

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ



Ә. Бүркітбаев атындағы Өнеркәсіптік автоматтандыру және цифрлау институты

«Роботтық техника және автоматиканың техникалық құралдары» кафедрасы

Досаев Аман Сердалыұлы

«Күн шағын электр станциясын жобалау»

ДИПЛОМДЫҚ ЖҰМЫС

5B071600 – Аспап жасау мамандығы

Алматы 2020



Ә. Бүркітбаев атындағы Өнеркәсіптік автоматтандыру және цифрлау институты

«Роботтық техника және автоматиканың техникалық құралдары» кафедрасы

ҚОРҒАУҒА РҰҚСАТ

РТжАТҚ кафедра меңгерушісі

техника ғылым кандидаты



Қ.А. Ожикенов

«23» мамыр 2020 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Күн шағын электр станциясын жобалау»

5B071600 – Аспап жасау мамандығы бойынша

Орындады

Досаев Аман

Ғылыми жетекшісі
Физ.-мат. ғыл. канд.,
қауымдастырылған профессор

Алдияров Н.У.

« 23 » мамыр 2020 г.

Алматы 2020



Ә. Бүркітбаев атындағы Өнеркәсіптік автоматтандыру және цифрлау институты

«Роботтық техника және автоматиканың техникалық құралдары» кафедрасы

5B071600 – Аспап жасау

БЕКІТЕМІН

РТжАТҚ кафедра меңгерушісі
техника ғылым кандидаты



Қ.А. Ожикенов
«23 » қантар 2020 ж.

ТАПСЫРМА

дипломдық жұмысты орындауға

Білім алушыға Досаев Аман Сердалыұлы

Тақырыбы: Күн шағын электр станциясын жобалау

Университет ректорының бұйрығымен бекітілген №726-б «27» қантар 2020 г.

Аяқталған жұмысты тапсыру мерзімі «15» мамыр 2020 ж.

Дипломдық жұмыстың бастапқы мәліметтері: екі осьті Күн трекерін жасау болып табылады. SPA алгоритмін жүзеге асыратын басқару моделін және бағдарламасын жазу.

Дипломдық жұмыста әзірленуге жататын мәселелер тізімі:

- а) Көкжиекте Күн координатасын анықтайтын формула алу;
- б) Proteus бағдарламасында жүйенің принципіалды схемасын ;
- г) AVR Studio4 бағдарламасында жүйенің жұмыс кодын жазу;

Графикалық материалдың тізбегі (міндетті сызбаларды дәл көрсете отырып):

10 слайд

Ұсынылатын негізгі әдебиеттер: 20 әдебиеттер тізімі

Дипломдық жобаны дайындау

КЕСТЕСІ

Бөлімдер атауы, әзірленетін сұрақтар тізбесі	Ғылыми жетекшіге ұсыну мерзімдері	Ескертпелер
Теориялық бөлім	22.01 – 15.02.2020 ж.	Орындалды
Есептеу бөлімі	22.01 – 15.02.2020 ж.	Орындалды
Бағдарламалық бөлім	15.03 – 20.04.2020 ж.	Орындалды
Зерттеу бөлімі	15.03 – 20.04.2020 ж.	Орындалды
Қорытынды бөлім	15.03 – 20.04.2020 ж.	Орындалды

Аяқталған дипломдық жобаға және оған қытысты бөлімдерінің кеңесшілері мен қалып бақылаушының

ҚОЛТАҢБАЛАРЫ

Бөлімдердің атауы	Ғылыми жетекшілер, кеңесшілер, (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қол қойылған күні	Қол
Қалып бақылаушы	Ж.С.Бигалиева, техника ғылымдары магистрі, лектор	23.05.2020 ж.	

Ғылыми жетекшісі



Алдияров Н.У.

Тапсырманы орындауға алған білім алушы



Досаев А.С

Күні «23» мамыр 2020 ж.

АҢДАТПА

Дипломдық жобада фиксацияланған күн панелінің төмен ПӘК көрсеткішін арттыру мақсатында, Күн трекер жүйесі ұсынылады. Күн трекерлерінің әр параметрі бойынша жіктелуі және тиімді қолдану аймақтары келтірілген. Аспан механикасын зерттеу барысында Жердің кез келген нүктесінен, шын уақытта Күн координатасын анықтайтын формулалар алынды. Proteus бағдарламасында дайын модульдар бойынша Күн трекерінің моделі құрастырылды. SPA алгоритмінің логикасы негізінде ассемблер тілінде жобаның жұмыс коды жазылды.

АННОТАЦИЯ

В дипломном проекте предлагается система солнечного трекера с целью повышения показателя низкой КПД фиксированной солнечной панели. По каждому параметру солнечного трекера приведены классификация и область эффективного использования. При исследовании небесной механики были получены формулы, определяющие в настоящее время координаты Солнца из любой точки Земли. В программе Proteus была разработана модель солнечного трекера по готовым модулям. На основе логики алгоритма SPA записан рабочий код проекта на языке ассемблер.

ABSTRACT

The diploma project proposes a solar tracker system to improve the low efficiency of a fixed solar panel. For each parameter of the solar tracker, the classification and zones of effective use are shown. In the study of celestial mechanics, formulas were obtained that currently determine the coordinates of the Sun from any point on Earth. The Proteus program developed a model of a solar tracker based on ready-made modules. Based on the logic of the SPA algorithm, the working code of the project is written in Assembly language.

МАЗМҮНҮ

Кіріспе	9
1 Күн бақылау жүйесінің жіктелуі және негізгі компоненттері	10
1.1 Бір осьтік трекерлер	10
1.1.1 Горизонталды бір осьті трекер (HSAT)	10
1.1.2 Көлбеу модульдері бар горизонтальді бір осьті трекерлер (HTSAT)	11
1.1.3 Вертикальды бір осьті трекерлер (VSAT)	12
1.1.4 Көлбеу бір осьті трекерлер (TSAT)	13
1.1.5 Бір осьті полярлықты түзету трекерлері (PSAT)	13
1.2 Екі осьтік трекер	13
1.2.1 Екі осьтік трекер Tip-Tilt (TTDAT)	14
1.2.2 Азимут-биіктік екі ось трекері	15
1.3 Күн трекерінде қолданылатын жетек.	16
1.3.1 Электр жетегі	16
1.3.2 Гидрожетек	16
1.3.3 Пассивті жетегі	17
1.3.4 Бақылау бойынша нұсқаушы	17
1.4 Күн трекерінің басқару тәсілі бойынша жіктелуі	17
2 Аспан механикасы бойынша күннің азимутын және зенитін есептеу	19
2.1 Аспан сферасының негізгі сызықтары	19
2.2 Координаталық жүйелер	20
2.3 Алматы қаласы үшін күн шырағының координатасын есептеу	24
3 Күн жүйе макетінің схемасы	25
3.1 Трекердің функционалдық схемасы	25
3.2 Трекердің принципіалды схемасы	26
3.3 Трекердің жұмыс алгоритмі	30
3.4 Ассемблер тілінде бағдарламаға шолу	31
Қорытынды	
Пайдаланылған әдебиеттер тізімі	

Әлемдік экономикада электр энергиясына деген сұраныс артуын және көміртек қорларының шектеулігіне байланысты проблемалық ушығуы, балама энергияға деген қызығушылықтың артуына алып келді. Жаңартылатын энергия көздерін пайдалану энергетикада пайдалы қазбаларды өндіруге және пайдалануға байланысты қауіптерді азайтуға, сондай-ақ дәстүрлі энергетиканың қоршаған орта мен климатқа зиянды әсерін азайтуға мүмкіндік береді. Жаңартылатын энергия көздері көмірсутектер қорларын болашақ ұрпаққа сақтауды және оларды басқа да энергетикалық емес мақсаттар үшін пайдалану мүмкіндігін қамтамасыз етеді. Күн энергиясы жаңартылатын энергия көздеріне жатады. Жер бетіне түсетін күн сәулесінің қуаты адамзаттың энергияға деген ағымдағы қажеттілігінен бірнеше есе асып түсетіні белгілі. Жаңартылатын энергия көздерін дамытуға негізгі кедергілер жоғары үлестік күрделі шығындар және соның салдарынан электр энергиясына жоғары тарифтер қойылуы болып табылады. Жаңартылатын энергия көздері, әсіресе Қазақстанның шалғайдағы және энергияға тапшы аймақтарында экономикалық тұрғыдан ақталуы мүмкін. "Жасыл экономика" концепциясы қазіргі заманғы әлемде үлкен мәнге ие.

Өкінішке орай, күн энергетикасы онда қолданылатын күн элементтерінің пайдалы әсерінің жоғары коэффициентімен ерекшеленбейді (орташа алғанда, пәк мәні 14-18% - ды құрайды), бұл күн инсоляциясы жыл мезгілінің өзгеруімен қатты түрленетін аймақтарда оның қолданылуының әлеуетін шектейді. Бұл фотоэлектрлік жүйелердің әзірлеушілерін жаңа іздегістерге және генерацияланатын энергияның тиімділігін, өнімділігін және өзіндік құнын төмендетуді қамтамасыз ететін белгілі техникалық шешімдерді жетілдіруге мәжбүр етеді. Күн батареясының негізгі элементі болып табылатын фотоэлектрлік түрлендіргіштің энергия өндіруінің тиімділігіне географиялық және климаттық жағдайлар, сондай-ақ ФЭА бетіне сәулелердің құлау бұрышы әсер етеді. Соңғы аспект күн батареяларының бетін күн сәулесінің ішінде автоматты түрде бағыттау құрылғыларының арқасында реттелуі мүмкін. Бұл құрылғы Күн трекері деген атау алды. Күн трекерлерде орнатылған панельдер белгіленген панелдерден көбірек Күн радиациясын алады. Бұл таңғы және кешкі сағаттарда күн сәулесінің құлау бағыты бекітілген панель жазықтығына параллель болуынан туады.

Дипломдық жобаның мақсаты-күн сәулесін бақылайтын трекердің алгоритмін және оның шағын екі осьті прототипін құру. Осы мақсатқа жету үшін келесідей тапсырмалар орындалуы қажет

- Аспан механикасын зерттеп кез келген уақытта Күннің Көкжиектегі биіктігін және Азимутын анықтайтын формула құру

- Күн трекер прототипінің функционалды және принципіалды схемасын құру. Proteous бағдарламасында модельдеу. Шағын макетін құру

- AVR STUDIO 4 бағдарламасына Ассемблер тілінде жұмыс кодын жазу

1 Күн бақылау жүйесінің жіктелуі және негізгі компоненттері

Қазіргі заманғы күн қадағалаушы жүйелер өте алуан түрлі және баға жағынан, жобалау және басқару принциптері бойынша айтарлықтай өзгеруі мүмкін. КБЖ-ға қойылатын негізгі талаптар IEC 62817–2014 халықаралық стандартқа сәйкес анықталған, онда күннің орналасуын бақылау жүйелерінің негізгі сипаттамалары, олардың жіктелу ерекшеліктері және сынақ әдістері көрсетілген. Айналу осьтерінің саны мен бағыты бойынша күн трекерлері бір осьтік (single axis trackers - SAT) және екі осьті (dual axis tracker - DAT) болып жіктеледі. Екі осьтік жүйелермен салыстырғанда, бір осьті күн трекерлері қарапайым дизайнға ие және сәйкесінше арзанырақ, сондықтан олар ең көп сұранысқа ие. Трекер түрін таңдау көптеген факторларға байланысты, оның ішінде монтаж өлшемі, электр параметрлері, жер шектеулері, ендік және жергілікті ауа-райы жағдайларына тәуелді.

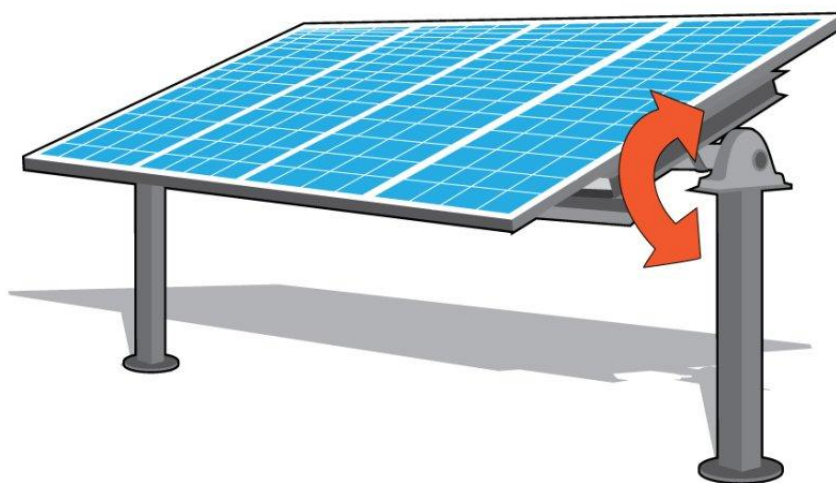
1.1 Бір осьтік трекерлер

Бір осьтік трекерлер, аты айтып тұрғандай, күнді қадағалай алатын бір ғана оське ие. Әдетте олар солтүстік меридианның бойында орналасады, бірақ кейбір трекерлерде қолданылатын жетілдірілген алгоритмдер оларды кез-келген координаталық бағытта орналастыруға мүмкіндік береді. Бір осьті Күнді бақылау жүйелерін монокоординаты деп те атауға да болады. Бір осьті трекерлердің өздері бірнеше топқа жіктеледі. Олардың қатарына горизонтальді бір осьті трекерлер (HSAT), көлбеу модульдері бар горизонтальді бір осьті трекерлер (HTSAT), вертикальді бір осьті трекерлер (VSAT), көлбеу бір осьті трекерлер жатады (TSAT) және бір осьті полярлықты түзету трекерлері (PSAT). Модульдің трекердің осіне қатысты бағыты жүйенің тиімділігін модельдеу кезінде маңызды. Құрылымдардың әрқайсысының өзіндік артықшылықтары мен кемшіліктері, сондай-ақ негізгі қолдану жағдайлары бар.

