**“AZƏRBAYCAN HAVA YOLLARI”**

**QAPALI SƏHMDAR CƏMİYYƏTİ**

**MİLLİ AVİASİYA AKADEMİYASI**



**Fakültə: “Fizika texnologiya”**

**Kafedra: “Aerokosmik cihazlar”**

**İxtisas: 050614 – “Aviasiya texnikası avadanlıqlarının istismarı mühəndisliyi**

**İxtisaslaşma: “Uçuş aparatlarının elektrik, radio və cihaz avadanlığı (avionik)”**

**BURAXILIŞ İŞİ**

**Mövzu:“ BORT QEYDİYYAT QURĞUSUNDAN UÇUŞ İNFORMASİYASINI OXUYAN CİHAZIN İŞLƏNMƏSİ.”**

|  |  |  |
| --- | --- | --- |
| **Tələbə:** |  | **Əliyev Rəşad Şəvkətbəy o.** |
|  |  | **(imza, tarix, S.A.A.)** |
| **Rəhbər:** |  | **Çıraqov Nəriman Əflatun o.** |
|  |  | **(imza, tarix, elmi dərəcə, elmi adı, S.A.A.)** |
| **Kafedra müdiri:** |  | **f.r.e.n. , dos. İsgəndərov İslam Əsəd o.** |
|  |  | **(imza, tarix, elmi dərəcə, elmi adı, S.A.A.)** |

**MƏSLƏHƏTÇİLƏR**

**NORMA NƏZARƏT: \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(imza, tarix) (s.a.a., elmi dərəcə və ad)**

**RƏYÇİ: \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(imza, tarix) (s.a.a., elmi dərəcə və ad)**

**BAKI-2021**

**Anotasiya**

Buraxılış işində Mi-35 tipli helikopterlərin sistemlərinin iş qabiliyyətinə, eləcə də uçuş heyətinin peşəkakarlığının qiymətləndirilməsinə uçuşlararası nəzarəti həyata keçirmək üçün uçuş informasiyasının “BUR-SL-1” qeydiyyat qurğusundan köçürülməsini təmin edən mikrokontroller qurğusunun sxemi və proqram təminatı işlənmiş, və həmçinin etalon verilənlərlə simulyator proqramında sınaqdan keçirilimişdir.

**Аннотация**

В выпускной работе разраблтаны схема и программное обеспечение микроконтроллерного устройства, обеспечивающего передачу полетной информации с регистрационного устройства «БУР-СЛ-1» для осуществления межполетного контроля работоспособности систем вертолета Ми-35, а также оценки состояния профессионализмп летного экипажа, a также тестирован на симуляторе учитывая эталонные дпнные.

**Annotation**

In the graduation work, the circuit and software of the microcontroller device were developed, which ensures the transfer of flight information from the registration device "BUR-SL-1" for interflight monitoring of the performance of the Mi-35 helicopter systems, as well as assessing the state of professionalism of the flight crew, and also tested on a simulator taking into account reference data.

**MÜNDƏRİCAT**

**“Bort qeydiyyat qurğusundan uçuş informasiyasını oxuyan cihazın işlənməsi”**

**Fəsil 1.Mövcud qurğular və onlarin müqayisəli təhlili.**

§1.1.Bort qeydiyyat qurğusundan uçuş informasiyasını oxuyan cihazın işlənməsi.

§1.2 Qurğunun əsas texniki xarakteristikaları:

§1.3. BUR-SL-1 informasiya kadrı.

§1.4 Qeydiyyat parametrlərinin formatı cədvəli

**Fəsil 2: Konkret BUR-SL-1 qurğusu,və onun quruluşu texniki xarakteristikaları**.

§2.1. BUR-SL-1 seriya 7" tipli BQQ-lərin məlumat interfeysləri

§2.2 "USI-TM" tipli ÇİC-in xüsusiyyətləri

§2.3. Yeni məlumat köçürmə cihazının sxemotexnikası

§2.4.Yeni məlumat köçürmə cihazının proqram təminatı

**Fəsil 3. İnformasiyanın oxunması üçün vasitələr,o cümlədən arduino platforması və onun serial potu haqqinda məlumat.**

§3.1 Arduino mega 2560 platforması haqqında ümumi məlumat.

