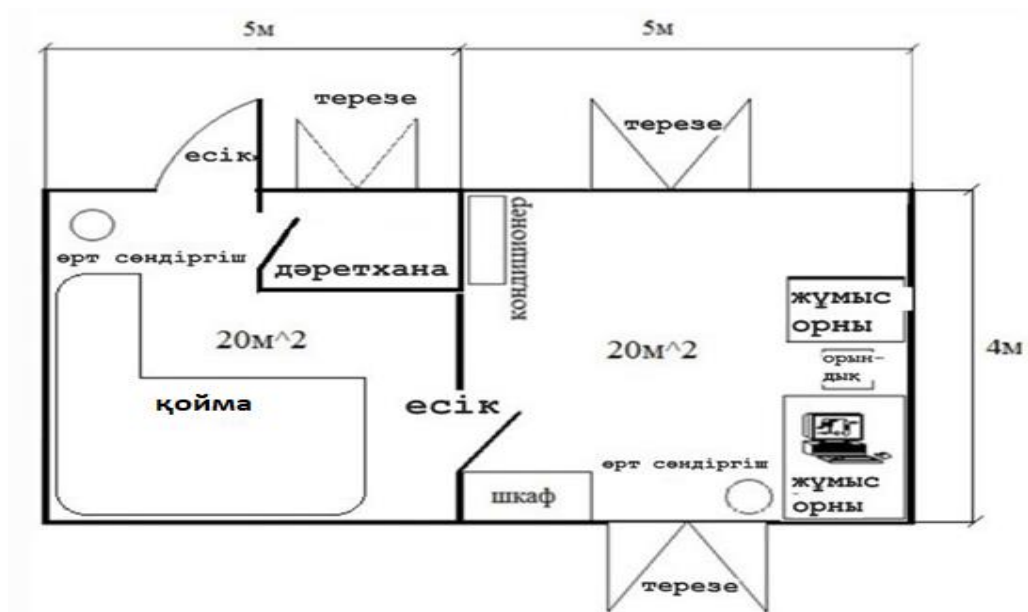
в) қабырғалары сары түсті жиектермен желімделген;

г) төбесі гипсокартон, су эмульсиялық бояумен боялған.

Үй-жай сейсмикалық төзімділікке және өрт қауіпсіздігіне байланысты барлық стандарттарға сәйкес болуы тиіс.

Үзіліс кезінде желдетуді қамтамасыз ету үшін терезелердің орналасуы маңызды, өйткені дәнекерлеу станциясымен жұмыс істеу кезінде ауаға ауыр металдар мен қышқыл қосылыстардың молекулалары бөлінеді.

4.1-суретте жұмыс орнының жоспары көрсетілген.



4.1 сурет – Бөлме жоспары

4.3 Жабдықты және оның оңтайлы орналасуын талдау

Мекемеде бір қызметкер жұмыс істейді. Жұмыс үшін келесі жабдықтар қажет:

- а) Lenovo G 5045 Ноутбук -1 дана;
- б) Gordak 968 дәнекерлеу станциясы.

Ноутбуктардың техникалық сипаттамалары 4.1-кестеде көрсетілген

4.1 кесте - Ноутбуктердің техникалық сипаттамалары

Атауы	Мәндері
Модель	Lenovo G5045
Өлшемі, мм	384 x265 x 25
Салмағы, кг	2.5
Экран жарығы	LED
Диагональ, см	39.624
Аккумулятор	<u>Li-ion</u> 5200 мАч

Электр тогымен зақымдану қаупінің дәрежесі бойынша үй-жай 1-сыныпқа жатады, яғни ол қауіптіліктің жоғары деңгейі бар үй-жайға (құрғақ, ауаның қалыпты температурасы, оқшауланған едені және жерге тұйықталған аспаптардың саны аз) жатпайды [13].

Орындықты компьютер мен дәнекерлеуші станция арасындағы ортаға орналастыру керек, ал компьютер мен дәнекерлеуші станциясы бар үстелдерді екі терезе арасындағы қабырғаға жақын орналастыру керек, табиғи желдету мен жарықтандырудың көп санын және компьютердің қызып кету қаупінің аз болуын қамтамасыз ету үшін. Дәнекерлеу кезінде канифольден немесе дәнекерлеу қышқылынан зиянды заттар бөлінуі мүмкін. Желдету дәнекерлеу процесінде пайда болатын артық жылу мен заттарды жою үшін ауа алмасуды қамтамасыз етуі тиіс. Бұл үшін кондиционер мәжбүрлі желдету режимінде болуы керек.

4.4 Табиғи және жасанды жарықтандыруды есептеу

Жарықтандыру люкспен (Лк) өлшенеді, ал Люмендегі жарық ағынының шамасы (Лм). Жарықтандыруды есептеудің бұл тәсілі дұрыс формадағы үй-жайлар үшін жеткілікті дұрыс болып табылады.

Жарық ағынының қажетті шамасын мына формула бойынша есептейміз:

$$X*Y*Z, \quad (4.1)$$

мұндағы, X – ҚНЖЕ мен сәйкес объектіні жарықтандыру нормасы, Лк;

Y – үй-жайдың ауданы, м²;

Z – төбенің биіктігіне түзету коэффициенті.

Бұл жағдайда жарықтану нормасы сәйкесінше 300, 50 және 50 Лк тең болады, яғни компьютермен жабдықталған үй-жайдың, қоймалық үй-жайдың және санитарлық тораптың оңтайлы жарықтандырылуы ҚНЖЕ-ге сәйкес.

Үй-жайдың ауданы компьютермен және дәнекерлеуші станциясы бар үй-жайда 20 м², қойма үй-жайында 18 м² және санитарлық торапқа 2 м² тең.

Барлық үй-жайлар үшін түзету коэффициенті 3м жоғары емес төбеге арналған ҚНЖЕ-ге сәйкес 1,2 тең болады. Бұл жағдайда төбенің биіктігі шамамен 2997 мм-ге тең.

4.2 кестеде жарық ағынының қуаттарының және оларға балама көрсеткіштерінің мәндері келтірілген.

4.2 кесте - Қуат көрсеткішіне байланысты шамның жарық ағынының мәні

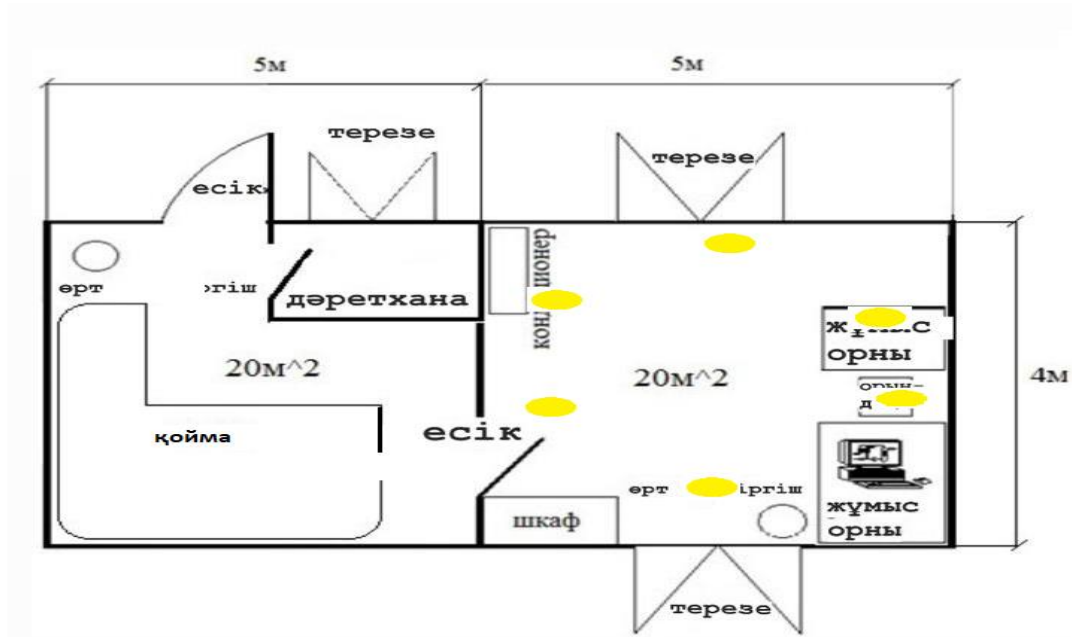
Қуат номиналы, Вт	Жарық ағының мәні, Лм
3-4	250-300
4-6	300-450
6-8	450-600
8-10	600-900
10-12	900-1100
12-14	1100-1250
14-16	1250-1400

Бөлмедегі қажетті жарықтандыруды есептеуден басқа әрбір жарық көзі үшін оптималды орынды таңдау қажет. Дұрыс орналасқан жарық көздері жұмыста қолайлы.

Компьютермен және дәнекерлеу станциясымен үй-жай үшін жарық ағынының қажетті мөлшерін есептеу:

$$X*Y*Z = 300 * 20 * 1,2 = 7200 \text{ [Лм]}.$$

Алынған нәтижелерге сүйене отырып, суретте 4.2 ұсынылған оптималды орналасуы алты шамдарды 12-14 Вт шамасына әрбір жарық ағымы 1200 Лм тең.