1.1.1 Горизонталды бір осьті трекер (HSAT)

Аты айтып тұрғандай, бұл трекерлердің бақылау осі жерге қатысты көлденең. HSAT әдетте үлкен генерацияланатын жүйелерде қолданылады. Бұл трекерлер төмен енді аймақтарда тиімдірек келеді. Көлденең трекерлерде әдетте панель беті айналу осіне параллель болады. Бұл типтегі трекерлер солтүстік-оңтүстік сызығына бағытталған және жер үсті дүрбісінің бетінде бір немесе бірнеше панельдер жалғануы мүмкін. Құрылғының өзіндік құнын төмендету үшін, бір осьті горизонталь трекерлерінің айналу осінің екі шетіндегі тіректер көптеген трекерлерге бөліп жалғау қажет. Панельдер горизонтальді жалғанғандықтан, ось құбырында өз-өзін көленкелеу қауіпінсіз ықшамды орналасуы мүмкін және тазалық үшін қол жетімді. Көлденең бір осьті

трекерлердің ұзын көлденең дүрбісі тірекпен немесе жақтаумен мойынтірек арқылы ұсталып тұр. Панельдер дүрбінің бойында орналастырылған , және дүрбі өз осімен күні бойы) күннің көрінетін қозғалыстарын бақылау үшін айналатын болады. HSAT өзінде үлкен энергетикалық мүмкіндіктер мен басқа трекерлерге қарағанда өнім бағасының төмендігімен ,сонымен қатар орналастырғанда және қолданыста шығындардың төменділігімен ерекшеленеді. Одан басқа, күннің екінші жартысында жоғары көрсеткіштер,бұл өндірістің ең жоғары уақыт талабына сәйкес үлкен желілер үшін аса қажет. HSAT жүйелері өз өнімділігін көктем мен жаз уақыттарында ,күн аспанда жоғары болғанда көбейтеді .



1.1 сурет- Горизонталды бір осьті трекер (HSAT)

1.1.2 Көлбеу модульдері бар горизонтальді бір осьті трекерлер (HTSAT)

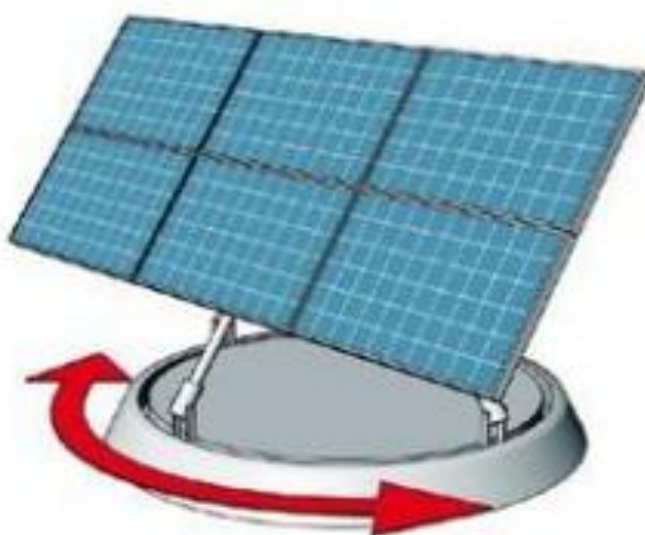
Бұл трекерлер HSAT трекерімен бірдей жұмыс принциптерін иеленеді, солтүстік-оңтүстік бағытындағы сызық бойында дүрбінің көлденең осін сақтай отырып , күні бойы күн модульдерін шығыстан батысқа қарай айналдырады. Тек айырмашылығы HSAT модулі көлденең бағытта 0 градус бұрышында монтаждалады,ал HTSAT модульдері белгілі бір еңкею градусында орналастырылады. Бұл трекерлер биік таулы аймақтарға көбірек келеді(қолайлы).HTSAT трекерлерінде, HSAT және VSAT трекерлерінің артықшылықтарын біріктіруге тырысады сонымен қатар олар азырақ орын алады. Осылайша ,ол көлденең трекерді VSAT артықшылықтарымен қамтып, күн жобасының жалпы құнын түсіріп, қолжетімдірек болып келеді.



1.2 сурет - Көлбеу модульдері бар горизонтальді бір осьті трекерлер (HTSAT)

1.1.3 Вертикальды бір осьті трекерлер (VSAT)

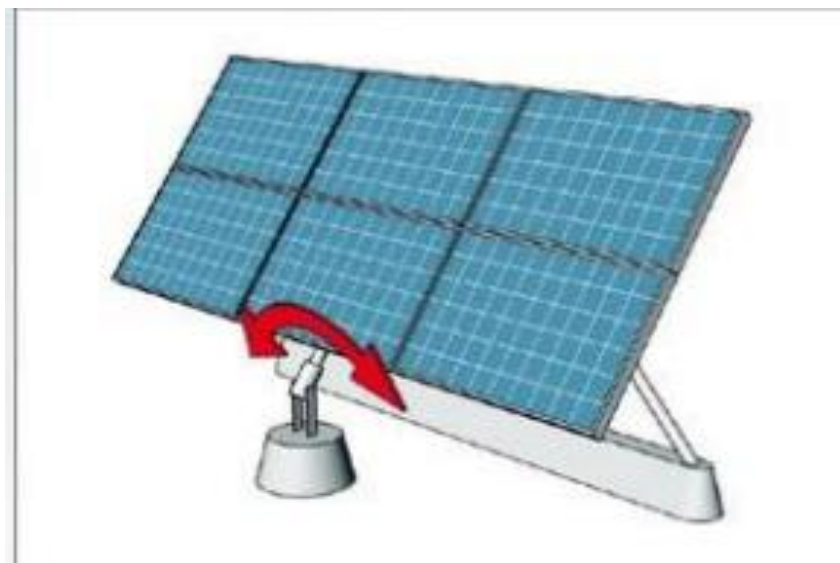
VSAT айналу осі жерге қатысты вертикалды орналасқан. Егер сіз күн ұзақтығы көп жаз мезгілімен күн жолы өте жоғары емес таулы аймақта болсаңыз, онда бұл трекер HSAT-қа қарағанда тиімдірек болады. VSAT жүйесімен жабдықталған панельдерді қатарға орнатқан кезде, қажет емес қуат жоғалуын болдырмау және жерді пайдалануды оңтайландыру үшін, панель айналу кезінде бір панельдің екіншісіне көлеңке түсірмейтіндей етіп орналасуды мұқият қарастырған жөн. Әдетте VSAT трекерлерінің жұмыс беті (фотоэлементтермен) айналу осіне қатысты бағытталған белгілі бір бұрышқа ие. Яғни, оның бекітілген көлбеуі бар.



1.3 сурет - Вертикальды бір осьті трекерлер (VSAT)

1.1.4 Көлбеу бір осьті трекерлер (TSAT)

Горизонтальді және вертикаль айналу осьтері бар Күн бақылау жүйелерінің ортасында орналасқан жүйе көлбеулі бір осьті Күн бақылау жүйесі ретінде саналады. Желдің кедергісін төмендету және бөктің биіктігін азайту үшін трекердің еңкею бұрышы жиі шектеледі. Модуль бақылау кезінде айналу осіне қатысты айналымы симметриялы болып келетін цилиндрді бұрады. Мұндай бақылау жүйелерін экваторлық бақылау жүйелері деп те атайды.



1.4 сурет - Көлбеу бір осьті трекерлер (TSAT)

1.1.5 Бір осьті полярлықты түзету трекерлері (PSAT)

Көлбеу униаксиалды трекердің ғылыми қызықты нұсқасы - PASAT полярлықты теңестіретін бір білікті трекерлер. Бір осьті көлбеу трекерді нақты орындау кезінде иілу бұрышы қондырғының еніне тең болады. Бұл трекердің айналу осін Жердің айналу осімен туралайды. Олар қатты жел профиліне байланысты сирек қолданылады.

1.2 Екі осьтік трекер

Күн энергиясын қолданудың максималды тиімділігі қос осьті трекер көмегімен қамтамасыз етіледі. Бір осьтік трекерлерден айырмашылығы, екі осьтік трекерлер үнемі күнге қарайды, себебі олар екі түрлі бағытта қозғала алады. Қос осьтік трекерлердің басты артықшылығы-олардың энергия өндірісін барынша арттыру қабілеттілігінде. Дегенмен, олардың механикалық күрделілігі неғұрлым жоғары дәрежеге ие, бұл оларды қымбатырақ етеді және белгіленген

қызмет мерзімі ішінде көп қызмет көрсетуді талап етеді. Жерге қатысты бекітілген осьті негізгі ось деп санауға болады. Негізгі оське байланысқан осьті екінші ось деп санауға болады. Екі осьтік трекерлердің таралған бірнеше іске асырылу жолдары бар. Олар негізгі осьтердің жерге қатысты бағдарлауы бойынша жіктеледі. Екі кең таралған түрлері- тірек бағанында айналу осі бар қос осьті трекерлер (tip-tilt dual axis tracker – TTDAT) және басты ось тік болып табылатын, тірек жазықтығы бар трекерлер (azimuth-altitude dual axis tracker – AADAT). Трекер осіне қатысты модульдің бағдары өнімділікті моделдеу кезінде маңызды. Әдетте DAT модульдері екінші айналу осіне параллель бағытта алған болады

1.2.1 Екі осьтік трекер Tip-Tilt (TTDAT)

Атауынан көрініп тұрғандай, көлбеу басы бар екі осьтік трекер (TTDA) тіреудің жоғарғы бөлігінде орнатылған панельдер массивін қамтиды. Әдетте шығыс-батыс қозғалысы, подшипик орнатылған полюстің жоғарғы жағындағы массивтарды қозғалысқа келтіру арқылы іске асады. Айналымды мойынтіректің жоғарғы бөлігінде панельдердің вертикалды айналуын қамтамасыз ететін Т - немесе Н-тәрізді механизм орнатылған және ол массив үшін бекітудің негізгі нүктелерін қамтамасыз етеді. Орнату құнын азайту үшін екі осьті трекердің негізгі айналу осінің екі ұшындағы тіректерді трекерлер арасында бөлуге болады. Әдетте трекерлер күн аспанда төмен болған жағдайда бір трекердің басқасын көлеңкеленуін болдырмау үшін өте төмен тығыздықта орналасуы тиіс. TTDAT трекерлерінің артықшылықтары-конструкцияның қарапайымдылығы және аумақтық орналасуының үлкен икемділігі, бұл оларды жеке қолдану мақсатындағы шағын қуатты ФЭС ретінде де, ірі энергетикалық жобаларда да пайдалануға мүмкіндік береді.



1.5сурет - Екі осьтік трекер Tip-Tilt (TTDAT)

1.2.2 Азимут-биіктік екі ось трекері

Екі осьтік азимут-биіктік трекері (немесе alt-azimuth) (AADAT) жерге қатысты вертикалды негізгі азимуталды оське ие. Осылайша, биіктік осі деп те аталатын екінші ось бастапқы оське нормаль болып табылады. Олар TTDAT-қа ұқсас, бірақ олар массивті айналдыру тәсілдерімен ерекшеленеді. AADAT трекерлерінде тірек конструкциясы ретінде үлкен сақина пайдаланылады, олар роликтерге немесе подшипниктері бар үлкен платформаларға орнатылады. Мұндай конструктивті шешімнің артықшылығы, TTDAT жүйелеріндегідей салмақ жүктеменің бір нүктесіне ғана емес, күн батареясының салмағы сақинаның бөліктері бойынша біркелкі таралуы болып табылады. Бұл AADAT-ға күн панельдерінің немесе шағылыстырғыштардың көп массивін орналастыруға мүмкіндік береді және осы типтегі трекерлерді үлкен қуатты күн батареялары бар ФЭЖ-де және жоғары жел жүктемелері бар аудандарда пайдалануға мүмкіндік береді, алайда мұндай жүйелер қымбат болып табылады және AADAT жүйесі сақина диаметрінен бір-біріне жақынырақ орналаса алмайды, бұл жүйе тығыздығын төмендеуіне алып келеді.



1.6 сурет - Азимут-биіктік екі ось трекері

1.3 Күн трекерінде қолданылатын жетек

Күн трекерінің жетегі айналу осінің айналасында КБЖ жылжуын қамтамасыз етеді. Жетек түрі бойынша электр жетегі, гидрожетегі және пассивті жетегі бар жүйелерді ажыратады.

1.3.1 Электр жетегі

Күн бақылау жүйелесінде кеңінен қолданылатыны электр жетегі болып табылады, оларда атқарушы құрылғылар ретінде әртүрлі типтегі электр машиналары қолданылады: сызықтық, кадамдық, сервомоторлар және т.б. ССС электржетегінің негізгі сипаттамалары: атқару құрылғысының типі, оның номиналды қуаты, қуат кернеуінің параметрлері, бақылау режимінде тұтынылатын қуат шамасы, электр энергиясын орташа тәуліктік немесе жылдық тұтынуы болып табылады.

1.3.2 Гидрожетек

Гидрожетекте гидравликалық қысымның пайда болуы үшін сорғылар қолданылады. Қысым гидрожүйе арқылы атқарушы гидромоторларға немесе цилиндрлерге беріледі, олар өз кезегінде қысымды айналмалы немесе

сызықтық қозғалысқа түрлендіреді және Күнді бақылау жүйесін айналу өсі бойынша жылжуын қамтамасыз етеді.

1.3.3 Пассивті жетегі

Пассивті жетегі бар Күнді бақылау жүйелерінде ось бойынша қозғалыс жұмыс сұйықтығы қысымының өзгеру есебінен қамтамасыз етіледі. Қысымның айырмашылығы әр түрлі дәрежедегі көленке есебінен құрылған жылу градиенттерінің нәтижесінде пайда болады. Күн бақылау жүйесі қысым теңгерімсіздігін теңестіру үшін қозғалады.

1.3.4 Бақылау бойынша нұсқаушы

Кейбір дамушы елдерде дискілер трекерді бағыттайтын операторлармен алмастырылған. Бұл сенімділік жағынан, қызмет көрсете алатын персонал болуынан және сол мекенге жақын орналасқан халықты жаңа жұмыс орындарымен қамтамасыз ету жағынан артықшылықтарын береді.