§3.2. Mikrokontrollerin xüsusiyyətləri.

§3.3 Arduino mega 2560 platformasında BUR-SL1 tipli Bort qeydiyyat qurğusundan uçuş informasiyasının oxunması üçün proqram təminatı(C++)

**Fəsil 4.Alqoritim və proqramların təsviri və simulyatorda test edilməsi.**

§4.1.Arduinodan istifadə edərək mətni və simvolları LCD ekranda göstərmək.

§4.2 Proqram kodları.

§4.3 Arduino Mega 2560 mikrokontrollerin və LCD ekranin köməyi ilə uçuşlarin oxunması.

**Giriş**

Məlum olduğu kimi, uçuş aparatlarında quraşdırılmış avadanlıqların funksionallığının monitorinqi və uçuş qəzalarının səbəblərinin müəyyən edilməsi üçün Bort Qeydiyyat Qurğularında (BQQ) toplanmış parametrik uçuş məlumatlarının işlənməsi tələb olunur. Bu məqsədlə həmin məlumatlar müxtəlif texniki vasitələrin köməyi ilə Yerüstü Emal Sisteminə (YES) köçürülür və orada emal olunurlar. Son zamanlar BQQ-lərin texniki vəziyyətinin yoxlanılması və oradakı məlumatların köçürülməsi üçün universal interfeysə malik əl cihazlarından (Çoxtəyinatlı İnterfeys Cihazı - ÇİC) geniş istifadə olunur. ÇİC-in BQQ-yə qoşulması üçün uçuş aparatlarının nəzarət hücrələrində xüsusi siqnal konnektorları quraşdırılır.

İstismarda olan helikopterlərin bəzi tipləri "BUR-SL-1 seriya 7" BQQ-ləri ilə təchiz olunmuşdur. Bu BQQ-lərdən məlumatın köçürülməsi üçün "УСИ-ТМ" tipli ÇİC-lərdən istifadə olunur. Bu ÇİC-lər müxtəlif məlumat interfeyslərini dəstəkləmək nöqteyi-nəzərindən unifikasiya olunmuş modullar əsasında yığılır: fərqli fiziki interfeyslər üçün fərqli modullardan, fərqli mübadilə protokolları üçün fərqli daxili proqram təminatından istifadə olunur.

Lakin istifadəçının iradəsindən asılı olmayan səbəblərdən sıradan çıxmış ÇİC-lərin təmiri və əlavələrinin alınmasında böyük izafi xərclər tələb olunur, uzunmüddətli vaxt itkiləri yaranır. Bu isə öz növbəsində helikopterlər parkının intensiv istismarı dövrlərində onlara operativ nəzarət reqlamentinin pozulması ilə nəticələnir. Bu problemlərin aradan qaldırılması üçün alternativ ÇİC-nin hazırlanması zərurəti yaranmışdır.

Buraxılış işi "BUR-SL-1 seriya 7" BQQ-lərindən parametrik uçüş məlumatlarının köçürülməsi üçün nəzərdə tutulmuş tək interfeysli məlumat köçürmə cihazının layihələndirilməsinə həsr olunmuşdur.

**Fəsil I. Mövcud qurğular və onlarin müqayisəli təhlili.**

**§1.1 Bort qeydiyyat qurğusundan uçuş informasiyasını oxuyan cihazın işlənməsi.**

İşin məqsədi Mi-35 tipli helikopterlərin “BUR-SL-1” qeydiyyat qurğusundan məlumatların köçürülməsini təmin edən elektron qurğunun hazırlanmasıdır.