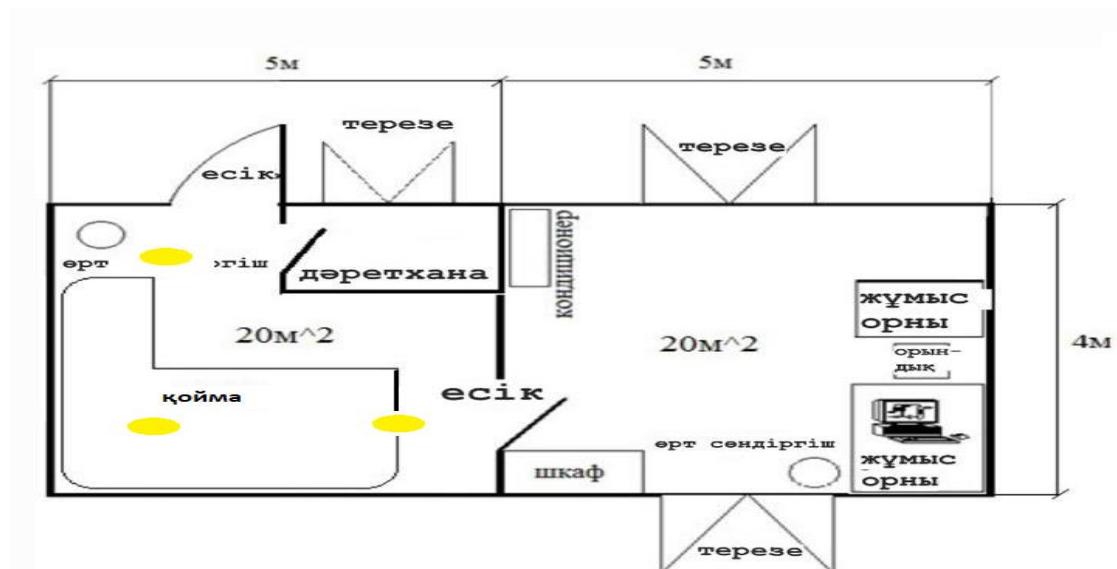


4.2 сурет – Жұмыстық үй-жайларды жарықтандыру

Қойма үй-жайы үшін:

$$X*Y*Z = 50*18*1,2 = 1080 \text{ [Лм]}.$$

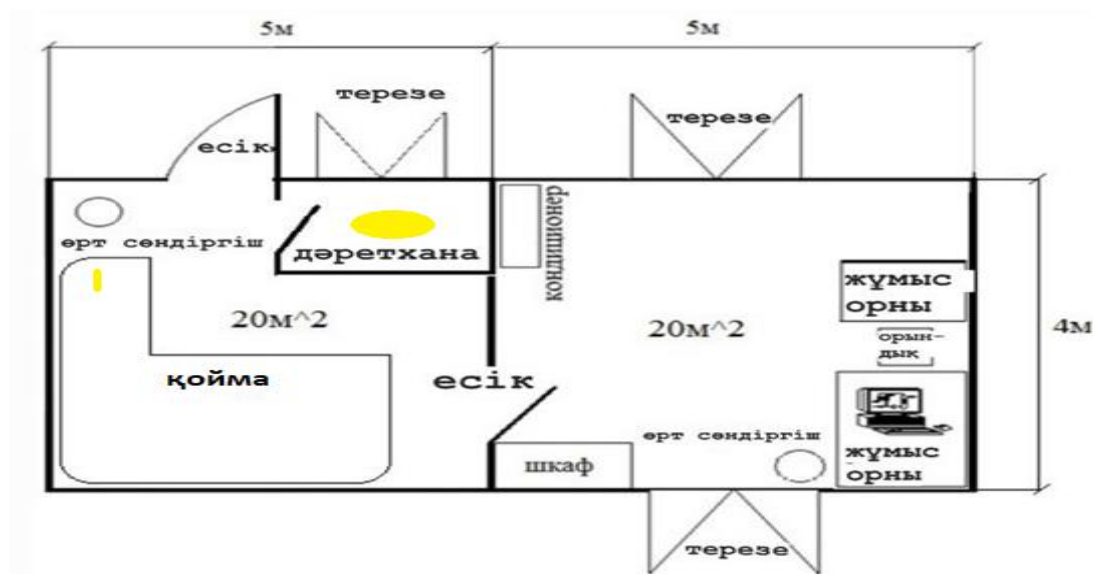
4.3 - суретте әрбір жарық ағынының шамасы 350-400 Лм бар 4-6 Вт үш оптималды шамдар орналасқаны ұсынылған.



4.3 сурет - Қойма үй-жайын жарықтандыру
Жуынатын бөлмесі үшін жарықтандыруды есептеу:

$$X*Y*Z = 50*2*1,2 = 120 \text{ [Лм]}.$$

4.4 суретте көрсетілген, яғни бір шам үшін 3 Вт-тық 250 Лм жеткілікті.



4.4 сурет - Санузелді жарықтандыру

Жарық көздерінің дұрыс орналасуын және олардың санын ескеру қажет.

4.5 Микроклимат және көру жүктемесі

4.2 кестеде МСТ 12.0.003-88 сәйкес оңтайлы микроклиматтық шарттар көрсетілген. ССБТ орташа ауырлықтағы дене жұмысына арналған. Үй-жайда жаз мезгілінде температура +28°C дейін, қыста температура +18-ден +20°C дейін көтеріледі.

Микроклиматтың қолданыстағы параметрлерін салыстыра отырып, орташа ауырлықтағы микроклиматтық жағдайларды талдай отырып, жазғы уақытта жылу артық екені анық болады. Сондықтан, оператор бөлмесіндегі микроклимат жағдайын қолдау үшін оны кондиционерлеу жүйесімен жабдықтау қажет. 4.3-4.6 кестелерінде барлық стандарттар көрсетілген.

4.3 кесте - Микроклимат параметрлерінің нормалары

Жұмыс периоды	Жұмыс категориясы	Температура, °C	Ауа қозғалысының жылдамдығы кем емес м/с
Суық	I а	18-20	0,2
Жылы	I а	22-24	0,3

Жұмыс тобына орташа ауырлық дәрежесімен, яғни ұсақ бөлшектермен жұмыс күні 8 сағатты құрайды, күніне немесе аптасына 40 сағат. Зиянды әсерлерге монитордан сәуле шығару және дәнекерлеу станциясымен бөлінетін жылу жатады.

Жұмыстық беттіктің биіктігі: 730 мм, отырғыш биіктігі: 410 мм (ГОСТ 12.2.032-78), 4.3 кестеде мәндер берілген.

4.4 кесте – Жұмыс түрі (ГОСТ 12.2.032-78)

Жұмыс атауы	Жұмыс кластары	Жұмыскер жынысы	Жұмыс орнын ұйымдастыру кезіндегі жұмыс бетінің биіктігі	Орындық биіктігі
Орта ауырлық	Ia классы (отыратын жағдайда орындалатын жұмыс)	Ер, әйел	710 мм	410 мм

Жұмыс процесінде ажыратылатын объектілердің көлемі: 5 м, объектіден қызметкердің көзіне дейінгі арақашықтық: 520 мм - көру жұмысының разряды: V (СНиП РК 2.04.-0.5-2002).

4.5 кесте – Көру жұмысының разряды (СНиП РК 2.04.-0.5-2002)

Ажырата алатын объектінің минималды мөлшері	Жұмыскердің көз мен объектіге дейін арақашықтық	Көру жұмысының разряды
1-10 мм	520 мм	V

4.6 кесте - Ағзаның энергия шығындары бойынша жұмыс санаттары

Жұмыс, Категориясы	Дененің энергия шығындары, Дж/с (ккал/час)	Сипаттамасы
Физикалық орта ауырлық, IIa	172 – 232	Жүрумен байланысты, тұрып немесе отырып орындалатын, бірақ ауыр заттарды ауыстыруды талап етпейтін

4.6 Кондиционерлеу жүйесін есептеу

Бөлмедегі кондиционерлеу жүйесін есептейміз. Кондиционерлеу бөлмедегі микроклиматтың ең жақсы жай-күйін және дәл және сезімтал аппаратураның жұмыс жағдайын қамтамасыз етеді және ҚНЖЕ 11-33-75 "Ауаны жылыту, желдету және кондиционерлеу" тарауына сәйкес орындалуы тиіс [14].

$Q_{арт}$ жылудың артығын жою үшін оператордың үй-жайынан бір сағат ішінде шығару қажет ауаның L м³/с мөлшерін анықтаймыз, мына формула бойынша анықталады:

$$L = \frac{Q_{арт}}{C_B * t * u_B}, [м^3/сағ], \quad (4.1)$$

мұндағы, C_B – құрғақ ауаның жылу сыйымдылығы, ккал/кг ($C_B=0,24$ ккал/кг град);

u_B – температураға байланысты анықталатын кететін ауаның тығыздығы, кг/м³ (есеп кезінде қабылданады $u_B=1,20$ кг/м³).

Артық жылуды анықтаймыз $Q_{арт}$, ккал/сағ:

$$Q_{арт} = Q_{п} - Q_{от}, \quad (4.2)$$

мұндағы, $Q_{п}$ – бөлмеге келіп түсетін ауадан бөлінетін жылу, ккал/ч;

$Q_{от}$ – сыртқы қоршаулар арқылы қоршаған ортаға жылу беру (жылдың жылы мезгілінде, есептеу кезінде нөлге қабылдауға болады).

Q_{Π} жылу бөлінуінің саны жабдықтың қуатына, жұмыс істейтін адамдардың санына және үй-жайға терезе ойықтары арқылы енгізілетін жылуға байланысты:

$$Q_{\Pi} = Q_{\text{об}} + Q_{\text{л}} + Q_{\text{р}} + Q_{\text{ио}}, \quad (4.3)$$

мұндағы, $Q_{\text{об}}$ – өндірістік жабдықтармен бөлінетін жылу, ккал/сағ;

$Q_{\text{ио}}$ – жасанды жарықпен бөлінетін жылу;

$Q_{\text{л}}$ – адамдардың жылу бөлінуі, ккал/сағ;

$Q_{\text{р}}$ – күн радиациясымен енгізілетін жылу, ккал/сағ.