Сіз жүйені қолмен немесе қосқыштар арқылы басқару сигналдарын актуаторларға бере отырып бағыттай аласыз. Бірақ мұндай әдіс негізінен қандай да бір уақыт кезеңінде тиісті көлбеу бұрышы қойылған кездегі трекерлердің маусымдық бағдарлауы үшін ғана қолайлы. Бұл ретте бағдарлау дәлдігі үлкен емес, оператор үнемі трекердің жаңында бола алмайды, сондықтан бұл әдіс кең таралмаған, бірақ аз бюджеттік жүйелерде маусымдық бағдарлау үшін ол өте ыңғайлы.

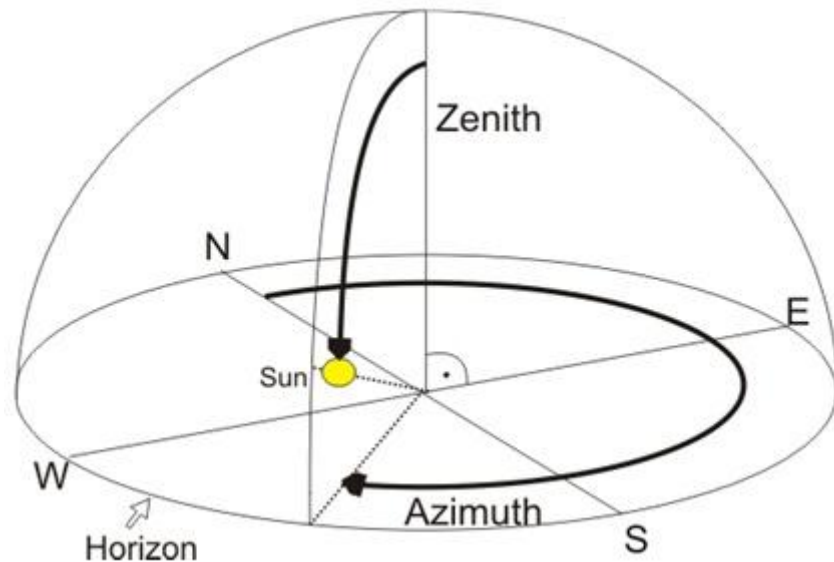
1.4 Күн трекерінің басқару тәсілі бойынша жіктелуі

Басқару жүйесінің негізгі сипаттамасы ретінде бақылау алгоритмі десек болады. Тәжірибеде күннің жағдайын қадағалаудың екі негізгі алгоритмі қолданылады – астрономиялық және жарық датчиктері бойынша.

Техникалық іске асырудың қарапайымдылығы салдарынан бірнеше фотоқабылдағыштардың көмегімен бақылау алгоритмі жиі қолданылады. Фотоқабылдағыштар-аспандағы күннің орны туралы ақпаратты алатын датчиктер, оның жұмыс істеу принципі күн батареясын аспандағы жарықтың максималды қарқындылығына орналастыруға негізделген. Олар барлық фотоэлементтердегі жарық ағыны бірдей болатын сәтке дейін басқарушы сигналдарды актуаторларға береді. Күннің қозғалысы салдарынан туындаған жүйенің тепе-теңсіздігі күн көзіне қарай жаңа қозғалысты жасауға импульс береді. Мұндай құрылғылардың принципіалдық сұлбалары қарапайым және арзан келеді. Жарық датчиктері бар трекер жүйелерінің маңызды кемшілігі -

бұлтты ауа-райында, қарқынды жаңбыр мен фотодетекторлардың ластануы кезінде жұмыс істемеуі.

Астрономиялық алгоритмдер неғұрлым сенімдірек жұмыс жасайды. Соның ішінде ең жоғары тиімділік күн позициясы алгоритмімен (SPA) қамтамасыз етілген, оның жұмыс принципі күн трекерінің орналасқан жерінің көрсетілген параметрлері бойынша күн батареясының оңтайлы орналасу бұрыштарын бағдарламалық есептеу негізінде құралған: ендік, бойлық, теңіз деңгейінен биіктігі



1.7 сурет - Көкжиектегі күн қозғалысы

Тіктеме сызық (немесе тік сызық) – тік бойынша орналасқан, аспан сферасының орталығы арқылы өтетін тік сызық. Тіктеме сызықтың жоғарғы нүктесі зенит (Z), төменгі нүктесі надир (Z') деп аталады.

Шын (математикалық, немесе астрономиялық NEWS шеңбері) горизонт - аспан сферасының үлкен шеңбері, оның жазықтығы тіктеме сызыққа перпендикуляр. Шын горизонт аспан сферасының бетін екі жарты сфераға бөледі: төбесі зенитте орналасқан көрінетін жартысфера және төбесі надирде орналасқан көрінбейтін жартысфера. Шын горизонт бақылау нүктесінің жер бетінен көтеріңкілігіне, сондай-ақ атмосферадағы жарық сәулелерінің қисық болуына байланысты көрінетін горизонтпен сәйкес келмейді.

Айналасында тәуліктік айналым жүретін аспан сферасының диаметрі әлемнің осі (PP') деп аталады. Әлемнің осі Жердің айналу осіне сәйкес келеді. Әлемнің осі аспан сферасының бетімен екі нүктеде қиылысады - әлемнің солтүстік полюсі және әлемнің оңтүстік полюсі. Аспан сферасының көрінімдік айналымы Жердің өз осінен айналуына қарама-қарсы бағытта жүреді.

Жазықтығы әлемнің осіне перпендикуляр аспан сферасының үлкен шеңбері аспан экваторы (QQ'). деп аталады. Аспан экваторы аспан сферасын екі жартышарға бөледі: солтүстік және оңтүстік.

Аспан меридианы ($ZPQZ'P'Q'$) - бақылаушының полюсі мен зениті арқылы өтетін аспан сферасындағы үлкен шеңбер. Оның N нүктесінде шын горизонтпен қиылысуы солтүстік нүктесі, ал S нүктесінде қиылысуы оңтүстік нүктесі деп аталады.

Аспан экваторының шын горизонтпен E нүктесінде қиылысуы шығыс нүктесі, W нүктесінде - батыс нүктесі деп аталады.

Талтүстік сызық - N және S нүктелерін жалғайтын түзу сызық.

Шырақтың тәуліктік параллелі (KCK_1) – аспан экваторына параллель шырақ арқылы жүргізілген аспан сферасындағы кіші шеңбер.

2.2 Координаталық жүйелер

Әр аспан шырағына сфера орталығын шырақтың орталығымен байланыстыратын түзу сызық оны қиып өтетін аспан сферасындағы нүкте сәйкес келеді. Аспан сферасындағы шырақтардың күйі мен көрінетін қозғалыстарын зерттеу кезінде сфералық координаталардың бір немесе басқа жүйесі таңдалады. Шырақтардың аспан сферасындағы күйін есептеу аспан механикасы мен сфералық тригонометрияның көмегімен жүргізіледі және сфералық астрономия пәні болып табылады.

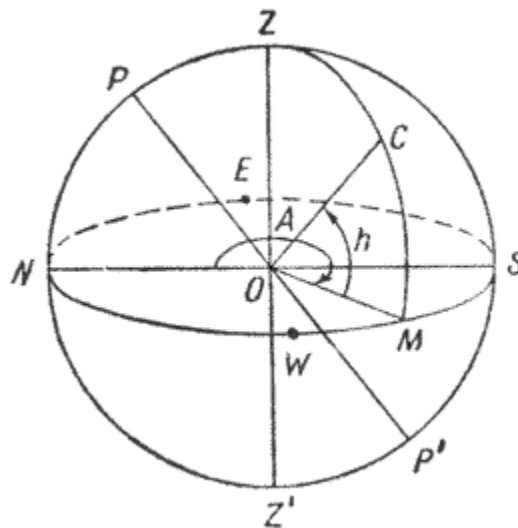
Аспан координаттар жүйесі астрономияда шырақтардың аспандағы орнын немесе жорамалды аспан сферасындағы нүктелердің орнын сипаттау үшін қолданылады. Шырақтар мен нүктелердің координаталары градусық немесе сағаттық шамамен өрнектелетін, сфераның үлкен доғаларын білдіретін екі шамамен беріледі. Осылайша, аспан координаттарының жүйесі сфералық

координаттар жүйесі болып табылады, онда үшінші координат - қашықтық - көбінесе белгісіз және рөл атқармайды.

Аспан координаттарының жүйелері бір-бірінен негізгі жазықтықты (негізгі жазықтықты қараңыз) және есеп басын таңдауымен ерекшеленеді. Қойылған міндетке байланысты бір немесе басқа жүйені пайдалану ыңғайлы болуы мүмкін, дегенмен аспан шырақтарының аспан сферасындағы орны екі сфералық координаттармен нақты анықталады. Ең жиі қолданылатын көлденең және экваторлық координат жүйелері. Сирек - эклиптикалық, галактикалық және басқалар.

Күннің орталығы жыл бойы эклиптика деп аталатын сфераның үлкен шеңберінің айналасында қозғалады. Бұл көрінімдік қозғалыс Жердің Күн айналасында айналуымен байланысты орын алады; бағыты бойынша ол аспан сферасының тәуліктік айналымына кері.

Көлденең координаталар жүйесі:



2.2 сурет -Горизонтальді координаталар

Бұл жүйеде шырақтың орны анықталатын негізгі шеңберлер шын горизонт және аспан меридианы болып табылады; олар, өз кезегінде, шырақтың биіктігі (h) және оның азимуты (A) координаттары болып табылады.

Шын горизонттың жазықтығы мен аспан сферасының ортасынан шыраққа бағыт бағыты арасындағы доғасы шырақтың биіктігі h деп аталады. Биіктікті шын горизонттың жазықтығынан шырақтың альмукантаратына дейінгі доғасымен де өлшеуге болады. Биіктік 0-ден $\pm 90^\circ$ -қа дейін өлшенеді.

Шырақтың оң биіктіктері зенитке, ал терістері надирге қарай есептеледі, яғни горизонттың үстінде орналасқан шырақтар оң биіктікке, ал горизонттың астында орналасқандар теріс биіктікке ие болады.

$$h = \arcsin (\cos t \cdot \cos \delta \cdot \cos \varphi + \sin \varphi \cdot \sin \delta) \quad (2.1)$$

Азимут A – талтүстік сызық (шамамен айтқанда, оңтүстікке бағыт) және мат.горизонт жазықтығының шырақтың тік шеңбер жазықтығымен қиылысу сызығы арасындағы бұрыш. Ол оңтүстік нүктесінен аспан сферасының тәуліктік айналым бағыты бойынша $0^\circ \dots 360^\circ$ аралығында немесе 0° -ден $+180^\circ$ -ге дейін батысқа және 0° -ден 180° шығысқа қарай есептеледі.

$$A = \text{arcctg}(\text{ctgt} \cdot \sin \varphi - \text{tg} \delta \cdot \cos \varphi \cdot \text{cosect}) \quad (2.2)$$

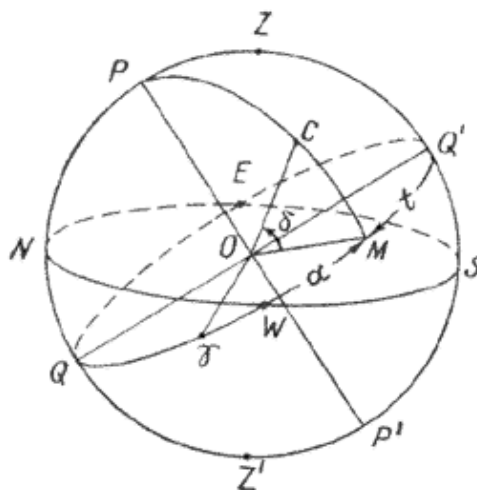
Шырақтың көлденең координаттары Жердің тәуліктік айналымына байланысты үнемі өзгеріп отырады.

Аспан сферасының айналуы кезіндегі координаттардың өзгеруі

Шырақтардың биіктігі h , зенит қашықтығы z , азимуты A және сағаттық бұрышы t аспан сферасының айналуына байланысты үнемі өзгеріп отырады, өйткені олар осы айналумен байланысты емес нүктелерден саналады. Шырақтардың қисаюы δ , полярлық қашықтығы p және тік шығуы α аспан сферасының айналуы кезінде өзгермейді, бірақ олар тәуліктік айналыммен байланысты емес шырақтардың қозғалуына байланысты өзгеруі мүмкін.

Көрініп тұрғандай, көлденең координаталарды есептеу үшін, шырақтың қисаюы мен сағаттық бұрышын есептеу керек. Бұл координаттар бірінші экваторлық координаттар жүйесіне қатысты, мұнда негізгі жазықтық аспан экваторының жазықтығы болып табылады.

Экваторлық координаталар жүйесі:



2.3 сурет -Экваториальді координаталар

Бұл жүйеде (12-сурет) шырақтың орны анықталатын негізгі шеңберлер аспан экваторы және аспан меридианы болып табылады. Координаталар: шырақтың қисаюы δ , оның сағаттық бұрышы t .

Шырақтың қисаюы- бұл аспан экваторының жазықтығы мен шырақтың бағыты арасындағы бұрыш. 0° -ден $\pm 90^\circ$ -қа дейін өлшенеді (оң мән

экватордан солтүстікке қарай, теріс мән – оңтүстікке қарай). Бір географиялық ендікте талтүсте Күннің зенит қашықтығының немесе биіктігінің өлшемдерімен (яғни оның жоғары кульминация сәтінде) Күннің жыл ішінде қисаюы $+23^{\circ} 27'$ бастап $23^{\circ} 27'$ дейінгі аралықта, нөл арқылы өтіп жылына екі рет өзгеретіні анықталды.

$$\delta = 23,45 \left[\frac{360}{365} (N + 284) \right] \quad (2.3)$$

Аспан меридианының батысына қарай аспан экваторының бойымен өлшенген бұрыштық қашықтық (оның ең жоғары кульминация сәтінде шырақтар кесіп өтетін бөлігі) аспан сферасындағы таңдалған нүктеден өтетін сағат шеңберіне дейін. Жұлдыздың сағаттық бұрышы жұлдыздық уақыттан жұлдыздың тік шығуының айырымына тең болады. Сағаттық бұрыш 24 сағат = 360° ($1 \text{ сағат} = 15^{\circ}$, $1 \text{ мин} = 15'$) негізінде сағаттық өлшеммен өрнектеледі.

$$t = T_{\text{ш.к}} - 12:00\text{с} \quad (2.4)$$

Шын күн тәуліктері – Күннің екі бір атаулы кезекті орталық кульминациялары арасындағы уақыт аралығы. Шын Күн уақытын жергілікті уақытты, сағаттық белдеуді және бақылау нүктесінің бойлығын біле отырып есептеуге болады:

$$T_{\text{ш.к}} = T_{\text{о.к}} + EOT, \text{ немесе } T_{\text{ш.к}} = UTC + \lambda + EOT \quad (2.5)$$

мұндағы $T_{\text{о.к}}$ – Жердің бір нүктесінде күннің орташа уақыты (нүктенің бойлығына байланысты), UTC – бүкіләлемдік үйлестірілген уақыт, λ – сағат бірлігіндегі бақылау нүктесінің бойлығы, EOT – уақыт теңдігі.