Mi-35 tipli helikopterlərin sistemlərinin iş qabiliyyətinə, eləcə də uçuş heyətinin peşəkakarlığının qiymətləndirilməsinə uçuşlararası nəzarəti həyata keçirmək üçün uçuş informasiyasının “BUR-SL-1” qeydiyyat qurğusundan köçürülməsini təmin edən elektron qurğunun hazırlanmasını nəzərdə tutur.

Elektron qurğu MI-35 helikopterinin **“BUR-SL-1 “** qurğusundan uçuş məlumatlarını flash tipli informasiya daşıyıcısina yazmaq üçün nəzərdə tutulur və aşağıdakı funksional imkanlara malik olmalıdır:

* Mi-35 tipli helikopterdə olan xüsusi hücrədəki konnektoruna qoşulmanı təmin etməli;
* “BUR-SL-1” bort qeydiyyat qurgusunda olan parametrik məlumatların uçuşdan sonra kompüterə köçürülməsini təmin etməli;
* uçuşların(reyslərin) siyahısını vizuallaşdırmaq, seçim etmək və seçilmiş uçuşa aid məlumatları daşıyıcıya köçürmək üçün indikator və müvafiq düyməciklər ilə təmin edilməli;
* məlumatlar daşıyıcıda bortun nömrəsi, reysin nömrəsi və məlumatın oxunduğu cari zamana uyğun identifikasiya olunnaqla müvafiq qovluqlarda və fayllarda saxlanılmalır;
* cari zamam avtomatik təyin edilməli və lazım gəldikdə istifadəçi tərəfindən düzəliş edilməsini nəzərdə tutmalıdır;
* informasiyanın daşıyıcıdan standart vasitələrlə (USB girişi ilə)kompyuterə köçürüıməsi təmin edilməlidir;
* elektron qurğu ilə helikopter arasında informasiya mübadiləsi müvafiq interfeyslə təmin edilməlidir;
* köçürülən məlumatların formatı uçuş məlumatlarının emalı üzrə mövcud sistem ilə uyğun olmalıdır.

BUR-SL-1 (seriya 7) qurğusu uçuş məlumatlarının toplanması və qeydiyyatı sistemi uçuş qəzalarının araşdırılması və təyyarələrin operativ idarəetmə problemlərinin həlli üçün lazımı məlumatları toplamaq və yığmaq üçün nəzərdə tutulmuşdur. Yüngül və ultra yüngül təyyarələrə quraşdırıla bilər.

BUR-SL-1-ə aşağıdakılar daxildir:

• bort qeyd cihazı BUR-SL-1;

• qoşqu BUR – kompüter;

• BUR-SL-1 proqram təminatı;

• əməliyyat və proqram sənədləri.

BUR-SL-1 üç iş rejimini təmin edir: “İş-1”, “İş-2” və “Xidmət”.

"İş-1" rejimində məlumat toplanır, çevrilir və buraxılışı ilə bərk cisimli saxlama cihazına qeyd olunur

"İş-2" rejimində məlumatlar toplanır və bərk cisimli saxlama cihazına yazılmadan çevrilir.

Qeydiyyat (qeyd) rejimi proqram təminatı ilə müəyyən edilir və ya hər güc açıldıqda, ya da birdəfəlik PK16 əmrinin iştirakı ilə başlaya bilər.

Bundan əlavə, "Əməliyyat" rejimində, kompüter BUR - kompüter vasitəsi ilə qoşulduqda, məlumat RS-422 portu vasitəsilə verilir:

• kompüter siqnalında ГОСТ 18977-79 и РТМ 1495-75- uyğun olaraq analog siqnalların girişlərində BUR-SL-1 tərəfindən alınan parametrlərin kod dəyərlərinə çevrilmiş göstəriciləri, birdəfəlik əmrləri və ardıcıl kodları.;

• analog siqnal sensorlarının kalibrlənməsi.