Үй-жайдағы жасанды жарықпен бөлінетін жылу ара қатынасымен анықталады:

$$Q_{\text{ио}} = 1000 \cdot N, \quad (4.4)$$

мұндағы, N – шамдардың шығыс қуаты.

$$Q_{\text{ио}} = 1000 \cdot 0,24 = 240 \text{ Вт} = 0,24 \text{ кВт} \cdot 860 = 206 \text{ [ккал/сағ]}.$$

Өндірістік бөлмедегі адамдар бөлетін жылу, орташа ауырлықтағы жұмыс категориясы бойынша бір адам бөлетін жылу 195 Вт құрайды.

Осыдан аламыз:

$$Q_{\text{л}} = 1 \cdot 195 = 195 \text{ Вт} = 0,195 \text{ кВт} \cdot 860 = 167,7 \text{ [ккал/сағ]}.$$

Үй-жайдағы өндірістік жабдықпен бөлінетін жылу арақатынасымен анықталады:

$$Q_{\text{об}} = 860 \cdot P_{\text{об}} \cdot n, \quad (4.5)$$

мұндағы, 860 – жылулық эквивалент 1 кВт·сағ, яғни жылу;

$P_{\text{об}}$ – жабдық тұтынатын қуат $P_{\text{об}} = 10 \text{ кВт}$;

n – жылудың үй-жайға өту коэффициенті, $n = 0,75$.

$$Q_{\text{об}} = 860 \cdot 10 \cdot 0,75 = 6450 \text{ [ккал/сағ]}.$$

Шынылау арқылы күн сәулесінен жылу:

$$Q_{\text{р}} = m \cdot F(q^{\text{I}} + q^{\text{II}}) \beta \cdot K_1 \cdot K_2, \text{ [Вт/м}^2\text{]}, \quad (4.6)$$

мұндағы, m – үй-жайдағы терезелер саны;

F – терезе ауданы, $2,4 \text{ м}^2$;

q' , q'' – географиялық ендігіне байланысты тәулікті есептік саны

үшін тікелей және шашыраңқы жылу ағындары;

$q' = 73$, $q'' = 77$ батыс жағы үшін;

$q' = 288$ және $q'' = 85$ оңтүстік жағы үшін;

β - жылу өткізу коэффициенті, $\beta=0.15$;
 K_1 – жарық ойықтарының қараңғылану коэффициенті, $K_1=0.65$;
 K_2 – вертикалды шынылауға арналған шынылаудың ластану коэффициенті, $K_2=0.95$.

Батыс жағы үшін:

$$Q_{\text{бат}} = 2 * 2,4 * (73+77) * 0,15 * 0,65 * 0,95 = 66,69 \text{ [Вт]}.$$

Оңтүстік жағы үшін:

$$Q_{\text{оңт}} = 1 * 2,4 * (288+85) * 0,15 * 0,65 * 0,95 = 82,92 \text{ [Вт]}.$$

Жиынтықта-күн сәулесінен шынылау арқылы тең жылу:

$$150 \text{ Вт} = 0,15 \text{ кВт} * 860 = 129 \text{ [ккал/сағ]}.$$

Сонда жылу бөлу болады:

$$Q_{\text{арт}} = Q_{\text{п}} = 6450 + 168 + 123 + 206 = 6953 \text{ [ккал/сағ]}.$$

Қажетті ауа алмасу мына формула бойынша есептеледі (4.1):

мұндағы, t - ауа кернеулігіне байланысты жылу $Q_{\text{н}}$.

ГОСТ 12.1.014-84 Жұмыс аймағының ауасы:

$$Q_n = \frac{Q_{\text{арт}}}{Q_{\text{н}}} \text{ [ккал/м}^3\text{]}, \quad (4.7)$$

мұндағы, $Q_n \leq 20 \text{ ккал/м}^3$, $\Delta t = 6^\circ \text{C}$;
 $Q_n \geq 20 \text{ ккал/м}^3$, $\Delta t = 8^\circ \text{C}$.

$$Q_n = \frac{6953}{120} = 57,94 \text{ [ккал/м}^3\text{]},$$

Сонда, $t = 8^\circ \text{C}$;

$$L = \frac{6953}{0,24 * 8 * 1,20} = 3018 \left[\frac{\text{м}^2}{\text{сағ}} \right].$$

Үй-жайға бір сағат ішінде келетін ауа санының үй-жайдың көлеміне қатынасы ауа алмасу еселігі деп аталады:

$$K = \frac{L}{V_n} \quad , (4.8)$$

мұндағы, V_n – үй-жайдың көлемі $V_{\text{п}} = 120 \text{ м}^3$;

$$K = \frac{3018}{120} = 25.15.$$

4.7 Бөлімнің қорытындысы

Бөлмеде жабдықпен қолдану мақсатында вентиляциялану дұрыс болу үшін, бөлмеге алынған мәліметтер бойынша барлық техникалық талаптарға жауап беретін кондиционерді қою керек керек. Vestelgreenplus 12000 СН кондиционері осы талаптарға сай келеді.

Келесі сипаттамалары бар:

- а) салқындату режиміндегі қуат: 6900 Вт;
- б) ең жоғары ауа ағыны: 4000 м³/ч;
- в) өлшемдері (ШхВхГ): 880×280×203 мм;
- г) екі бағытта ауа беру;
- д) шудың төмен деңгейі;
- е) жылдам суық және жылыту режимі.

5 Техникалық-экономикалық негіздеме

5.1 Жобаның мақсаттары мен міндеттері

Дауыстық басқару жүйесі-модульдер мен жеке элементтердің жиынтығы, оны жинау және бағдарламалау кезінде 220В жүктемеге немесе басқа да аз жүктемеге қосылған құрылғыларды бақылау мүмкіндігі пайда болады. Қазіргі дауыстық басқару жүйесі кең таралуда, өйткені енді стандартты қосқыштарға/ажыратқыштарға жүгінбей, үйде техниканы бақылауға мүмкіндік бар. Дауыстық басқару жүйесінің үздіксіз дамуы болашақта техника мен онымен жұмыс істеуге қолжетімділікті барынша жеңілдетуге мүмкіндік береді, алайда адам тарапынан толық бақылауды болдырмаудың қажеті жоқ. Егер жүйе істен шықса немесе дұрыс жұмыс істемесе, үйде, өндірістік бөлмеде қарапайым тумблерлер орната отырып, техниканы басқару органдарын дұрыс қайталайтын болады.

Arduino - екі негізгі компоненттен тұратын аппараттық есептеу платформасы: енгізу-шығару платасы және Processing/Wiring тілінде өңдеу ортасы. Arduino жаңадан келгендерге де, кәсіби мамандарға да электрондық құрылғыларды жасау үшін ыңғайлы. Бұл платформа қарапайым бағдарламалау тілі, ашық архитектурасы және бағдарламалық коды үшін бүкіл әлемде кең танымал. Бұл платформаның ерекшелігі, ол USB арқылы программаторларды пайдаланбай ол бағдарлана береді [15].

Arduino Uno контроллері ATmega328 –да салынған. Платформада 14 сандық кіріс/шығыс бар (оның 6 ШИМ шығу ретінде пайдаланылуы мүмкін), 6 аналогтық кіріс, 16 МГц кварц генераторы, USB қосқышы, күш қосқышы, ICSP қосқышы және қайта жүктеу түймесі бар. Жұмыс істеу үшін платформаны компьютерге USB кабелі арқылы қосу немесе AC/DC адаптерінің немесе батареяның көмегімен қорек беру қажет.

Сипаттамалары:

- Микроконтроллер ATmega328;
- Жұмыс кернеуі 5 В;
- Кіріс кернеуі (ұсынылатын) 7-12 В;
- Кіріс кернеуі (шектеу) 6-20 В;
- Сандық кіріс / Шығу 14 (олардың 6 ШИМ шығу ретінде пайдаланылуы мүмкін);
- Аналогтық кіріс 6;
- Кіріс/шығыс арқылы тұрақты ток 40 мА;
- 3.3 В шығару үшін тұрақты ток 50 мА;
- Флеш-жады 32 Кб (ATmega328) оның 0.5 Кб жүктеуші үшін қолданылады;
- ОЗУ 2 Кб (ATmega328);
- EEPROM 1 Кб (ATmega328);
- Тактикалық жиілігі 16 МГц.

5.2 Шығындар жоспарын анықтау

Аспаптың өзіндік құнын анықтау және дайындау үшін шығаруға байланысты барлық шығындарды табу және оған қосу қажет.

Жоспарлы өзіндік құнның калькуляциясы мынадай баптар бойынша жасалады:

Өндірістік шығыстар:

- а) шикізат пен материалдар;
- б) сатып алынатын жиынтықтаушы бөлшектер;
- в) электр энергиясы мен отынға арналған шығыстар;
- г) негізгі жалақы;
- д) қосымша жалақы;
- е) Әлеуметтік салық аударымдары;
- ж) тасымалдауға арналған шығыстар;
- и) өндірістік алаңның жалдау ақысы.