Бүкіләлемдік үйлестірілген уақытты жергілікті уақыттан (T_m) және сағат белдеуінен (N) есептеуге болады: $UTC = T_m - N$. Орналасу бойлығы λ нөлдік меридианнан (шығыс бойлық) 0° бастап 180° дейін шығысқа қарай және 0° бастап -180° дейін батысқа қарай (батыс бойлығы) өлшенеді. Жоғарыда келтірілген формулаға алмастыру кезінде бойлықты сағат бірлігіне айналдыру керек ($1^{\circ} = 4 \text{ мин}$).

Уақыт теңдеуі EOT күннің орташа уақыты мен шын күн уақыты арасындағы айырмашылықты көрсетеді, өйткені Күннің тәуліктік қозғалысы Жер орбитасының эллиптілігіне, сондай-ақ Жердің осінің эклиптикалық жазықтыққа бейімделуіне байланысты біркелкі емес:

$$EOT = 9,87 \cdot \sin 2B - 7,53 \cdot \cos B - 1,5 \cdot \sin B, \quad (2.6)$$

мұндағы $B = (360^{\circ} (N-81))/365$, N – күннің жыл ішіндегі реттік нөмірі.

2.3 Алматы қаласы үшін күн шырағының координатасын есептеу

Алматы 77 бойлықта және 43 ендікте орналасқан

Алматы үшін кез келген уақыт моментіге Күн координаталары келтірілген формулалар бойынша келесідей анықталады. Есеп 2-ші шілде сағат 12:00 үшін алынған

UTC=12:00-6=6:00 Алматы 6-шы сағаттық белдікте орналасқан

77° ендік сағат бірлігінде 5:08 тең

$B = (360^\circ (N-81))/365 = 71$ N=153 ол 2-ші шілденің жылдық кезегі

$EOT = 9,87 \cdot \sin 2B - 7,53 \cdot \cos B - 1,5 \cdot \sin B = 9,87 \cdot \sin 142 - 7,53 \cdot \cos 71 - 1,5 \cdot \sin 71 = 6,12 - 2,48 - 1,425 = 2,215$

$T_{ш.к} = UTC + \lambda + EOT = 6:00 + 5:08 + 0:2 = 11:10$

$t = T_{ш.к} - 12:00 = 11:10 - 12:00 = -50$ минут, градуста = -12,5°

$\delta = 23,45 \left[\frac{360}{365} (N + 284) \right] = 22.277^\circ$

$h = \arcsin (\cos t \cdot \cos \delta \cdot \cos \varphi + \sin \varphi \cdot$

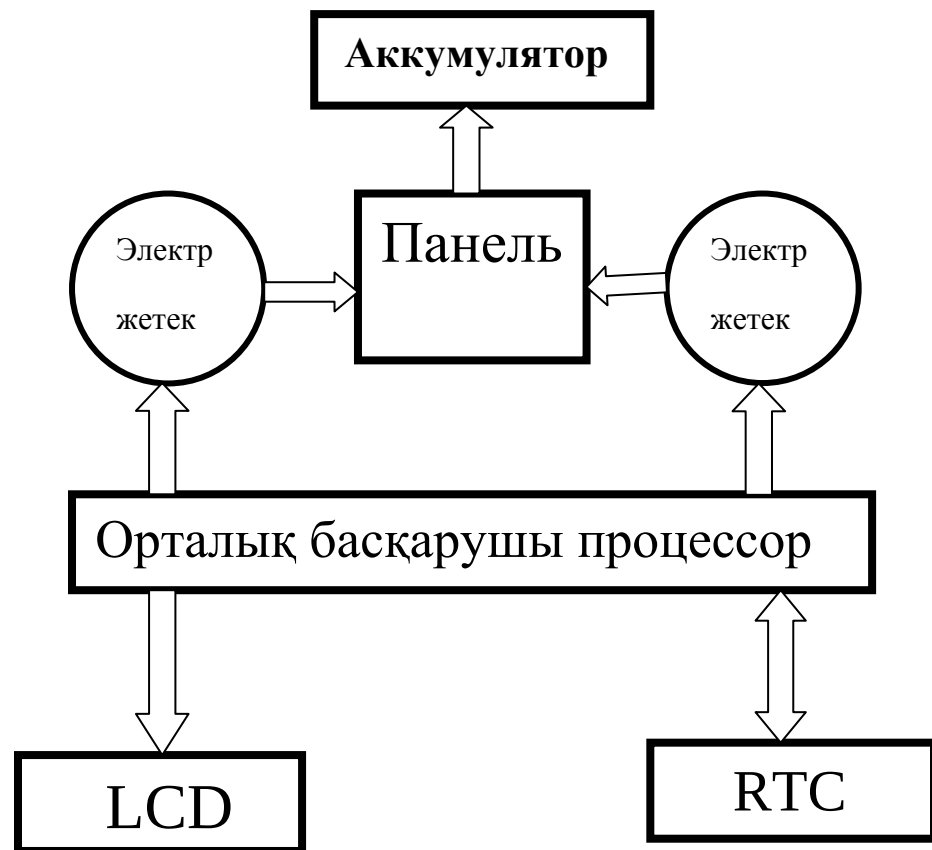
$\sin \delta) = \arcsin (0,98 \cdot 0,73 \cdot 0,92 + 0,38 \cdot 0,65) = \arcsin (0,92) = 66.92^\circ$

$A = \text{arcctg}(\text{ctgt} \cdot \sin \varphi - \text{tg} \delta \cdot \cos \varphi \cdot \text{cosect}) = \text{arcctg}((-4.5) * 0.68 - 0.4 * 0.73 * (-4.62)) = \text{arcctg}(-3.06 + 1.34) = 149.826^\circ$

3 Күн жүйе макетінің схемасы

Дипломдық жұмыстың мақсаттарының бірі екі осы ті Күн бақылау жүйесінің шағын макетін жасау болып табылады. Сол орайда төменде шағын күн трекерінің функционалды және принципіалды схемалары көрсетілген. Дегенмен SPA алгоритмін аппараттық тұрғыда толықтай жүзеге асырады.

3.1 Трекердің функционалды схемасы

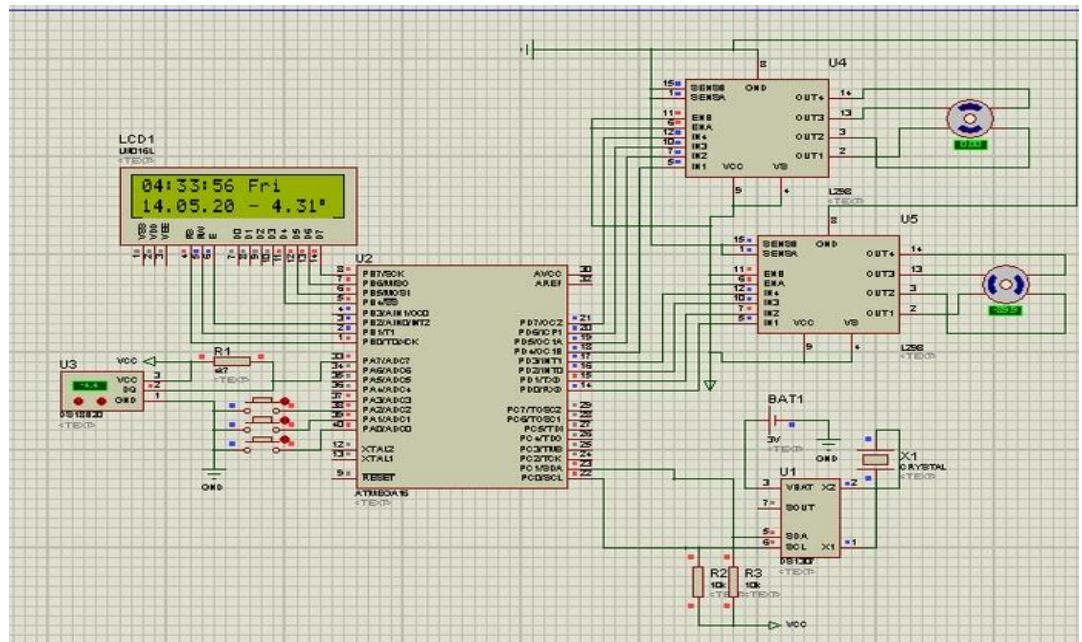


3.1 сурет -Трекердің функционалды схемасы

Бұл Күн бақылау жүйесі RTC модулінің негізінде құрастырылған. RTC модулі дәл қазіргі уақыт ағымын көрсету үшін қажет. Ол модульден келген ақпарат Орталық басқару процессорінде алдын ала енгізілген алгоритм бойынша есептеліп электржетектердің бұрылу бұрышы және бағыты тағайындалады. Сол жетектердің бағыты негізінде Күн панелінің Көкжиектегі Зенитті және Азимуталды позициясы анықталады. Сол арқылы Күн панелі жалғанған аккумуляторды зарядтайды, және сол аккумулятордың қуатынан бақылау жүйесі қоректенеді. Ағымдағы уақыт шамасы LCD модулінде көрсетіліп тұрады.

3.2 Трекердің принципіалды схемасы

Бұл жұмыс макетін жасау барысында тек дайын модульдер қолданылды.



3.2 сурет -Трекердің принципіалды схемасы

Бұл жүйенің басты құрылғысы ATMEGA 16 микроконтроллері болып табылады.

ATMEGA 16 микроконтроллерінің басты сипаттамасы

VR RISK архитектурасы негізінде құрастырылған:

-125 тез жылдамды командалары бар, олардың көпшілігі бір такт циклінде орындалады

-8-разрядты 32 жалпы мақсаттағы регистрлерден тұрады

-Тактілік жиілігі 16МГц

Жады:

-16 КБайтты Flash жады

-512 байт EEPROM

-1 КБайт Оперативті сақтау жадысы

Енгізілген периферия :

-Арнайы бөлгіші бар екі 8-разрядты санауыш таймері бар, олардың біреуі салыстыру режимінде жұмыс жасай алады

-Бір 16-разрядты таймер

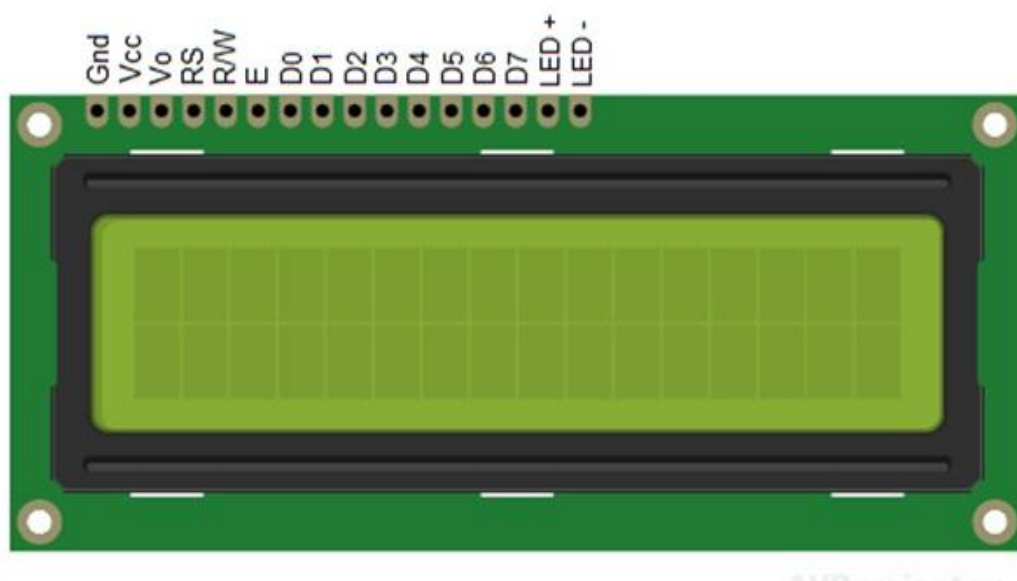
-4 PWM каналы

-10 разрядты 8 АСТ каналы бар

-Жұмыс кернеуі 4,5-5,5В

-Жұмыс істеу жиілігі 0-16МГц

Lm016l индикаторы:



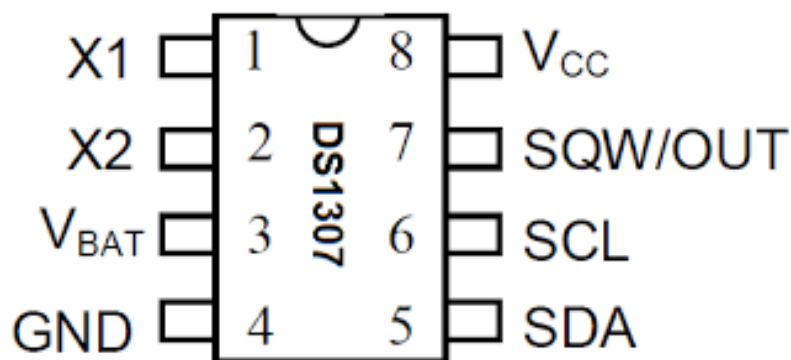
3.3 сурет -Lm016l индикаторы

Мұндағы:

- GND-жер (кейде VSS жазады);
- Vcc қуат кернеуі +5В (кейде Vdd жазады);
- Vo-0В-дан +5В-ға дейінгі контраст кернеуі, бұл қорытындыны реттеу үшін потенциометрге қосу керек;
- RS - дисплей оған деректер не пәрмендер түсетінін анықтайды;
- RW-дисплей Деректерді жіберуді немесе алуды анықтайды;
- E-синхрондау сызығы;
- D0 – D7-командалар/деректер шинасы;
- LED +, LED - - жарық беруге арналған қорытындылар;

Дисплей 2 режимінде жұмыс істей алады: 8 биттік (яғни, ақпарат алмасу үшін D0-дан D7-ге дейінгі байланыстар пайдаланылады) деректер бір тактқа жіберіледі, 4 разрядтық (алмасу үшін тек D4 – D7 байланыстары ғана қолданылады), бұл жағдайда деректер 2 тактта беріледі, алдымен алғашқы 4 бит, одан кейін келесі 4 бит ақпараты жіберіледі. Біз микроконтроллердің шықпаларын үнемдеу мақсатында 4 разрядтық режимді қолданамыз

DS1307:



3.4 сурет -DS1307 модулі

Мұндағы:

- Қуат кернеуі: 5В;
- Жұмыс температурасы: - 40°C ... + 85°C;
- Жады: 56 байт (энергияға тәуелді);
- Батарея: LIR2032 (автоматты қуат көзін анықтау);
- Интерфейс: I2C;
- Өлшемдері: 28мм x 25мм x 8 мм;

DS1307 уақытты есептеуге арналған шағын модуль. Литий батареикасынан (LIR2032) қоректендіруді жүзеге асыратын ds1307zn микросхема базасында жиналған, бұл ұзақ уақыт бойы автономды жұмыс істеуге мүмкіндік береді. Сондай-ақ, модульде көлемі 32 Кбайт (AT24C32) EEPROM энергияға тәуелсіз жады орнатылған. Мекен-жайлар мен деректер екі сымды екі бағытты шина бойынша дәйекті түрде беріледі. AT24C32 және DS1307ZN микросхемасы I2C жалпы шиналық интерфейсмен байланысты.