5.3 Аспапты әзірлеуге арналған шығындарды есептеу

Шығындарды есептеу мынадай формула бойынша анықталады:

$$C = ETK + Oc + Ao + Pm + Pk + Ээ + Pa, \quad (5.1)$$

мұндағы, ЕТҚ – еңбекақы төлеу қоры;

Oc - әлеуметтік салық аударымдары;

Ao - амортизациялық аударымдар;

Pm - материалдарға арналған шығыстар;

Pk - жинақтаушы шығыстар;

Ээ - электр энергиясына арналған шығыстар;

Pa - жалдау шығыстары.

5.3.1 Еңбекақы төлеу қоры

Негізгі және қосымша жалақыдан еңбек ақы қоры қалыптасады:

$$ETK = Z_{ocn} + Z_{доп}, \quad (5.2)$$

мұндағы, Z_{ocn} – негізгі еңбек ақы;

$Z_{доп}$ – қосымша еңбек ақы.

Орындаушылардың негізгі жалақысы мына формула бойынша анықталады:

$$Z_{ocn} = C_{cp} * t, \quad (5.3)$$

мұндағы, C_{cp} – тапсырманы орындаушы жұмыскердің күндізгі еңбек ақысы (6819 тг./күн.адам);

t – аспапты әзірлеудің еңбек сыйымдылығы (күн.адам).

Күндік еңбекақысы негізге алына отырып айқындалады айлық жалақысы әзірлеуші және жұмыс айдағы күн (орташа есеппен қабылдауға болады 22 жұмыс күні). Әзірлеуге тартылған қызметкерлер бойынша мәліметтерді 5.1 кесте түрінде ұсыну қажет.

5.1 кесте – Жоба бойынша кірістірілген, қызметкерлері бойынша мәліметтер

Маман – Орындаушы	Адам саны	Айына өтем ақы, тенге
1 Программист	1	150 000
2 Барлығы	1	150 000

Осылайша, бағдарламашының күндізгі өтем ақысын құрайды:

$$З_{дн} = 150000/22 = 6819 \text{ [тг]}.$$

Нақты міндет үшін бағдарламалық өнімді әзірлеудің еңбек сыйымдылығын (t_p) әзірлеу кезеңдері бойынша еңбек шығынының сомасы ретінде қарастыруға болады:

- Тапсырманың сипаттамасын дайындау – t_o , адам/сағ;
- Есепті шешу алгоритмін әзірлеу – t_a , адам/сағ;
- Алгоритмнің блок-сұлбасын құру – t_b , адам/сағ;
- Бағдарламану – t_p , адам/сағ;
- ЭЕМ-де бағдарламаны жөндеу – $t_{отл}$, адам/сағ;
- Тапсырма бойынша құжаттама дайындау – t_d , адам/сағ.

Әрбір жұмыс түрінің еңбек сыйымдылығы мынадай формула бойынша анықталады:

$$T_i = \frac{3T_{min} + 2T_{max}}{5} \quad (5.4)$$

мұндағы, T_{min} – жоспар бойынша жеке жұмыс түрін орындаудың ең аз мүмкін еңбек сыйымдылығы;

T_{max} – практикада жұмыстың жеке түрін орындаудың барынша мүмкін болатын еңбек сыйымдылығы.

Жұмыстың әрбір түрінің ұзақтығы күнтізбелік күнмен (t_i) мына формула бойынша күнмен анықталады:

$$t_i = \frac{T_i}{\chi_i} * K_{шығ} \quad (5.5)$$

мұндағы, T_i – жұмыстың еңбек сыйымдылығы, адам-күн;

χ_i – орындаушылар саны, адам;

$K_{шығ}$ – демалыс және мереке күндерін ескеретін коэффициент;

$$K_{шығ} = \frac{K_{кал}}{K_{раб}}, \quad (5.6)$$

мұндағы, $K_{кал}$. – күнтізбелік күндер саны;
 $K_{раб}$. – жұмыс күндері.

2020 жылғы өндірістік және салық күнтізбесіне сәйкес, жұмыс күндерінің саны 245 күнді құрайды, мереке күндерінің саны – 21, осылайша: $K_{шығ}=2.1$

Шығындар бойынша жұмыс кезеңдерінің толық тізімі, сараптамалық бағалау және олардың еңбек сыйымдылығының есептік шамасы, сондай-ақ әрбір жұмыс түрінің ұзақтығы 5.2 кестеде көрсетілген).

5.2 кесте - Сөйлеуді тану құрылғысын әзірлеудің кезеңдері мен еңбек сыйымдылығы

№ жұмыстар	Жасау кезеңдері	Адамның еңбексыйымд			Жұмыскерлер саны, адам	Жұмыстың ұзақтылығы, күнтізбелік күндер
		T_{min}	T_{max}	T_i	$Ч_i$	t_i
1	2	3	4	5	6	7
Техникалық тапсырма						
1	Тапсырманы қою талаптары	1	1	1	1	1
2	Материалдарды жинау және қолданыстағы әзірлемелерді талдау	2	2	2	1	3
3	Әдебиеттерді таңдау	6	7	6	1	8
4	Жүйеге қойылатын талаптарды анықтау	2	3	2	1	3
5	Әзірлеу және дайындау кезеңдерін, кезеңдері мен мерзімдерін айқындау	2	3	2	1	3
Эскиздік жоба						

5.2 кестенің жалғасы

6	Аспапты дайындау бойынша ақпаратты талдау	7	8	7	1	9
7	Аспаптың функционалдық схемасын әзірлеу	3	5	4	1	5
8	Эл дауысын басқару құрылымын әзірлеу. жабдық	4	6	5	1	7
9	Құжаттандыру	1	3	2	1	3
Техникалық жоба						
10	Басқару бағдарламасына қойылатын талаптарды анықтау	3	3	3	1	3
11	Аспаптық құралдарды таңдау	1	1	1	1	1
12	Аппараттық қамтамасыз етудің қасиеттері мен талаптарын анықтау	2	3	2	1	3
Жұмыстық жоба						
13	Бағдарламалау	5	7	6	1	8
14	Басқару бағдарламасын тестілеу және жөндеу	10	15	12	1	17
15	Жүйенің жұмысқа қабілеттілігін келісу және бекіту	9	12	10	1	14
Ендіру						
16	Тәжірибелік пайдалану	5	7	6	1	8
17	Пайдалану нәтижесінде алынған деректерді талдау	4	5	5	1	6
18	Сынақ нәтижелері бойынша техникалық құжаттаманы түзету	3	3	3	1	4

5.2 кестенің жалғасы

19	Жалпы еңбек сыйымдылығы	-	-	78	-	106
----	-------------------------	---	---	----	---	-----

5.1 кестедегі келтірілген жұмыстарының барлық кезендері тізбекті орындалған кезде, демек, бір жолғы жұмыстарды жүргізудің жалпы ұзақтығы 106 күндік күнді құрайды, ал еңбек сыйымдылығы 78 сағ/күн.

$$Z_{нег} = 6819 * 78 = 531\,882 \text{ [тг]}.$$

Демалуға және ауруға кететін уақыт шығынын ескеретін орындаушылардың қосымша жалақысы (негізгі еңбекақыдан орташа алғанда 10% қабылданады);

$$Z_{доп} = 10\% * Z_{осн}, \quad (5.7)$$

Осыдан,

$$Z_{қос} = 0,1 * 531\,882 = 53\,188 \text{ [тг]}.$$

Алынған деректер негізінде,

$$\text{ФОТ} = 531\,882 + 53\,188 = \mathbf{585\,070} \text{ [тг]}.$$

5.3.2 Әлеуметтік салық

Соц. әлеуметтік салық зейнетақы аударымдарын алып тастағанда еңбекақы төлеу қорының 11% мөлшерінде есептеледі. ФОТ-қа зейнетақы аударымы C_n 10% құрайды [16].

Әлеуметтік салық – с 01.01.18 в РК – 9,5%.

Сбор соцобеспечения – 1,5%.

ЕТҚ-дан ҚР 10% зейнетақы есебін құрайды.

$$C_{соц} = (\text{ФОТ} - C_n) * 11\%, \quad (5.8)$$

мұндағы, $C_{соц}$ – әлеуметтік салық;

C_n – зейнетақы фондынан аударымдар.

$$C_n = (\text{ФОТ} * 0,1), \quad (5.9)$$

$$C_n = 585\,070 * 10\% = 58\,507 \text{ [тг]},$$

$$C_{соц} = (585\,070 - 58\,507) * 11\% = \mathbf{57921,93} \text{ [тг]}.$$

5.3.3 Амортизация

Амортизациялық аударымдар A_o формула бойынша есептелінеді:

$$A_o = (H * \Phi * 100\% * 78) / 360, \quad (5.10)$$

мұндағы, Н - амортизацияның орташа жылдық нормасы (Мұнай-газ өндіру машиналары мен жабдықтарын, сондай-ақ ақпаратты өндеуге арналған компьютерлер мен жабдықтарды қоспағанда, машиналар мен жабдықтар - 25%);

Φ - жабдықтың бастапқы құны, тенге.

Жобалау және әзірлеу үшін қажетті жабдықтың құны 170 000 теңгені құрайды (5.3 кесте)

5.3 кесте – Әзірлеу және жобалау үшін пайдаланылатын жабдықтар

Құрал-жабдықтар	Бірлік бағасы тг.	Мөлшері	Сомасы тг.
Дәнекерлеу станциясы	19000	1	19000
Қысқыш (үшінші қол)	3000	1	3000
Ноутбук	110000	1	110000
Ұзартқыш	1000	2	2000
Кондиционер	15000	1	15000
Жиһаз жиынтығы	18000	1	18000
Өрт сөндіргіш	1500	2	3000
Қорытынды			170000

$$A_o = (0,25 * 170000 * 78) / 360 = 9208 \text{ [тг]}.$$

5.3.4 Материалдар мен жинақтаушылардың шығыны

"Материалдар және жинақтаушылар" тармақшасына құнынан процентпен белгіленетін көлік шығындарын ескере отырып, материалдарға арналған шығындар енгізіледі (5.4 кесте).

5.4 кесте – Материалдар шығыны

Материалдық ресурстар	Өлшеу бірлігі	Саны	Бірлік үшін бағасы тг.	Сомасы тг.
Дәнекер оловян	гр.	100	600	600
Канифоль	дана	1	200	200
Макеттік төлем	дана.	1	300	300

5.4 кестенің жалғасы

Дәнекерлеу қышқылы	гр.	30	7	210
Қорытынды				1310

Сондай-ақ "Материалдар және жинақтаушылар" тармақшасына бағасынан пайызбен белгіленетін көлік шығындарын ескере отырып, жинақтауыштарға арналған шығындар енгізіледі (5.5 кесте).

5.5 кесте – Жинақтаушы шығындары

Аталуы	Тип	Мөлшері	Бірлік үшін бағасы тг.	Сомасы тг.
Дауыспен басқару модулі V3	модуль	1	17000	17000
Микрофон	динамикалық	1	1000	1000
Arduino UNO R3	микроконтроллер	1	4200	4200
Реле 1 канал, 5В	Аралық	1	800	800
Сымдар	Папа-папа	40	15	600
Сымдар шлейфі	Папа-мама	40	15	600
Вентилятор 220В	тұрмыстық	1	1500	1500
Қорытынды:				25700

Жинақтаушылардың құны - 25700 тг.

5.3.5 Электр энергиясына шығындар

Өндірістік қажеттіліктер (Ээ) тарапынан электр энергиясына ақы төлеу шығындары жабдық тұтынылатын қуатқа, кәсіпорынның жұмыс уақыты мен электр энергиясына тарифтерге байланысты анықталады [12].

$$Зэ = M \cdot k_z + T [C_k B_t + \text{ч}], \quad (5.11)$$

мұндағы, M – ЭЕМ қуаты, кВт;

k_z – жүктеме коэффициенті (0.8);

$C_k B_t \cdot \text{ч}$ – 1 кВт-сағаттағы электр энергия бағасы, тенге/кВт-сағ;

T – жұмыс уақыты, сағат (328с.).

$$Зэ = 0.24 \times 0.8 \times 328 \times 23 = 1449 \text{ [тенге/мес]}.$$

5.3.6 Үйді жалға алу шығындары

Жалға алынатын үй-жайдың ауданы – 40 м². Алматы қаласындағы жалдау ақысы 1м² үшін коммуналдық қызметтер айына орташа есеппен 1500 теңгені құрайды.

Жалға алу шығындарын формула бойынша есептеу

$$P_a = T \cdot P_a, \quad (5.12)$$

мұндағы, P_a – жалға алу бағасы, тг/ай;

T – жалға алу мерзімі, ай.

$$P_a = 60000 \cdot 2 = 120000 \text{ [тг]}.$$

Аспапты жасау бағасы сонымен, құрайды:

$$\begin{aligned} C &= 585\,070 + 57921,93 + 9208 + 1310 + 25700 + 1449 + 120000 = \\ &= 800\,658,93 \text{ [тг]}. \end{aligned}$$

$K_{\text{пр}i}$ - күрделі салымдар бойынша i -ші нұсқа инвестициялық жобаның теңге. Жобаны жүзеге асыру үшін қажетті күрделі салымдар жабдықты сатып алуға және өнімді бағдарламалық қамтамасыз етуді әзірлеуге арналған салымдарды қамтиды:

$$K_{\text{пр}} = K_{\text{об}} + K_{\text{пп}}, \quad (5.13)$$

мұндағы, $K_{\text{пр}}$ – жобаны іске асыру үшін қажетті күрделі салымдар;

$K_{\text{об}}$ – жабдыққа қажетті күрделі салымдар;

$K_{\text{пп}}$ – өнімнің бағдарламалық қамтамасыз етуінің капитал салымдары.

Жабдыққа күрделі салымдарды мына формула бойынша есептеу керек:

$$K_{\text{об}} = C_{\text{об}} + 3\text{тр} + 3\text{м}, \quad (5.14)$$

мұндағы, $K_{\text{об}}$ – жабдыққа күрделі салымдар;

$C_{\text{об}}$ – жабдықтың құны;

3тр – тасымалдауға арналған шығындар;

3м – монтажға арналған шығындар.

Тасымалдау шығындары шамамен жабдықтың жалпы құнының 5-10% құрайды және мынадай формула бойынша есептеледі:

$$3\text{тр} = C_{\text{об}} \cdot 0.05, \quad (5.15)$$

Монтаждық жұмыстарды жүргізуге арналған шығындарды жабдықтың құнынан 5-тен 7% - ға дейінгі аралықтан алу қажет, мына формула бойынша есептеледі:

$$З_m = C_{об} \cdot 0.05, \quad (5.16)$$

Жобаны іске асыру үшін қажетті жабдықтың жалпы құны 101 600 теңгені құрайды. Қажетті жабдықтар тізбесі және олардың құны 5.6 кестеде көрсетілген.

5.6 кесте - Өндіруші жобаны іске асырудың бірінші нұсқасы үшін жабдықтың құны (Paradox)

Аталуы	Мөлшері, дана	Бірлік үшін бағасы (тг.)	Барлығы (тг.)
Бақылау және басқару пульті	1	25 300	25 300
Инфрақызыл датчигі	2	7700	15 400
Реле	2	6500	13 000
DS18B20 датчик температурысы	1	7 000	7 000
Микроконтроллер	2	11 200	22 400
Bluetooth-модуль HC-06	1	11 300	11300
Дыбыс датчигі	1	7 200	7 200
Барлығы:			101600

(5.15) және (5.16) формулаларға сәйкес тасымалдау және монтажда жұмсалатын шығындар:

$$З_{тр} = 127\,100 \cdot 0.05 = 6355 \text{ [тг]},$$

$$З_m = 127\,100 \cdot 0.04 = 5\,084 \text{ [тг]}.$$

Осылайша, формулаға сәйкес жобаны іске асырудың нұсқасы үшін жабдыққа күрделі салымдар (5.14) құрайды:

$$C_{об} = 101\,600 + 6355 + 5084 = 113\,039 \text{ [тг]}.$$

Жеке баптар бойынша алынған деректер негізінде 5.6 кестеде келтірілген нысан бойынша бағдарламалық өнімді әзірлеуге арналған шығындар сметасы жасалады.

5.1 формуласы бойынша күрделі салымдар:

$$K = 800\,658,93 + 113\,039 = 913\,697,93 \text{ [тг]}.$$

Жеке баптар бойынша алынған деректер негізінде 5.6 кестеде келтірілген нысан бойынша бағдарламалық өнімді әзірлеуге арналған шығындар сметасы жасалады.

5.7 кесте - Бағдарламалық өнімді әзірлеуге кеткен шығындарды есептеу

Шығындар баптары	Шартты белгілер	Сомасы, тенге	Жалпы сомадан пайызбен
Еңбекақы төлеу қоры	З _{фот}	779 922	76,8%
Әлеуметтік салық	З _{сз}	77 212,3	7,6%
Материалдар	З _м	25 700	1,5%
Амортизациясы	З _а	9 208	0,9%
Электр энергиясы	З _э	2755	0,2%
Үстеме шығыстар	З _н	120 000	13%
Қорытынды:		1 014797,3	100%

5.4 Динамикалық көрсеткіштерді есептеу

Жобаны енгізуден күтілетін жылдық экономикалық әсердің шамасы мына формула бойынша есептеледі:

$$\mathcal{E}_e = \mathcal{E}_{ye} - K \times E_n, \quad (5.17)$$

мұндағы, \mathcal{E}_e – күтілетін жылдық экономикалық тиімділік, тенге;

\mathcal{E}_{ye} – күтілетін шартты-жылдық экономия, тенге;

K – күрделі салымдар, тенге;

E_n – капиталдық салымдардың экономикалық тиімділігінің нормативтік коэффициенті.

Шартты жылдық үнем мына формула бойынша есептеледі:

$$\mathcal{E}_{yt} = \mathcal{E}_1 - \mathcal{E}_2, \quad (5.18)$$

мұндағы, \mathcal{E}_{yt} – күтілетін шартты-жылдық үнем, тенге;

\mathcal{E}_1 – жүйені енгізгенге дейінгі пайдалану жылдық шығындар;

\mathcal{E}_2 – енгізгеннен кейінгі пайдалану жылдық шығындар.

Жүйені енгізгенге дейінгі пайдалану жылдық шығындар мынадай формула бойынша есептеледі:

$$\Xi_1 = \text{ФОТ} + C_n, \quad (5.19)$$

мұндағы, ФОТ- еңбекақы төлеу қоры;

C_n - әлеуметтік салық.