DS1307 модулін пайдалану көбінесе өте ақталған, мысалы, деректер сирек, бір аптадан астам аралықпен саналған кезде, контроллердің меншікті ресурстарын пайдалану орынды емес немесе мүмкін емес.

- VCC негізгі қорек көзі;
- X1, X2 32.768 кГц кварц резонаторының қосылуы;
- VBAT қосалқыбатарея жалғану кірісі(+3 В);
- GND ортақ шина;
- SDA ақпараттың кіріс шығыс тасымалы;
- SCL тізбекті байланыс синхронизациясы;

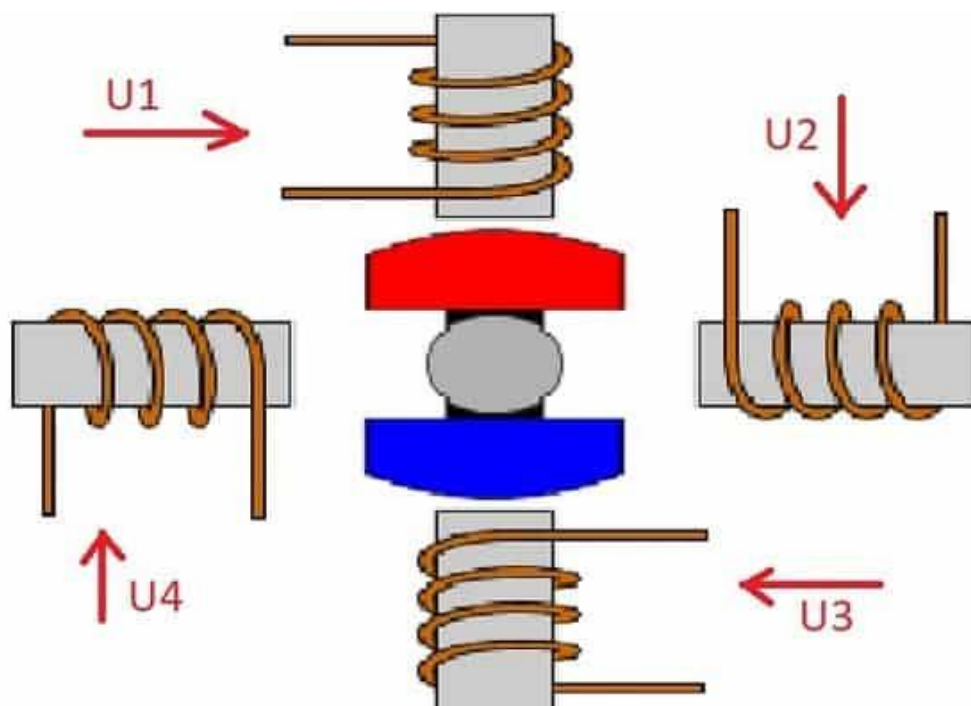
L298 – екі бағытты жүктемелерді басқаруға арналған екі көпірлі сұлба болып табылады. Ол 2А-ге дейін және 4,5- 46В кернеу аралығында жұмыс жасайтын жүктемелерге негізделген. Сұлба релелерді, соленоидты, қадамдық жетектерді қозғалтуға арналған. L298 драйверінде логикалық сұлба және жүктеме бөлек қоректенеді. Ол микросхемадан қарағанда көбірек немесе азырақ кернеу қажет ететін жүктеме қосуға мүмкіндік береді және шу ақаулығын азайтады.

L298 микросхемасы қызып кетуден сақтайтын еңгізілген қорғанысы бар. Жүйенің шығысы тек $+70^{\circ}\text{C}$ температурасы кезінде ғана өшеді.

- Vss — Қозғалтқыш кернеуі 4,5- 46В аралығында;
- GND- ортақ жер;
- Vs- логикалық сұлбаның кернеуі;
- ENA- А қозғалтқышының жылдамдығын басқару үшін қолданады;
- IN1 және IN2 - А қозғалтқышының бағытын басқару үшін қолданады;
- IN3 және IN4 - В қозғалтқышының бағытын басқару үшін қолданады;
- ENB - В қозғалтқышының жылдамдығын басқару үшін қолданады;
- OUT1 және OUT2 – А қозғалтқышының кірісі;
- OUT3 және OUT4 – В қозғалтқышының кірісі;

Қадамдық қозғалтқыш

Қадамдық қозғалтқыштарды пайдалану дәл позициялау жүйелерін іске асыру үшін ең қарапайым, арзан және жеңіл шешімдердің бірі болып табылады. Қадамдық қозғалтқыш электр энергиясын механикалық энергияға түрлендіруші. Конструкциялық тұрғыда статор орамынан және магнитті-жұмсақ ротордан тұрады.

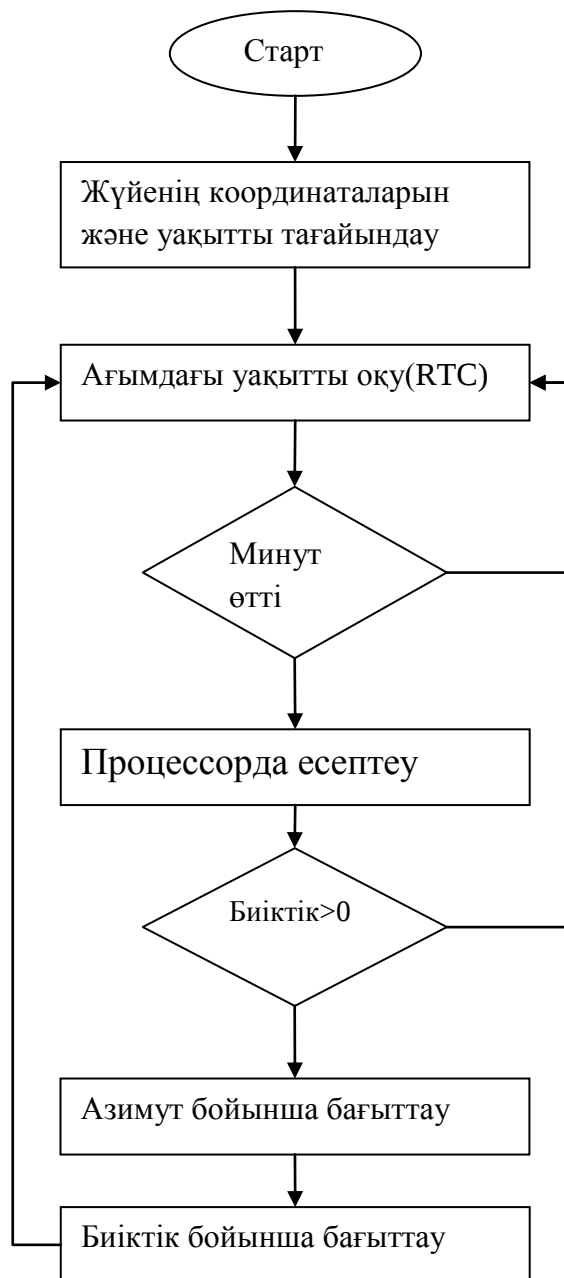


3.5 сурет -Қадамдық қозғалтқыштың жұмыс принципі

3.5 суретте орамнан тұратын статор бейнеленген және олар бір-біріне қатысты 90 бұрышта орналасқан. Сондықтан ол 90 бұрыштық қадамға ие. U_1 , U_2 , U_3 , U_4 кернеуін кезек беру әсерінен білік толық айналым жасайды. Толық айналым біткен соң цикл қайта қайталанады.. Айналым бағытын өзгерту үшін импульстер кезегін ауыстыру қажет.

3.3 Трекердің жұмыс алгоритмі

Құрастырылған схеманың жұмыс ісетеу процессін қамтамасыз ететін алгоритмі құрастырылып оның бағдарламалық коды алынған



3.6 сурет -Трекердің жұмыс алгоритмі

Алгоритм бойынша Орталық есептеуші процессор әр берілген уақыт интервалы бойынша ағымдағы уақыт арқылы есептеулер жүргізеді.

Соның нәтижесінде егер Күн көкжиектің көрінетін аймағында болса, яғни биіктігі 0 градустан артық болса азимут және биіктік бойынша қадамдық

қозғалтқышты қозалтады. Жүйенің программасы асемблер тілінде AVR STUDIO 4 бағдарламасында жазылды.

3.4 Ассемблер тілінде бағдарламаға шолу

Енді жоғарыда көрсетілген алгоритмді программалық тұрғыда қарап өтсек.

Программаның негізгі блоктарын қарастыратын болсақ , кейіннен әрқайсысына қысқаша жеке тоқталатын боламыз.

- Тақырып файлын еңгізу
- Жады орнату
- Reset
- Proga
- Ішкі программалар

Тақырып файлын еңгізу

Бұл блокта ATMEGA 16 контроллерінің "m16def.inc" кітапханасын еңгіземіз

Кітапхана өзінде микроконтроллердің барлық регистрлерін, порттардың атауын және константаларды сақтайды. Бұл бөлікте тағы жұмыстың ыңғайлылығы үшін жалпы мақсаттағы регистрлерге атау тағайындаймыз және процессордың жұмыс істеу жиілігін орнатамыз.

Жады орнату

Бұл жобада біз тек энергияға тәуелді ОСҚ арқылы жұмыс жасаймыз. Оперативті жүйеде айнымалылармен жұмыс істеу үшін әрқайсысына ұяшықтар тағайындаймыз. 6 байт- аралық санауыштарға болса , 7 байт- уақыт ақпаратына арналған. Тағы да программа сегментін және үзіліс векторларын көрсету тиіс.

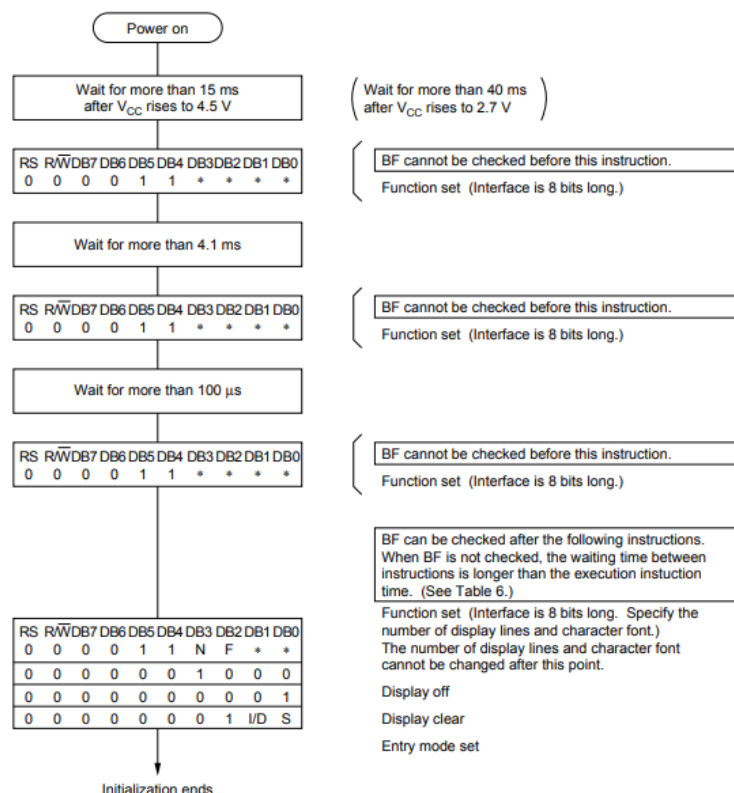
Reset

Программаның алдын-ал орнатуларын жүргіземіз. Ол программада бір рет орындалады , содан оралмайды

Ең алғашқыда стэк облысын көрстеміз, ол әдетте жадының соңына апарылады. Стэк программamızдағы ішкі программалардың қателіксіз орындалуы үшін қажет.

Сыртқы перифериямен жұмыс жасайтын порттарды тағайындаймыз. Порттын оқу немесе жазу режимі DDR регистрі арқылы жүзеге асады. Микроконтроллер бізге ақпаратты жазу кезінде ішкі регистрді қосуға мүмкіндік береді.

Келесіде LCD дисплейін инициализациялау қажет. Жұмыста қолданылған LM016L дисплейі HD44780 контроллері негізінде құрстырылған. Дисплейдің бастапқы параметрлерін тағайындау үшін құрылғының құжаттамасында көрсетілген нұсқаулық бойынша қадамды түрде орнатылуы тиіс.