5.8 кесте - Жобаға қатысатын қызметкерлер туралы мағлұматтар

Маман	Адам саны	Айлық табыс, тг	Жылғы табыс
Күзетші	3	70 000	2 520 000
Парковщик	2	60 000	1 440 000
Қорытынды		330 000	3 960 000

Еңбекақы төлеу қоры мына формула бойынша есептеледі:

$$\text{ФОТ} = Z_{\text{осн}} + Z_{\text{доп}}, \quad (5.20)$$

мұндағы, $Z_{\text{осн}}$ - қызметкерлердің негізгі жалақысы;

$Z_{\text{кос}}$ - негізгі еңбекақының 20% - ын құрайтын қосымша жалақы.

$$\text{ФОТ} = 3\,960\,000 + (3\,960\,000 \times 0.2) = 4\,752\,000 \text{ [тг]}.$$

Әлеуметтік салық 11% құрайды жұмыскедің табысынан, және осы формуламен есептелінеді:

$$C_n = (\text{ФОТ} - \text{ПО}) \times 11\%, \quad (5.21)$$

мұндағы, $ЗА$ - ФОТ-тың 10% зейнетақы құрайды және әлеуметтік салықпен салық салынбайды:

$$ЗА = \text{ФОТ} - 10\%, \quad (5.22)$$

$$ЗА = 4\,752\,000 - 475\,200 = 4\,276\,800 \text{ [тг]}.$$

5.21 формула бойынша әлеуметтік салық:

$$C_n = (4\,752\,000 - 4\,276\,800) \times 11\% = 52\,272 \text{ [тг]}.$$

Жүйені енгізгенге дейін, бір жылдың эксплуатациялық шығындар:

$$\Xi_1 = 4\,752\,000 + 52\,272 = 4\,804\,272 \text{ [тг]}.$$

Пайдалану шығыстары жүйені енгізгеннен кейін жылдық мынадай формула бойынша есептеледі:

$$\Xi_2 = A + \Xi + H, \quad (5.23)$$

мұндағы, А- амортизация;

Ξ - электр энергияға шығындар;

Н-үстеме шығындар.

Жылдық амортизация мына формула бойынша есептеледі:

$$A = K_{об} \times H_a, \quad (5.24)$$

мұндағы К- күрделі салымдар;

H_a- норма амортизация нормасы (25%).

$$A = 913\,697,93 \times 0,25 = 228\,424,4825 \text{ [тг]}.$$

Жылдық электр энергиясына арналған шығыстар мынадай формула бойынша есептеледі:

$$\Xi = W \times k_3 \times T \times C_{квт-ч}, \quad (5.25)$$

мұндағы, W- жабдық қуаты;

k₃- жүктеме коэффициенті (0.8);

T- бір жылғы жұмыс уақыты;

C_{квт-ч}- 1 кВт-сағаттағы электр энергия бағасы (3-деңгейлі=27,05 тг).

$$\Xi = 9 \times 0.8 \times 365 \times 24 \times 27,05 = 1\,706\,097 \text{ [тг]}.$$

Үстеме шығындар мынадай формула бойынша есептеледі:

$$H = Z_{oi} \times H_{pн} / 100\%, \quad (5.26)$$

$$H = 3\,960\,000 \times 40\% / 100\% = 1\,584\,000 \text{ [тг]}.$$

Жобаны енгізгеннен кейінгі пайдалану жылдық шығындар:

$$\Xi_2 = 228\,424,4825 + 1\,706\,097 + 1\,584\,000 = 3\,518\,521,4825 \text{ [тг]}.$$

5.3.2 формула бойынша шартты – жылдық экономика тең:

$$\Xi_{yt} = 4\,804\,272 - 3\,518\,521,4825 = 1\,285\,750,5175 \text{ [тг]}.$$

Күрделі салымдардың экономикалық тиімділігінің нормативтік коэффициенті мынадай формула бойынша анықталады:

$$E_n = \frac{1}{T_n}, \quad (5.27)$$

мұндағы, T_n – күрделі салымдардың өтелімділігінің нормативтік мерзімі, жылдар.

Күрделі салымдардың өтелімділігінің нормативтік мерзімі техникалық құралдардың моральдық ескіру мерзіміне және АЖ жобалық шешімдерге сүйене отырып қабылданады. ($T_n=1,2,3...n$), бағдарламалық өнімдер үшін өтелімділік мерзімі 4 жылға тең [17].

Күрделі салымдардың экономикалық тиімділігінің есептік коэффициенті:

$$E_p = \frac{\Delta_{уг}}{K}, \quad (5.28)$$

мұндағы, E_p – күрделі салымдардың экономикалық тиімділігінің есептік коэффициенті;

$\Delta_{уг}$ – күтілетін шартты-жылдық үнем, теңге;

K - жүйені құруға арналған күрделі салымдар, теңге.

Экономикалық тиімділіктің есептік коэффициенті мына формула бойынша есептеледі:

$$E_p = 1285\,750,5175 / 913\,697,93 = 1,4$$

Күрделі салымдардың өтелімділігінің есептік мерзімі:

$$T_p = \frac{1}{E_p}, \quad (5.29)$$

мұндағы, E – күрделі салымдардың экономикалық тиімділігінің коэффициенті.

$$T_p = 1 / 1,4 = 0,7 \text{ [жыл]}.$$

Сонда:

$$E_n = 1 / 0,7 = 1,4$$

Осылайша, жобаны енгізуден күтілетін жылдық экономикалық тиімділік:

$$\Delta_z = \Delta_{yz} - K \times E_n, \quad (5.30)$$

$$\Delta_z = 1285\,750,5175 - 913\,697,93 \times 1,4 = 6574 \text{ [тг]}.$$

Орындалған есептеулер нәтижелері кестеде келтірілген 5.9.

5.9 кесте - Жобаның салыстырмалы экономикалық тиімділік көрсеткіштері

Көрсеткіштердің атаулары	Значение
Шығындардың шартты жылдық экономиясы, тенге	1285 750,5175
Капиталдық салымдардың экономикалық тиімділік коэффициенті, (Ер)	1,4
Күрделі салымдардың өтелімділігінің есептік мерзімі, (Тр), жыл	0,7
Жылдық экономикалық тиімділік, тенге	6574
Капиталдық салымдар , тенге	913 697,93

5.5 Бөлім бойынша қорытынды техникалық-экономикалық негіздеме

Жобаны әзірлеуге және іске асыруға арналған шығындар бағдарламалық өнім мен жабдықтарды әзірлеуге арналған шығындарды қамтиды және **913 697,93** теңгені құрайды. Ағымдағы пайдалану шығыстарының төмендеуі нәтижесінде жобаны енгізуден шартты-жылдық үнем **1285 750,5175** теңгені, жылдық экономикалық тиімділік **6574** теңгені және өзін-өзі ақтау мерзімі 1,4 жылды құрады.

Қорытынды

Arduino бойынша дипломдық жұмыс - біздің идеяларымызды іске асыруға арналған құрал ретінде электроника, информатика және Arduino платформасын практикалық қолдану әлеміне кіріспе.

Жұмыс Arduino теориялық сипаттамасын қамтиды және практикалық маңызы бар нақты жобаларда пәнді оқытуды ұсынады. Бұл жұмыста жобаларда пайдалану үшін жарамды аяқталған шешімдерді білдіретін Arduino қолдануының бірнеше үлгілері мен жобалары бар. Қазіргі уақытта жоба пысықтауға дайын.

Дамудың одан әрі бағыттары:

- жарық датчигінің интеграциясы;
- сағат интеграциясы;
- сыртқы түрін эстетикалық жақсарту.

Практикалық жұмыстарды жүргізу барысында Arduino платформасына арналған бағдарламалар жазылды. Бірінші жұмыста SMT (Surface-mount technology) құрылған жарықдиодты жыпылықтауға қабілетті Arduino платформасына арналған бағдарлама жазылған. Екінші жұмыста схема жиналып, Морзе әліппесі бойынша кодталған сөз жасай отырып, жарықдиодты жыпылықтауға қабілетті Arduino платформасына арналған бағдарлама жазылды. Өріпті кодтау басқа әріппен белгілі бір уақыт аралығымен бөлінген тиісті белгілермен орындалды. Үшінші жұмыста Arduino платформасына арналған бағдарлама жазылған, онда әр жарық диодтың жарығының біртіндеп өсуімен белгілі бір уақыт аралығымен түрлі түсті үш жарықдиодты жыпылықтатады. Төртінші жұмыста фоторезистордың көмегімен жарықтандыруды өлшеу орындалды. Үй-жайдың жарықтандырылуына байланысты жарықдиодты қосу бағдарламасы жасалды. Бесінші жұмыста пьезокерамикалық сәуле шығарғыш зерттелді. Пьезокерамикалық сәуле шығарғыш - бұл кері пьезоэлектрлік әсердің арқасында дыбысты шығаруға қабілетті акустикалық құрылғы. Жарықтандыруды бағалау үшін аспаптың сұлбасы құрастырылды, Arduino контроллеріне арналған бағдарлама құрастырылды, онда аспап жарықтандыруды өзгерту кезінде тиісті жарықдиодты қосып, қысқа әуен ойнатылды.

Дипломдық жұмыс аясында негізгі кезеңдер орындалды:

- 1) Arduino платформасының мүмкіндіктері;
- 2) микроконтроллерлер базасында қарапайым құрылғыларды жобалаудың негізгі шешімдері;
- 3) бағдарламалардың тестілік үлгілері жазылған.

Одан әрі дипломдық жұмыста сипатталған мысалдарды жетілдіру, сондай-ақ Arduino платформаларын бағдарламалау үшін жаңа функционалдық және күрделі мысалдар жазу жоспарлануда.

Әдебиеттер тізімі

1. Белов А. В. Микроконтроллеры AVR в радиолубительской практике. — СПб.: Наука и Техника, 2007. - 339 с.
2. Белов А. В. Конструирование устройств на микроконтроллерах. - СПб.: Наука и Техника, 2005. - 256 с.
3. Бекешева А.И. Методические указания по экономической части выпускной работы. – Алматы: АУЭС, 2018. – 32 с.
4. Гололобов В.Н. С чего начинаются роботы? - М.: Солон-Пресс, 2011. - 189 с.
5. Голубцов М. С. Микроконтроллеры AVR от простого к сложному. - М.: Солон-Пресс, 2003. - 288 с.
6. ГОСТ 12.1.0179-2009 Система стандартов безопасности труда. Электробезопасность. Общие требования и номенклатура видов защиты. - Астана: МИИТ РК, 2009. – 36 с.
7. Иго Т. Arduino, датчики и сети для связи устройств: Пер. с англ. – 2 изд. – СПб.: БХВ-Петербург, 2015. - 544 с.
8. Национальный обзор по охране труда в республике Казахстан <http://base.safework.ru/safework?print&nd=444400037&spack=110listid=010000000100&listpos=3&lsz=9&nd=444400034&nh=0&> (16.04.20)
9. Охрана труда на предприятиях связи и охрана окружающей среды / И.Н. Баклашов. - М.: Радио и связь, 2009. – 247 с.
10. Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things. – СПб.: БХВ-Петербург, 2016. – 320 с.
11. Петин В.А. Микрокомпьютеры Raspberry Pi. Практическое руководство. – СПб.: БХВ-Петербург, 2015. - 240 с.
12. Петин В.А. Проекты с использованием контроллера Arduino. – СПб.: БХВ-Петербург, 2015. - 464 с.
13. Предпринимательство / под ред. В.Я.Горфинкеля. Рус. – М.: «Юнити», 2010. – 267 с.
14. СНиП РК 2.04-05-2002. Естественное и искусственное освещение. Общие требования. – М.: Стройиздат, 2002.
15. Соммер У. Программирование микроконтроллерных плат Arduino/Freeduino. – СПб.: БХВ-Петербург, 2012. – 256 с.
16. Трамперт В. Измерение, управление и регулирование с помощью AVR-микроконтроллеров. - К.: «МК-Пресс», - 2006. - 208 с.
17. Эванс Б. Arduino блокнот программиста. — London: CUP, 2007. - 240 с.

А қосымшасы

Бағдарлама коды

«Main activity» қосымша жасалатын барлық функциялардың тікелей орындалуына жауап береді [9].

```
package net.sytes.text_speech_bll;
import java.io.DataInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Locale;
import java.util.Timer;
import java.util.TimerTask;
import java.util.UUID;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.speech.tts.UtteranceProgressListener;
import android.util.Log;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends Activity implements
TextToSpeech.OnInitListener { // Синтезатор инициализациясы үшін
TextToSpeech.OnInitListener интерфейсін қолданумен класс ашу;
    public int upbluetooth=1;
    public int r19=10;
    public static final int CODE = 1234;
    public TextToSpeech tts;
    private TextView txtText;
    public TextView temtext;
    public int value=100;
    private int bluecancel=0;
    public int bytes;
    public byte[] buffer = new byte[128];
    public String readMessage;
    public Timer time;
```

```
//Socket, көмегімен Arduino bluetooth-ге деректер жіберіледі
private BluetoothSocket clientSocket;
// тізбекті bluetooth құрылғыға UUID қосылу үшін
private UUID uuid = UUID.fromString("00001101-0000-1000-8000-
00805F9B34FB");
public String spout;
public String sp;
// сұрақтар жолдарының мысалдары
public String sp0 = "қызылды өшіру";
public String sp1 = "қызылды қосу";
public String sp2 = "жасылды өшіру";
public String sp3 = "жасылды қосу";
public String sp4 = "көкті өшіру";
public String sp5 = "көкті қосу";
public String sp6 = "температурасы";
public String sp10 = "сенің атын кім?";
public String sp11 = "сен не жасай аласың";
public String sp12 = "сенің командаларын";
public String sp13 = "рахмет";
public String sp14 = "сен қайдансың? ";
public String sp15 = "салем ";
public String sp19 = "байланыс бітті";
// жауаптар жолдарының мысалдары
public String sp20 = "менің атым Күн";
public String sp21 = " мен жарықты қосып, өшіре аламын және
температураны анықтай аламын ";
public String sp22 = " Жасыл қосу. Жасыл өшіру. Қызыл және көк үшін
бірдей. Температурасы.";
public String sp23 = " өтінеміз ";
public String sp24 = " мен Хмельницкий ";
public String sp25 = "салеметсіз";
public String sp29 = "сау болыңыз";
public String sp30 = "қызылды өшірдім";
public String sp31 = "қызылды қостым";
public String sp32 = "жасылды өшірдім";
public String sp33 = "жасылды қостым";
public String sp34 = "көкті өшірдім";
public String sp35 = "көкті қостым";
// Сақтау үшін хеш картаны сипаттау
// кілтті (біздің жағдайда
TextToSpeech.Engine.KEY_PARAM_UTTERANCE_ID)
```

Бағдарлама кодының жалғасы

```
// қандай да болса мәні - жолдар (мысалы - "5678")
private HashMap<String, String> params = new HashMap<>();
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    txtText = (TextView) findViewById(R.id.input_text);
    temtext = (TextView) findViewById(R.id.textView2);
    tts = new TextToSpeech(this, this);
    tts.setOnUtteranceProgressListener(new utteranceProgressListener());
    // Тыңдаушыны сөйлеу синтезін орнату
    bluet(); // Bluetooth қосу
}
public void spi() { // Сөйлеуді тану белсенділігін шақыру
    Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, " Дауыстап және
анық айт!");
    startActivityForResult(intent, CODE);
}
@Override
public void onDestroy() {
    super.onDestroy();
    // Сөйлеу синтезаторын тоқтату tts!
    if (tts != null) {
        tts.stop();
        tts.shutdown();
    }
    try
    clientSocket.close(); // Bluetooth –бен қосылысты жабу
    } catch (IOException e2) { }
}
@Override
public void onInit(int status) { // Сөйлеу синтезінің алдында
инициализациялау
    if (status == TextToSpeech.SUCCESS) {
        int result = tts.setLanguage(Locale.getDefault());
        params.put(TextToSpeech.Engine.KEY_PARAM_UTTERANCE_ID, "
HELLO");
        if (result == TextToSpeech.LANG_MISSING_DATA
        || result == TextToSpeech.LANG_NOT_SUPPORTED) {
```

```
Log.e("TTS", "This Language is not supported");
}
} else {
Log.e("TTS", "Init Failed!");
}
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{ // Кейін шақырылады
    // дауысты тану нәтижесін алу
    super.onActivityResult(requestCode, resultCode, data);
    ArrayList<String> spee;
    value=100;
    switch (requestCode){
    case CODE: {
        if (resultCode == RESULT_OK && null != data) {
            // текстік жолдардың тізімін аламыз – тану нәтижесі.
            // Жолдар бірнеше болуы мүмкін, өйткені әрдайым сөйлеуді дәл тануға
            болмайды.
            // Дұрыс нәтижелер тізімнің басында жүреді
            spee = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
            sp = spee.get(0).toString();
            txtText.setText(sp);
            spout = sp;
            int r0 = sp.compareTo(sp0);
            int r1 = sp.compareTo(sp1);
            int r2 = sp.compareTo(sp2);
            int r3 = sp.compareTo(sp3);
            int r4 = sp.compareTo(sp4);
            int r5 = sp.compareTo(sp5);
            int r6 = sp.compareTo(sp6);
            int r10 = sp.compareTo(sp10);
            int r11 = sp.compareTo(sp11);
            int r12 = sp.compareTo(sp12);
            int r13 = sp.compareTo(sp13);
            int r14 = sp.compareTo(sp14);
            int r15 = sp.compareTo(sp15);
            int r19 = sp.compareTo(sp19);
            if (r0 == 0) spout = sp30;
            if (r1 == 0) spout = sp31;
            if (r2 == 0) spout = sp32;
            if (r3 == 0) spout = sp33;
            if (r4 == 0) spout = sp34;
```

```
if (r5 == 0) spout = sp35;
if (r6 == 0) spout = sp6;
if (r10 == 0) spout = sp20;
if (r11 == 0) spout = sp21;
if (r12 == 0) spout = sp22;
if (r13 == 0) spout = sp23;
if (r14 == 0) spout = sp24;
if (r15 == 0) spout = sp25;
if (r19 == 0) spout = sp29;
if (upbluetooth == 0) { // Bluetooth байланысы бар болса, онда
// салыстыру нәтижесі таңбамен ұсынылады
// Мысалы, егер дауысты тану нәтижесі жолға сәйкес келсе
// "жасыл қосу" bluetooth арқылы 3 таңбасын жібереміз (код 51)
if (r1 == 0) value = 49;
if (r0 == 0) value = 48;
if (r3 == 0) value = 51;
if (r2 == 0) value = 50;
if (r5 == 0) value = 53;
if (r4 == 0) value = 52;
if (r6 == 0) value = 54;
// деректерді жібереміз
if (value != 54) outData(value);
if (value == 54) {
temtext.setText(readMessage); // Температураны басып шығарамыз
spout = readMessage;
}
}
tts.speak(spout, TextToSpeech.QUEUE_ADD, params); // Сөйлеуді
синтездеу
}
}
break;
}
}

public void click(View view) {
if(upbluetooth==0) { // Если Bluetooth подключен
// Батырманы басқан кезде температураны беру және оқу бағдарламасы
жұмысының таймерін тоқтатамын
time.cancel(); time.purge();
// Жаңа бастапқы бағдарлама таймерін іске қосамын
```

```
// деректермен (температура сұранымын жіберу және температураны
кабылдау)
bluecancel = 0;
time = new Timer();
bluetoothInOut bInOut = new bluetoothInOut();
time.schedule(bInOut,500,1000);
}
spi(); // Мен сөйлеуді тануды бастаймын
}
public void bluet() { // Bluetooth қосылу
Intent enableBt = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
enableBt.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
startActivity(enableBt);
// әдетте Bluetooth қолданылады
BluetoothAdapter bluetooth = BluetoothAdapter.getDefaultAdapter();
try {
// Қарапайымдылық үшін нақты мекен-жайы бар bluetooth таңдаңыз
BluetoothDevice device =
bluetooth.getRemoteDevice("98:D3:31:B0:86:16");
// Құпия socket үшін RFCOMM жасау - кіріс және шығыс хабарлар үшін
clientSocket = device.createRfcommSocketToServiceRecord(uuid);
// Қашықтағы bluetooth қосылу әрекеті
clientSocket.connect();
// Егер әрекет орындалса, экранның төменгі жағында хабар шығады
Toast.makeText(getApplicationContext(), "bluetooth байланыс
орнатылды", Toast.LENGTH_LONG).show();
upbluetooth=0;
time = new Timer();
bluetoothInOut bInOut = new bluetoothInOut();
time.schedule(bInOut,500,1000); // Бағдарлама іске қосылғаннан кейін
500 миллисекундтан кейін бастау
// әр секунд сайын температураны сұрау
}
// Қате пайда болған жағдайда, bluetooth қосылмағанын хабарлаймыз
catch (IOException e) {
upbluetooth=1;
Toast.makeText(getApplicationContext(), "Bluetooth тексер!",
Toast.LENGTH_LONG).show();
}
}
```

```
// Сөйлеу синтезінің аяқталуын бекіту үшін қажетті класс жасаңыз
public class utteranceProgressListener extends UtteranceProgressListener {
    @Override
    public void onDone(String utteranceId) { // Синтезатормен сөйлеуді
аяқтағаннан кейінгі әрекеттер
        r19 = sp.compareTo(sp19);
        if(r19 != 0) spi(); // Егер де "байланыс бітті" десе, онда сөйлеуді
активизациялау қайтадан бастайды
        else{
            if(upbluetooth==0) { // Егер Bluetooth қосылған болса
                // Егер "байланыс соңы" айтылса-таймерді тоқтатамын
                time.cancel(); time.purge();
                // жаңа бастапқы деректермен бағдарламаның таймерін іске қосамын
                // (температураны сұратпай, бірақ температураны қабылдаумен)
                bluecancel = 1;
                time = new Timer();
                bluetoothInOut bInOut = new bluetoothInOut();
                time.schedule(bInOut,500,1000);
            }
        }

        Log.d("TtsUtteranceListener", "utterance Done: " + utteranceId);
    }
    @Override
    public void onStart(String utteranceId) {
        Log.d("TtsUtteranceListener", "utterance Start: " + utteranceId);

        @Override
        public void onError(String utteranceId) {
            txtText.setText("Error");
            Log.d("TtsUtteranceListener", "utterance Error: " + utteranceId);
        }
    }
    public void outData(int value) {
        try {
            // Деректерді жіберу үшін шыққан ағынын аламыз
            OutputStream
outStream = clientSocket.getOutputStream();
            // деректерді жібереміз
            outStream.write(value);
        } catch (IOException e) {
        }
    }
}
```

```
    }  
    public class bluetoothInOut extends TimerTask {  
    public void run() {  
    try {  
    if( bluecancel == 0 )  
    // Тек осы жағдайда ғана температураға сұранысты жібереміз {  
    OutputStream outputStream = clientSocket.getOutputStream();  
    outputStream.write(54);}  
    // Деректерді қабылдау үшін кіріс ағынын аламыз  
    InputStream inb = clientSocket.getInputStream();  
    // Bluetooth – ден строкаға қарай кіріс ағынның түрлендіреміз  
    DataInputStream in = new DataInputStream(inb);  
    bytes = in.read(buffer);  
    if (bytes > 10 ) // bluetooth арқылы қабылданды (мысалы) 10 байттан  
    жоғары, онда  
    {  
    // байттарды нөлдік индекстен bytes индексіне дейінгі жолға  
    түрлендіреміз  
    readMessage = new String(buffer, 0, bytes);  
    }  
    } catch (IOException e) {  
  
    }  
    }  
    }  
    }
```