3.7 сурет - Hd4478 инициализациялау реті

Proga:

```
rcall Proga_Time_Read // DS1307 ден уақытты алу
rcall Proga_Otoibr_Time // LCD дисплейіне шығару
rcall Proga_Shagov_Dvig // Қозғалтқышты қозғалту
ldi try, 4 // Шамаен жарты секунд кідіріс
```

Proga0:

```
sbis PINA, 0 // Мәзірге кіру батырмасын тексереміз
rjmp Menu_Nastr_Hoir
rcall Delay
dec try
brne Proga0
rjmp Proga // Proga-ға секіру
```

Біздің программамыз осы цикл ішінде айналып отырады. Ең алдымен уақыт модулінен ағымдағы уақытты оперативті жадыға жазамыз. Келесі командада сол нәтижені дисплейде шығарамыз. Соның нәтижесінде тағайындалған уақыт интервалы сайын қадамдық қозғалтқыштарды басқарамыз. Егер жұмыс процессі кезінде мәзірдегі көрсетілген уақыт шамасын өзгерту қажет болатын болса А портындағы нөлінші пинге сигнал беру қажет. Сол жағдайда программа Menu_Nastr_Hoir ішкі программасына өтеді. Сол арқылы DS1307 модуліндегі ағымдағы уақытты 1-ші және 2-ші Пиндерге жалғанған

батырмалар арқылы өзгерте аламыз. Циклге қайту үшін 0-ші піндегі батырманы қайта басу қажет

Ішкі программалар

Программаның ең ауқымды және көлемді бөлігі болып ішкі программалар болып табылады. Ішкі программалар rcall командасы арқылы шақыртылады. Шақырту алғандағы адресі стэк аумағында жазылады және ret командасы бойынша программаны қайта орнына апарады. Жоба көптеген ішкі программалардан тұрады, солардың ішіндегі ең маңыздысы уақытты оқу (Proga_Time_Read) бөлігін қарастырып өтейік.

RTC модулі I2C шинасы арқылы жұмыс жасайды, оны two-wire немесе қысқаша TWI деп те атайды, яғни екі сым бойынша ақпарат алмасу. I2C шинасы арқылы ақпарат алмасып жатқан апараттар кез келген уақытта Master немесе Slave ретінде жұмыс жасай алады. Бұл жобада AVR контроллері Master ал ds1307 Slave болады. ATMEGA контроллерінде I2C ге арналған арнайы бөлігі бар, ол интерфейспен жұмысты жеңілдетеді . TWI модулімен жұмысжасау үшін оның регистрлерімен таңыс болу қажет. Оны бізге AVR STUDIO4 бағдарламасы жақсы көрсетеді

Name	Address	Value	Bits
TWAR	0x02 (0x22)		□□□□□□□□
TWBR	0x00 (0x20)		□□□□□□□□
TWCR	0x36 (0x56)		□□□□□□□□
TWINT			□
TWEA			. □
TWSTA			. . □
TWSTO			. . . □ . . .
TWWC			 □ . .
TWEN			 □ .
TWIE			 □
TWDR	0x03 (0x23)		□□□□□□□□
TWSR	0x01 (0x21)		□□□□□□□□

3.8 сурет -TWI регистрлері

Кез келген интерфейсте оның ақпарат тасымалдау жылдамдығын беру қажет. Оны модульде TWBR және TWSR статус регистрінің екі кіші биттары TWSR1, TWSR0 орындайды. Біздің ақпаратты салып немесе оқыйтын регистрі болып 8-разрядты TWDR табылады. TWAR регистрі микроконтроллер SLAVE режимінде жұмыс жасағанда қажет болады. Үлкен 7 разряд адрес құрса кіші бит ортақ шинаға жауап ретінде қолданылады . TWCR регистрі TWI модулінің ең басты модулі болып келеді. Оның әр регистрі өзіндік функцияны орындайды және оларды жеке қарастыру қажет.

TWINT- TWI модулінің үзіліс орнатылу туы. Ол модуль операцияны аяқтағанда апаратты түрде орындалады

TWEA- Ол Master немесе Slave ақпарат алған жағдайда растау сигналын жібереді(Ack)

TWSTA- Старт жағдайының туы

TWSTO- Стоп жағдайының туы

TWWC- Конфликт жағдайында қойылатын ту

TWEN- TWI модулінің жұмыс істеу рұқсат биті

TWIE- TWI модулінің үзуіне рұқсат бер биті

Осы регистрлерді пайдалана отырып DS1307 сағат модуліне ақпарат жазу және алу программасы жазылған болатын. TWI режимінде жазу немесе оқу алгоритмдері ұқсас болып келеді, сондықтан біз тек бір жағдайды қарастыратын боламыз. Төменде контроллердің оқу режимінде жұмыс жасау коды көрсетілген

```
rcall TWI_Start // Шинаны тексереміз
```

```
ldi r16, 0xD0 // модуль адресін жібері
```

```
rcall TWI_TByte
```

```
ldi r16, 0x00
```

```
rcall TWI_Tbyte // ішкі регистр адресін жіберу
```

```
rcall TWI_Start // реестарт
```

```
ldi r16, 0xD1
```

```
rcall TWI_Tbyte // Оқу битін жіберу
```

```
rcall TWI_RByteAck
```

```
sts TimeWatch, r16
```

```
rcall TWI_RByteAck
```

```
sts TimeWatch+1, r16
```

```
rcall TWI_RByteAck
```

```
sts TimeWatch+2, r16
```

```
rcall TWI_RByteAck
```

```
sts TimeWatch+3, r16
```

```
rcall TWI_RByteAck
```

```
sts TimeWatch+4, r16
```

```
rcall TWI_RByteAck
```

```
sts TimeWatch+5, r16
```

rcall TWI_RbyteNack // Nack жауабы келген кезде Stop операциясы орындалады

```
sts TimeWatch+6, r16
```

```
rcall TWI_Stop
```

```
ret
```

ҚОРЫТЫНДЫ

Қорытындай келе, қазіргі таңда маңызды болып келетін балама энергиясның пайдалы әрекет коэффициентін арттыру жұмысы атқарылған болатын. Ол көкжиектегі күн бағыты бойынша қозғалатын Күн трекер жүйелері арқылы қол жеткізілді. Жобада Күн трекерінің ось саны бойынша классификациясы, қозғалтқыш типтері бойынша қолданылысы және панель басқару жолы бойынша жіктелініп әрқайсының қолдану аймағы, артықшылығы немесе кемшілігі көрсетілді. Соның ішінде, өндірісте көбірек қуат өндіретін екі осьті Күн трекері жұмыс мақсатына таңдалынған болатын.

Күн трекерінің көрінетін көкжиек бойында ориентациясы үшін аспан механикасы бойынша зерттелік жұмыстар жүргізілді. Яғни, аспан сферасы ұғымы және ондағы негізгі сызқтар көрсетілді. Көкжиектегі Күн орнын есептеу үшін екі координаталық жүйе қолданылды: горизонтальді және экваторлық. Әрқайсының өзіне тән шамалары және негізгі бұрыштары бойынша формулалары алынды. Мысал ретінде Алматы қаласының координаталары бойынша есептеу жолы көрсетілді.

SPA алгоритмін жүзеге асыратын трекердің функционалды және принципіалды схемасы сызылды. Оны Proteos бағдарламасынды құрастырып тексеру жұмыстары атқарылды.

AVR STUDIO 4 бағдарламалық платформасында ассемблер тілінде жобаның алгоритмін жүзеге асыратын жұмыс коды алынды.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Ревич Ю. В. P32 Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. — 2-е изд., испр. — СПб.: БХВ-Петербург, 2011. — 352 с.: ил. — (Электроника)
- 2 Столяров А. В. C81 Программирование: введение в профессию. II: Низкоуровневое программирование. М : МАКС Пресс, 2016. 496 с.
- 3 ГОСТ Р 57229–2016. Системы фотоэлектрические. Устройства слежения за Солнцем. Технические условия. – М.: Стандарт информ, 2016. – 64 с.
- 4 <https://pdf1.alldatasheet.com/datasheetpdf/view/174743/ATMEL/ATMEGA16.html>
- 5 <https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>
- 6 <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
- 7 https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf
- 8 <https://radiolaba.ru/microcotrollers/solnechnyy-treker.html>
- 9 http://scask.ru/q_book_aa.php?id=34
- 10 <http://ust.su/solar/media/section-inner79/12844/>
- 11 Солнечная энергетика: Учебное пособие для вузов / В.И. Виссарионов, Г.В. Дерюгина, В.А. Кузнецова; Под ред. В.И. Виссарионов. - 2-е изд.
- 12 С.У.Исмаилов Тіректі-айналмалы механизмді күн батареясының жұмысын басқаруға арналған Heliostat_and_aktuator_01 бағдарламалық қамтамасыздандырудың мүмкіндіктеріне шолу жасау Вестник КазНТУ.№3,Алматы 2015.,стр.222-226 ISBN 1680-9211
- 13 Тхеин Лин У. Система управления шагового двигателя для перемещения солнечной рамы энергетической установки. Естественные и технические науки. - М.: «Компания Спутник+». № 1. 2009.- С. 292–295.
- 14 Comparison of performance of solar photovoltaics on dual axis tracker with fixed axis at 13°N latitude / R.R. Rao, H.R. Swetha, J. Srinivasan, S.K. Ramasesha // Current science. – 2015. – V. 108 (11). – P. 2087–2094.
- 15 А. с. 900262 (СССР). Следящая система для солнечной электростанции. Хакимов Р.А., Захидов Р.А. - Б.И. 1982. № 3.
- 16 Tilt and azimuth angles in solar energy applications – a review / A.Z. Hafeza, A. Solimana, K.A. El-Metwallya, I.M. Ismaila // Renewable and Sustainable Energy Reviews. – 2017. – V. 77. – P. 147–168.
- 17 Monthly profile analysis based on a two-axis solar tracker proposal for photovoltaic panels / F.M. Hoffmann, R.F. Molz, J.V. Kothe et al. // Renewable Energy. – 2018. – V. 115. – P. 750–759.
- 18 Система слежения за солнцем (трекер) модель HS-1500. Санкт-Петербург 2011. Сплит-системы.htm.
- 19 Bahrami A., Okoye C.O., Atikol U. Technical and economic assessment of fixed, single and dualaxis tracking PV panels in low latitude countries // Renewable Energy. – 2017. – V. 113. – P. 563–579.

20 A review of principle and suntracking methods for maximizing solar systems output / H. Mousazadeh, A. Keyhani, A. Javadi et al. // Renewable and Sustainable Energy Reviews. – 2009. – V. 13. – P. 1800–1818

ҚОСЫМША А

```
.include "m16def.inc" // Кіріспе файлын еңгіземіз
.def temp = r16 // регистрлерге ат береміз
.def razr1 = r17
.def razr2 = r18
.def razr3 = r19
.def razr4 = r20
.def sys = r21
.def try = r22
.def count = r23

.equ FREQ = 8000000 // процессор жиілігі

////////////////////// LCD уст
.equ HD44780_CLEAR =0x01
.equ HD44780_HOME =0x02
.equ HD44780_ENTRY_MODE =0x04
.equ HD44780_EM_SHIFT_CURSOR =0
.equ HD44780_EM_SHIFT_DISPLAY =1
.equ HD44780_EM_DECREMENT =0
.equ HD44780_EM_INCREMENT =2
.equ HD44780_DISPLAY_ONOFF =0x08
.equ HD44780_DISPLAY_OFF =0
.equ HD44780_DISPLAY_ON =4
.equ HD44780_CURSOR_OFF =0
.equ HD44780_CURSOR_ON =2
.equ HD44780_CURSOR_NOBLINK =0
.equ HD44780_CURSOR_BLINK =1
.equ HD44780_FUNCTION_SET =0x20
.equ HD44780_FONT5x7 =0
.equ HD44780_FONT5x10 =4
.equ HD44780_ONE_LINE =0
.equ HD44780_TWO_LINE =8
.equ HD44780_4_BIT =0
.equ HD44780_8_BIT =16
.equ LCD_PORT = PORTB // LCD порты
.equ LCD_DDR = DDRB // LCD порты
.equ LCD_PIN = PINB // LCD порты
.equ LCD_D4 = 4 // Аяқтардын қайда жалғанғаны
.equ LCD_D5 = 5
.equ LCD_D6 = 6
.equ LCD_D7 = 7
.equ LCD_RS = 0
```

```
.equ LCD_EN = 2
```

```
////////// LCD настройка соңы
```

```
.equ W1_DDR = DDRA // ds18b20 датчик портының регистрі
```

```
.equ W1_PORT = PORTA // ds18b20 датчик портының регистрі
```

```
.equ W1_PIN = PINA // ds18b20 датчик портының регистрі
```

```
.equ W1_BIT = 7 // Порт биті
```

```
.equ FreqSCL = 100000 // I2C жұмыс жиілігі 100 КГц
```

```
.equ FreqTWBR = ((FREQ/FreqSCL)-16)/2 // Регистр TWPS == 0
```

```
.dseg // ОЗУ
```

```
Trm: // 2 ұяшық температуранікі
```

```
.byte 2
```

```
OtoBr: // 6 ұяшық аралық есептеулер
```

```
.byte 6
```

```
TimeWatch: // 7 ұяшық сағаттан келген ақпарат үшін
```

```
.byte 7
```

```
DECSH:
```

```
.byte 1
```

```
.cseg // Программа сегменті
```

```
.org 0 // Үзу векторыReset
```

```
jmp Reset // Reset ке көшу
```

```
Reset: // Установка
```

```
ldi temp, high(RAMEND) //
```

```
out sph, temp
```

```
ldi temp, low(RAMEND)
```

```
out spl, temp
```

```
ldi temp, 0xff //
```

```
out DDRD, temp //
```

```
ldi temp, 0xFF //
```

```
out DDRB, temp
```

```
ldi temp, 0x0F //
```

```
out PORTA, temp
```

```
ldi temp, FreqTWBR //
```

```
out TWBR, temp
```

```
rcall LCD_Init //
```

```
ldi sys, 0b10000000
```

```
rcall TWI_Start
```

```
ldi r16, 0xD0
```

```

rcall TWI_TByte
ldi r16, 0x00
rcall TWI_TByte
rcall TWI_Start
ldi r16, 0xD1
rcall TWI_TByte
rcall TWI_RByteNack
mov r18, r16          sts TimeWatch+6, r18
rcall TWI_Stop
sbrs r18, 7
rjmp Proga_Pred
rcall TWI_Start
ldi r16, 0xD0
rcall TWI_TByte
ldi r16, 0x00
rcall TWI_TByte andi r18, 0b01111111
mov r16, r18
rcall TWI_TByte
rcall TWI_Stop

```

```

Proga_Pred:
rcall W1_Sbros
cpi r17, 0
breq Proga_Pred
rcall W1_Init_12bit

```

```

Proga:
rcall Proga_Otoibr_Time
rcall Proga_Thermometr
ldi try, 4
Proga0:
sbis PINA, 0
rjmp Menu_Nastr_Hoir
rcall Delay
dec try
brne Proga0
rjmp Proga

```

```

Menu_Nastr_Hoir:
rcall Pop_Button
ldi temp, HD44780_DISPLAY_ONOFF | HD44780_DISPLAY_ON |
HD44780_CURSOR_OFF | HD44780_CURSOR_BLINK
rcall LCD_Com
rcall Proga_Otoibr_Hoir

```

```
Menu_Nastr_Hoir0:  
sbis PINA, 0  
rjmp Menu_Nastr_Number  
sbis PINA, 1  
rjmp Menu_Nastr_Minute  
sbis PINA, 2  
rcall Menu_Nastr_Hoir_Izm  
rcall Proga_Otoibr_  
rcall Delay  
rjmp Menu_Nastr_Hoir0
```

```
Menu_Nastr_Hoir_Izm:  
rcall Pop_Button  
lds razr1, TimeWatch+2  
mov razr2, razr1  
andi razr2, 0b00001111  
andi razr1, 0b11110000  
swap razr1  
inc razr2  
cpi razr2, 10  
brne Menu_Nastr_Hoir_Izm0  
clr razr2  
inc razr1 Menu_Nastr_Hoir_Izm0:  
swap razr1  
add razr2, razr1  
cpi razr2, 0x24  
brne Menu_Nastr_Hoir_Izm_Vix  
clr razr2  
Menu_Nastr_Hoir_Izm_Vix:  
sts TimeWatch+2, razr2  
ret
```

```
Menu_Nastr_Minute:  
rcall Pop_Button  
rcall Proga_Otoibr_Minute  
Menu_Nastr_Minute0:  
sbis PINA, 0  
rjmp Menu_Nastr_Number  
sbis PINA, 1  
rjmp Menu_Nastr_Second  
sbis PINA, 2  
rcall Menu_Nastr_Minute_Izm  
rcall Proga_Otoibr_Minute  
rcall Delay
```

```
rjmp Menu_Nastr_Minute0
```

```
Menu_Nastr_Minute_Izm:  
rcall Pop_Button  
lds razr1, TimeWatch+1  
mov razr2, razr1  
andi razr2, 0b00001111  
andi razr1, 0b11110000  
swap razr1  
inc razr2  
cpi razr2, 10  
brne Menu_Nastr_Minute_Izm0  
clr razr2 //  
inc razr1  
Menu_Nastr_Minute_Izm0:  
swap razr1  
add razr2, razr1  
cpi razr2, 0x60 //  
brne Menu_Nastr_Minute_Izm_Vix  
clr razr2  
Menu_Nastr_Minute_Izm_Vix:  
sts TimeWatch+1, razr2  
ret
```

```
Menu_Nastr_Second:  
rcall Pop_Button  
rcall Proga_Otoibr_Second  
Menu_Nastr_Second0:  
sbis PINA, 0  
rjmp Menu_Nastr_Number  
sbis PINA, 1  
rjmp Menu_Nastr_Hoir  
sbis PINA, 2  
rcall Menu_Nastr_Second_Izm  
rcall Proga_Otoibr_Second  
rcall Delay  
rjmp Menu_Nastr_Second0
```

```
Menu_Nastr_Second_Izm:  
rcall Pop_Button  
lds razr1, TimeWatch  
mov razr2, razr1  
andi razr2, 0b00001111  
andi razr1, 0b11110000
```

```
swap razr1
inc razr2
cpi razr2, 10
brne Menu_Nastr_Second_Izm0
clr razr2
inc razr1
Menu_Nastr_Second_Izm0:
swap razr1
add razr2, razr1
cpi razr2, 0x60
brne Menu_Nastr_Second_Izm_Vix
clr razr2
Menu_Nastr_Second_Izm_Vix:
sts TimeWatch, razr2
ret
```

```
Menu_Nastr_Number:
rcall Pop_Button
rcall Proga_Otoibr_Number
Menu_Nastr_Number0:
sbis PINA, 0
rjmp Menu_Nastr_Ned
sbis PINA, 1
rjmp Menu_Nastr_Month
sbis PINA, 2
rcall Menu_Nastr_Number_Izm
rcall Proga_Otoibr_Number
rcall Delay
rjmp Menu_Nastr_Number0
```

```
Menu_Nastr_Number_Izm:
rcall Pop_Button
lds razr1, TimeWatch+4
mov razr2, razr1
andi razr2, 0b00001111
andi razr1, 0b11110000
swap razr1
inc razr2
cpi razr2, 10
brne Menu_Nastr_Number_Izm0
clr razr2
inc razr1
Menu_Nastr_Number_Izm0:
swap razr1
```

```
add razr2, razr1
cpi razr2, 0x32
brne Menu_Nastr_Number_Izm_Vix
ldi razr2, 0x01
Menu_Nastr_Number_Izm_Vix:
sts TimeWatch+4, razr2
ret
```

```
Menu_Nastr_Month:
rcall Pop_Button
rcall Proga_Otoibr_Month
Menu_Nastr_Month0:
sbis PINA, 0
rjmp Menu_Nastr_Ned
sbis PINA, 1
rjmp Menu_Nastr_Year
sbis PINA, 2
rcall Menu_Nastr_Month_Izm
rcall Proga_Otoibr_Month
rcall Delay
rjmp Menu_Nastr_Month0
```

```
Menu_Nastr_Month_Izm:
rcall Pop_Button
lds razr1, TimeWatch+5
mov razr2, razr1
andi razr2, 0b00001111
andi razr1, 0b11110000
swap razr1
inc razr2 //
cpi razr2, 10
brne Menu_Nastr_Month_Izm0
clr razr2
inc razr1
Menu_Nastr_Month_Izm0:
swap razr1 //
add razr2, razr1
cpi razr2, 0x13
brne Menu_Nastr_Month_Izm_Vix
ldi razr2, 0x01
Menu_Nastr_Month_Izm_Vix:
sts TimeWatch+5, razr2
rcall Pop_Button
ret
```

Menu_Nastr_Year:
rcall Pop_Button
rcall Proga_Otoabr_Year
Menu_Nastr_Year0:
sbis PINA, 0
rjmp Menu_Nastr_Ned
sbis PINA, 1
rjmp Menu_Nastr_Number
sbis PINA, 2
rcall Menu_Nastr_Year_Izm
rcall Proga_Otoabr_Year
rcall Delay
rjmp Menu_Nastr_Year0

Menu_Nastr_Year_Izm:
rcall Pop_Button
lds razr1, TimeWatch+6
mov razr2, razr1
andi razr2, 0b00001111
andi razr1, 0b11110000
swap razr1
inc razr2 //
cpi razr2, 10
brne Menu_Nastr_Year_Izm_Vix
clr razr2
inc razr1
cpi razr1, 10
brne Menu_Nastr_Year_Izm_Vix
clr razr1
Menu_Nastr_Year_Izm_Vix:
swap razr1
add razr2, razr1
sts TimeWatch+6, razr2
rcall Pop_Button
ret

Menu_Nastr_Ned:
rcall Pop_Button
rcall Proga_Otoabr_Ned
Menu_Nastr_Ned0:
sbis PINA, 0
rjmp Menu_Vixod_Osn
sbis PINA, 2
rcall Menu_Nastr_Ned_Izm

```
rcall Proga_Otoibr_Ned  
rcall Delay  
rjmp Menu_Nastr_Ned0
```

```
Menu_Nastr_Ned_Izm:  
rcall Pop_Button  
lds temp, TimeWatch+3  
inc temp  
cpi temp, 8 //  
brne Menu_Nastr_Ned_Izm_Vix  
ldi temp, 1  
Menu_Nastr_Ned_Izm_Vix:  
sts TimeWatch+3, temp  
ret
```

```
Menu_Vixod_Osn:  
rcall Pop_Button  
ldi temp, HD44780_DISPLAY_ONOFF | HD44780_DISPLAY_ON |  
HD44780_CURSOR_OFF | HD44780_CURSOR_NOBLINK  
rcall LCD_Com  
rcall Time_Record  
rjmp Proga
```

```
Pop_Button:  
sbis PINA, 0  
rjmp Pop_Button  
sbis PINA, 1  
rjmp Pop_Button  
sbis PINA, 2  
rjmp Pop_Button  
ret
```

```
Proga_Otoibr_Time:  
rcall Time_Read  
Proga_Otoibr_Time0:  
rcall Proga_Otoibr_Hoir  
rcall Proga_Otoibr_Minute  
rcall Proga_Otoibr_Second  
rcall Proga_Otoibr_Number  
rcall Proga_Otoibr_Month  
rcall Proga_Otoibr_Year  
rcall Proga_Otoibr_Ned  
ret
```

```
Proga_Otoabr_Hoir:
ldi razr3, 0x30
ldi temp, 128+0x00
rcall LCD_Com
lds temp, TimeWatch+2
andi temp, 0b11110000
swap temp /
add temp, razr3
rcall LCD_Dan
lds temp, TimeWatch+2
andi temp, 0b00001111
add temp, razr3
rcall LCD_Dan
ldi temp, ':' //
rcall LCD_Dan
ldi temp, 128+0x00
rcall LCD_Com
ret
```

```
Proga_Otoabr_Minute:
ldi razr3, 0x30
ldi temp, 128+0x03
rcall LCD_Com
lds temp, TimeWatch+1
andi temp, 0b11110000
swap temp
add temp, razr3
rcall LCD_Dan
lds temp, TimeWatch+1
andi temp, 0b00001111
add temp, razr3
rcall LCD_Dan
ldi temp, ':'
rcall LCD_Dan
ldi temp, 128+0x03
rcall LCD_Com
ret
```

```
Proga_Otoabr_Second:
ldi razr3, 0x30
ldi temp, 128+0x06
rcall LCD_Com
lds temp, TimeWatch
andi temp, 0b11110000
```

```
swap temp
add temp, razr3
rcall LCD_Dan
lds temp, TimeWatch
andi temp, 0b00001111
add temp, razr3
rcall LCD_Dan
ldi temp, 128+0x06
rcall LCD_Com
ret
```

```
Proga_Otoabr_Number:
ldi razr3, 0x30
ldi temp, 128+0x40
rcall LCD_Com
lds temp, TimeWatch+4
andi temp, 0b11110000
swap temp
add temp, razr3
rcall LCD_Dan
lds temp, TimeWatch+4
andi temp, 0b00001111
add temp, razr3
rcall LCD_Dan
ldi temp, '.'
rcall LCD_Dan
ldi temp, 128+0x40
rcall LCD_Com
ret
```

```
Proga_Otoabr_Month:
ldi razr3, 0x30
ldi temp, 128+0x43
rcall LCD_Com
lds temp, TimeWatch+5
andi temp, 0b11110000
swap temp
add temp, razr3
rcall LCD_Dan
lds temp, TimeWatch+5
```

```
andi temp, 0b00001111
add temp, razr3
rcall LCD_Dan
ldi temp, '.'
rcall LCD_Dan
ldi temp, 128+0x43
rcall LCD_Com
ret
```

```
Proga_Otoabr_Year:
ldi razr3, 0x30
ldi temp, 128+0x46
rcall LCD_Com
lds temp, TimeWatch+6
andi temp, 0b11110000
swap temp
add temp, razr3
rcall LCD_Dan
lds temp, TimeWatch+6
andi temp, 0b00001111
add temp, razr3
rcall LCD_Dan
ldi temp, 128+0x46
rcall LCD_Com
ret
```

```
Proga_Otoabr_Ned:
lds temp, TimeWatch+3 //
ldi ZH, High(CH_Ned*2)
ldi ZL, Low(CH_Ned*2)
Proga_Otoabr_Ned0:
lpm razr2, Z+ //
lpm razr3, Z+ //
lpm razr4, Z+ //
dec temp //
brne Proga_Otoabr_Ned0 //
ldi temp, 128+0x09 //
rcall LCD_Com
mov temp, razr2
rcall LCD_Dan
mov temp, razr3
rcall LCD_Dan
mov temp, razr4
rcall LCD_Dan
```

```
brne Proga_Otoabr_Ned0 //
ldi temp, 128+0x09 //
rcall LCD_Com
ret
```

```
Proga_Thermometr:
rcall W1_Sbros //
cpi r17, 0
breq Proga_Thermometr //
rcall W1_ReadMem //
```

```
ldi temp, 0x30 //
sts Otoabr, temp
sts Otoabr+1, temp
sts Otoabr+2, temp
lds temp, Trm
sbrs temp, 7
rjmp Proga_Plusovay
lds temp, Trm
andi temp, 0b00001111
ldi razr1, 0x0F
sub razr1, temp
lds temp, Trm+1
andi temp, 0b11110000
ldi razr2, 0xF0
sub razr2, temp
```

```
rcall Rasch_Decimal //
rcall Ubrat_NULL //
ldi temp, '-' //
sts Otoabr, temp
lds razr1, Trm+1
andi razr1, 0b00001111
ldi temp, 0x0F
sub temp, razr1
rcall Temper_Des_Conv
sts Otoabr+3, razr1
sts Otoabr+4, razr2
rcall Otoabr_Visible
Proga_Thermometr0:
rcall W1_Sbros
cpi r17, 0
breq Proga_Thermometr0
rcall W1_ConvTemp
```

ret

Proga_Plusovay:

```
lds razr1, Trm
lds razr2, Trm+1
andi razr2, 0b11110000
rcall Rasch_Decimal
rcall Ubrat_NULL
lds temp, Trm+1
andi temp, 0b00001111
rcall Temper_Des_Conv
sts Otoabr+3, razr1
sts Otoabr+4, razr2
rcall Otoabr_Visible
rjmp Proga_Thermometr0
```