Қысқаша айтқанда, қолданба орнатылған Google тану жүйелері арқылы сөйлейді, бұрын жазылған нұсқаларға сәйкес іздейді және егер сәйкестік тапса, 0-ден 6-ға дейінгі мәндерді жібереді [10]. Болашақ қосымшаның түрі.

Белгі қою файлының коды, болашақ қосымшаның түрін анықтайды:

```
<RelativeLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
android:paddingLeft="@dimen/activity_horizontal_margin"  
android:paddingRight="@dimen/activity_horizontal_margin"  
android:paddingTop="@dimen/activity_vertical_margin"  
android:paddingBottom="@dimen/activity_vertical_margin"  
tools:context=".MainActivity"
```



```
android:background="#ff074bff">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="68dp"
    android:hint="Распознанный текст"
    android:id="@+id/input_text"
    android:textSize="45dp"
    android:gravity="center"
    android:textColor="#ff3c09ff"
    android:background="#ff05ff14"
    android:textColorHint="#ff2f06ff"
    android:width="600dp"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true" />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Нажми для начала диалога"
    android:id="@+id/button"
    android:textSize="65dp"
    android:onClick="click"

    android:textColor="#ffffff"
    android:textStyle="bold"
    android:background="#ffff04bb"
    android:layout_marginBottom="187dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView2"
    android:hint="Температура"
    android:textSize="45dp"
    android:gravity="center"
    android:textColorHint="#ff1004ff"
    android:background="#fffcfffa"
    android:textColor="#ff0f07ff"
    android:width="600dp"
    android:layout_marginTop="270dp"
```

```
android:layout_below="@+id/input_text"
android:layout_centerHorizontal="true" />
</RelativeLayout>
```

Манифест файлының коды

Манифест файлында Android-ортасында жұмыс істеуге арналған қосымшаның барлық құқықтары мен рұқсаттары жазылады:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="net.sytes.text_speech_bll" >
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission
android:name="android.permission.BLUETOOTH_ADMIN" />
<application
android:allowBackup="true"
android:icon="@drawable/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme" >
<activity
android:name=".MainActivity"
android:label="@string/app_name" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
</manifest>
```

Манифест файлында Bluetooth құрылғысына рұқсат беру керек:

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission
android:name="android.permission.BLUETOOTH_ADMIN"
```

Ардуиномен жұмыс істеуге арналған скетч

Ардуиноға арналған бағдарлама «скетч» деп аталады [11]. Ардуиноның стандартты ортасында жазамыз:

```
#include <OneWire.h> //Мына жолда библиотеканы косып жатыр
OneWire ds(10); //мынау 10 аяғына микроконтроллердің датчикті қосатын
портты инициализациялау
#define BAUDRATE 9600 //блютубен жіберетін жылдамдықтың өлшем
бірлігі бод/с
```

Бағдарлама кодының жалғасы

```
boolean started = false; //логикалық айнымалы белгілі бір команданын  
жағдайын көрсетеді  
//бастапқы функция - бұл функцияда керек командалар инициализацияланады  
void setup(void) {  
  Serial.begin(BAUDRATE); //жылдамдықты бекіту  
  Serial.println("*****"); //мынау блютузбен  
бастапқы ақпаратты жіберу  
  Serial.println("  Diploma project");  
  Serial.println("  Send Temperature data from Bluetooth");  
  Serial.println("*****");  
  delay(1000); //мынау 1 с күту  
}  
//мынау мәңгі функция, микроконтроллер әрқашан осы функция ішінде  
болады  
void loop(void) {  
  byte i; //айнымалылар  
  byte present = 0;  
  byte type_s;  
  byte data[12]; //массив айнымалы  
  byte addr[8];  
  float celsius, fahrenheit; //бөлшекті айнымалы  
  if ( !ds.search(addr)) { //мына шарт датчик қосулыма соны тексереді  
    ds.reset_search();  
    delay(250);  
    return;  
  }  
  if (started == false) //мына шарт датчиктің адресін жазады  
  { present = ds.reset();  
    ds.select(addr);  
    ds.write(0x4E);  
    ds.write(0x4B);  
    ds.write(0x46);  
    ds.write(0x6f);  
    started = true;  
  }  
  ds.reset();  
  ds.select(addr);  
  ds.write(0x44, 1); //start conversion, with parasite power on at the end  
  delay(1000); // maybe 750ms is enough, maybe not  
  present = ds.reset();  
  ds.select(addr);
```

```
ds.write(0xBE); //Read Scratchpad
for ( i = 0; i < 9; i++) { //мұнда датчикті оқиды
data[i] = ds.read();
}
int16_t raw = (data[1] << 8) | data[0];
if (type_s) {
raw = raw << 3; //9 bit resolution default
if (data[7] == 0x10) {
raw = (raw & 0xFFF0) + 12 - data[6];
}
} else {
byte cfg = (data[4] & 0x60);
if (cfg == 0x00) raw = raw & ~7; // 9 bit resolution, 93.75 ms
else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 187.5 ms
else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms
}
celsius = (float)raw / 16.0; // сол қойылған мәндерді түрлендіреді целсийға,
фаренгейтке
fahrenheit = celsius * 1.8 + 32.0;
Serial.print(" Temperature = "); //мұнда соларды жібереді блютузбен
Serial.print(celsius);
Serial.print(" C, ");
Serial.print(fahrenheit);
Serial.println(" F");
}
```