Rasch_Decimal:

```
swap razr1
swap razr2
add razr1, razr2
cpi razr1, 0
breq NULL_Temp
```

Rasch_Decimal0:

```
ldi XH, High(Otoabr+2)
ldi XL, Low(Otoabr+2)
ld temp, X
inc temp
st X, temp
cpi temp, 0x3A
brne Rasch_Decimal1
ldi temp, 0x30
st X, temp
ld temp, -X
inc temp
st X, temp
cpi temp, 0x3A
brne Rasch_Decimal1
ldi temp, 0x30
st X, temp
ld temp, -X
inc temp
st X, temp
Rasch_Decimal1:
```

```
dec razr1
brne Rasch_Decimal0
ret
```

```
Rasch_Otoibr:
lds temp, Otoibr+2
rcall Rasch_Otoibr_Podp
sts Otoibr+2, razr1 //
lds temp, Otoibr+1
rcall Rasch_Otoibr_Podp
sts Otoibr+1, razr1
lds temp, Otoibr+0
rcall Rasch_Otoibr_Podp
sts Otoibr+0, razr1
ret
```

```
Rasch_Otoibr_Podp: //
ldi ZH, High(CH_Cif*2)
ldi ZL, Low(CH_Cif*2)
inc temp //
Rasch_Otoibr_Podp0:
lpm razr1, Z+
dec temp
brne Rasch_Otoibr_Podp0
ret //
```

```
NULL_Temp:
ret
```

```
Ubrat_NULL:
lds temp, Otoibr
cpi temp, 0x30 //
brne Ubrat_NULL_Vix //
ldi temp, '' //
sts Otoibr, temp
lds temp, Otoibr+1
cpi temp, 0x30
brne Ubrat_NULL_Vix
ldi temp, ''
sts Otoibr+1, temp
Ubrat_NULL_Vix:
ret
```

```
Otoibr_Visible:
```

```

ldi temp, 128+0x49
rcall LCD_Com
lds temp, Otober
rcall LCD_Dan
lds temp, Otober+1
rcall LCD_Dan
lds temp, Otober+2
rcall LCD_Dan
ldi temp, '.'
rcall LCD_Dan
lds temp, Otober+3
rcall LCD_Dan
lds temp, Otober+4
rcall LCD_Dan
ldi temp, 0xDF
rcall LCD_Dan //
ret

```

```

Temper_Des_Conv:
inc temp //
ldi ZH, High(CH_Temp*2)
ldi ZL, Low(CH_Temp*2)
Temper_Des_Conv0:
lpm razr1, Z+
lpm razr2, Z+
dec temp
brne Temper_Des_Conv0
ret

```

```

Time_Record:
rcall TWI_Start
ldi r16, 0xD0
rcall TWI_TByte
ldi r16, 0x00
rcall TWI_TByte
lds r16, TimeWatch
rcall TWI_TByte
lds r16, TimeWatch+1
rcall TWI_TByte
lds r16, TimeWatch+2
rcall TWI_TByte
lds r16, TimeWatch+3
rcall TWI_TByte
lds r16, TimeWatch+4

```

```
rcall TWI_TByte
lds r16, TimeWatch+5
rcall TWI_TByte
lds r16, TimeWatch+6
rcall TWI_TByte
rcall TWI_Stop
ret
```

```
Time_Read:
rcall TWI_Start
ldi r16, 0xD0
rcall TWI_TByte
ldi r16, 0x00
rcall TWI_TByte
rcall TWI_Start
ldi r16, 0xD1
rcall TWI_TByte
rcall TWI_RByteAck
sts TimeWatch, r16
rcall TWI_RByteAck
sts TimeWatch+1, r16
cpi r16, $50
breq second_set
cpi r16, $55
breq second_set
lds r29, DECSH
cpi r29, $0
brne DATA
```

```
endm:
rcall TWI_RByteAck
sts TimeWatch+2, r16
rcall TWI_RByteAck
sts TimeWatch+3, r16
rcall TWI_RByteAck
sts TimeWatch+4, r16
rcall TWI_RByteAck
sts TimeWatch+5, r16
rcall TWI_RByteNack
sts TimeWatch+6, r16
rcall TWI_Stop
ret
```

```
second_set:
```

```

lds r25, TimeWatch
cpi r25, $59
breq datakoi
rjmp endm
datakoi:
ldi r29, $4
sts DECSH, r29
ldi r24, $08
lds r25, TimeWatch+2
cpi r25, $12
breq seven
cpi r25, $13
breq seven
cpi r25, $14
breq seven
rjmp DATA
seven:
ldi r29, $7
sts DECSH, r29
rjmp DATA
DATA:
rcall delay
lsr r24
out PORTD, r24
cpi r24, 0
breq zpsn
rjmp endm
zpsn:
lds r29, DECSH
dec r29
sts DECSH, r29
ldi r24, $08
out PORTD, r24
rjmp endm

```

```

//////////////////// TWI
TWI_Start:

```

```
ldi r17, (1<<TWINT) | (1<<TWEN) | (1<<TWIE) | (1<<TWSTA)
out TWCR, r17
rcall TWI_Byte
ret
```

```
TWI_Stop:
ldi r17, (1<<TWINT) | (1<<TWEN) | (1<<TWIE) | (1<<TWSTO)
out TWCR, r17
ret
```

```
TWI_TByte:
out TWDR, r16
ldi r17, (1<<TWINT) | (1<<TWEN) | (1<<TWIE) | (1<<TWEA)
out TWCR, r17
rcall TWI_Byte
ret
```

```
TWI_RByteAck:
ldi r17, (1<<TWINT) | (1<<TWEN) | (1<<TWIE) | (1<<TWEA)
out TWCR, r17
rcall TWI_Byte
in r16, TWDR
ret
```

```
TWI_RByteNack:
ldi r17, (1<<TWINT) | (1<<TWEN) | (1<<TWIE) | (0<<TWEA)
out TWCR, r17
rcall TWI_Byte
in r16, TWDR
ret
```

```
TWI_Byte:
in r17, TWCR
sbrs r17, TWINT
rjmp TWI_Byte
ret
```

//////////////////// TWI

//////////////////// 1-Wire

```
W1_Sbros: //
lds r16, W1_BIT
sbi W1_DDR, W1_BIT
cbi W1_PORT, W1_BIT
rcall W1_DelayH
```

```
cbi W1_DDR, W1_BIT
rcall W1_DelayI
sbis W1_PIN, W1_BIT
ldi r17, 1
sbic W1_PIN, W1_BIT //
ldi r17, 0
rcall W1_DelayJ
ret //
```

```
W1_ReadMem:
ldi r16, 0xCC
rcall ds_byte_wr
ldi r16, 0xBE
rcall ds_byte_wr
rcall ds_byte_rd
sts Trm+1, r16
rcall ds_byte_rd
sts Trm, r16
ret
```

```
W1_ConvTemp:
ldi r16, 0xCC
rcall ds_byte_wr
ldi r16, 0x44
rcall ds_byte_wr
ret
```

```
W1_Init_12bit:
ldi r16, 0xCC
rcall ds_byte_wr
ldi r16, 0x4E
rcall ds_byte_wr
ldi r16, 0xFF //
rcall ds_byte_wr
ldi r16, 0xFF //
rcall ds_byte_wr //
ldi r16, 0x7F //
rcall ds_byte_wr
ret
```

```
ds_byte_rd:
ldi r17, 8 //
clr r16 //
ds_byte_rd_0:
```

```
sbi W1_DDR, W1_BIT
cbi W1_PORT, W1_BIT
rcall W1_DelayA
cbi W1_DDR, W1_BIT
rcall W1_DelayE //
sbis W1_PIN, W1_BIT
clc //
sbic W1_PIN, W1_BIT
sec //
ror r16
rcall W1_DelayF
dec r17 //
brne ds_byte_rd_0
ret
```

```
ds_byte_wr:
ldi r17, 8 //
ds_byte_wr0:
sbi W1_DDR, W1_BIT
cbi W1_PORT, W1_BIT
sbrc r16, 0
rjmp ds_byte_write_1
rjmp ds_byte_write_0
ds_byte_wr1:
lsr r16
dec r17 //
brne ds_byte_wr0
ret
```

```
ds_byte_write_0:
rcall W1_DelayC
cbi W1_DDR, W1_BIT
rcall W1_DelayD
rjmp ds_byte_wr1
```

```
ds_byte_write_1:
rcall W1_DelayA
cbi W1_DDR, W1_BIT
rcall W1_DelayB
rjmp ds_byte_wr1
```

```
W1_DelayA:
ldi XH, high(FREQ/1000000)
ldi XL, low(FREQ/1000000)
```

```
rcall W1_Delay
ret
W1_DelayB:
ldi XH, high(FREQ/65000)
ldi XL, low(FREQ/65000)
rcall W1_Delay
ret
W1_DelayC:
ldi XH, high(FREQ/68000)
ldi XL, low(FREQ/68000)
rcall W1_Delay
ret
W1_DelayD:
ldi XH, high(FREQ/500000)
ldi XL, low(FREQ/500000)
rcall W1_Delay
ret
W1_DelayE:
ldi XH, high(FREQ/600000)
ldi XL, low(FREQ/600000)
rcall W1_Delay
ret
W1_DelayF:
ldi XH, high(FREQ/75000)
ldi XL, low(FREQ/75000)
rcall W1_Delay
ret
W1_DelayH:
ldi XH, high(FREQ/8332)
ldi XL, low(FREQ/8332)
rcall W1_Delay
ret
W1_DelayI: //
ldi XH, high(FREQ/58000)
ldi XL, low(FREQ/58000)
rcall W1_Delay
ret
W1_DelayJ: //
ldi XH, high(FREQ/9756)
ldi XL, low(FREQ/9756)
rcall W1_Delay
ret
W1_Delay: //
sbiw XH:XL, 1 //
```

```

brne W1_Delay //
ret //
//////////

////////// LCD ішкі програма
LCD_Init:
sbi LCD_DDR, LCD_D4
sbi LCD_DDR, LCD_D5
sbi LCD_DDR, LCD_D6
sbi LCD_DDR, LCD_D7
sbi LCD_DDR, LCD_RS
sbi LCD_DDR, LCD_EN
cbi LCD_PORT, LCD_RS
cbi LCD_PORT, LCD_EN
ldi r16, 100
rcall WaitMilliseconds
ldi r17, 3
InitLoop:
ldi r16, 0x03
rcall LCD_WriteNibble
ldi r16, 5
rcall WaitMilliseconds
dec r17
brne InitLoop
ldi r16, 0x02
rcall LCD_WriteNibble
ldi r16, 1
rcall WaitMilliseconds
ldi r16, HD44780_FUNCTION_SET | HD44780_FONT5x7 |
HD44780_TWO_LINE | HD44780_4_BIT
rcall LCD_Com
ldi r16, HD44780_DISPLAY_ONOFF | HD44780_DISPLAY_OFF
rcall LCD_Com
ldi r16, HD44780_CLEAR
rcall LCD_Com
ldi r16, HD44780_ENTRY_MODE | HD44780_EM_SHIFT_CURSOR |
HD44780_EM_INCREMENT
rcall LCD_Com
ldi r16, HD44780_DISPLAY_ONOFF | HD44780_DISPLAY_ON |
HD44780_CURSOR_OFF | HD44780_CURSOR_NOBLINK
rcall LCD_Com
ret //

;-----
LCD_Dan: //

```

```

sbi LCD_PORT, LCD_RS //
push r16
swap r16
rcall LCD_WriteNibble
pop r16
rcall LCD_WriteNibble //
clr XH //
ldi XL, LOW(FREQ/80000)
rcall Wait4xCycles //
ret

```

```

;-----

```

```

LCD_Com: //
cbi LCD_PORT, LCD_RS
push r16
swap r16
rcall LCD_WriteNibble
pop r16 //
rcall LCD_WriteNibble //
ldi r16, 2
rcall WaitMilliseconds //
ret

```

```

;-----

```

```

LCD_WriteNibble: //
sbi LCD_PORT, LCD_EN // EN +5
sbrs r16, 0
cbi LCD_PORT, LCD_D4
sbrc r16, 0
sbi LCD_PORT, LCD_D4
sbrs r16, 1
cbi LCD_PORT, LCD_D5
sbrc r16, 1
sbi LCD_PORT, LCD_D5
sbrs r16, 2
cbi LCD_PORT, LCD_D6
sbrc r16, 2
sbi LCD_PORT, LCD_D6
sbrs r16, 3
cbi LCD_PORT, LCD_D7
sbrc r16, 3
sbi LCD_PORT, LCD_D7
cbi LCD_PORT, LCD_EN
ret

```

```

;-----

```

```

WaitMilliseconds:

```

```

    push r16 //
WaitMsLoop:
    ldi  XH,HIGH(FREQ/17777)
    ldi  XL,LOW(FREQ/17777) //
    rcall Wait4xCycles //
    ldi  XH,HIGH(FREQ/17777)
    ldi  XL,LOW(FREQ/17777)
    rcall Wait4xCycles
    dec  r16
    brne WaitMsLoop
    pop  r16 //
    ret
;-----
Wait4xCycles: //
    sbiw  XH:XL, 1 //
    brne  Wait4xCycles //
    ret //
////////////////////////////////////

Delay:
    ldi  razr1, 255
    ldi  razr2, 255
    ldi  razr3, 5
Pdelay:
    dec  razr1
    brne Pdelay
    dec  razr2
    brne Pdelay
    dec  razr3
    brne Pdelay
    ret

CH_Cif:
    .db 0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39

CH_Temp: //
    .db 0x30,0x30, 0x30,0x36, 0x31,0x33, 0x31,0x39 // 00,06,13,19
    .db 0x32,0x35, 0x33,0x31, 0x33,0x38, 0x34,0x34 // 25,31,38,44
    .db 0x35,0x30, 0x35,0x36, 0x36,0x33, 0x36,0x39 // 50,56,63,69
    .db 0x37,0x35, 0x38,0x31, 0x38,0x38, 0x39,0x34 // 75,81,88,94

CH_Ned: //
    .db 'M','o','n', 'T','u','e', 'W','e','d', 'T','h','u'
    .db 'F','r','i', 'S','a','t', 'S','u','n',

```