"Xidmət" rejimində, BUR-SL-1-ə qoşulmuş fərdi kompüter istifadə edərək, aşağıdakılar həyata keçirilir:

• iş proqramı üçün BUR-SL-1 məlumatlarının yüklənməsi;

• yığılmış məlumatların qorunan bərk cisim qurğusundan köçürülməsi.

BUR-SL-1-in “İşləmə” rejimindən “Baxım” rejiminə keçməsi BUR - kompüter əlaqə qutusuna quraşdırılmış Əməliyyat - XİDMƏT çeviricisi ilə həyata keçirilir.  
BUR-kompüter BUR-SL-1-dən çıxardarkən, sonuncusu avtomatik olaraq “İşləmə” rejiminə keçir.

BUR-SL-1-in görünüşü Şəkil 1-də göstərilmişdir.



1 - qapaq, 2 - korpus, 3 - işıqəksetdirici zolaqlar,

4 - plomb yüei, 5 – xarici elektrik birləşdiricisi

Şəkil 1.1 BUR-SL-1 xarici görünüşü

BUR-SL-1 Uçuş aparatına - UA ya amortizasiya qurğusu olmadan quraşdırılır.

BUR-SL-1 parametrlərinin gətirilmiş sistematik xətası



kimi tapılır. Gd-sensorun gətirilmiş xətası

Gацп- analoq-rəqəm çeviricisinin gətirilmiş xətasıdır.

Diskret siqnalların xarakteristikaları:

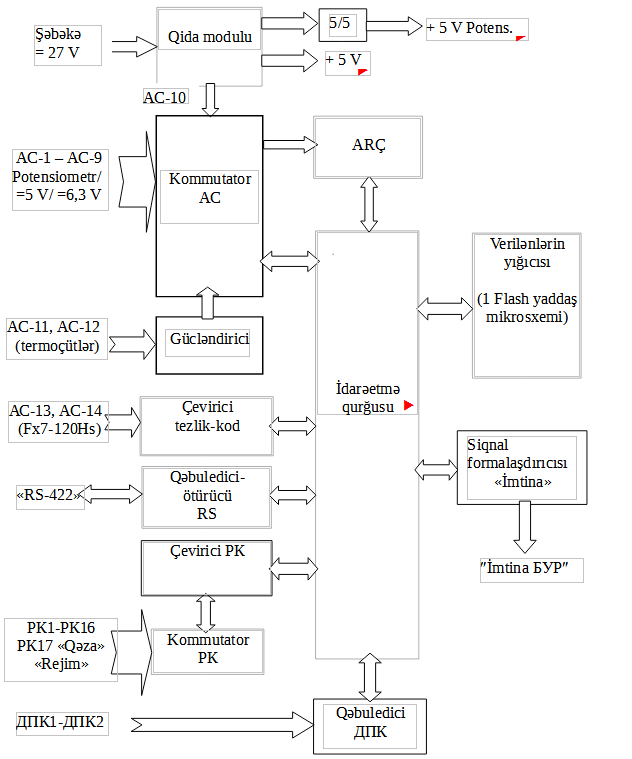
Cədvəl 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Diskret siqnalın tipi | Əlaməti | Vəziyyəti | Kodlaşması |
| I | Yox | Dövrə açıq | 0 |
| Var | Korpusa qpanma | 1 |
| II | Yox | - Korpusa qpanma  -Dövrə açıq - 3 V gərginlik | 0 |
| Var | 7 - 33 V gərginlik | 1 |

**§1.2 Qurğunun əsas texniki xarakteristikaları:**

|  |  |
| --- | --- |
| Kütləsi | 6 kq |
| Qabaritləri | 243х153х96 mm |
| Qida gərginliyi | 27 V(18−31) V |
| Güc | 5 Vt |
| Proqramlaşdırılan girişlərin sayı  ·        analoq ·        diskret | 12 11 |
| Proqramlaşdırılmayan girişlərin sayı:  ·   analoq 0−33 V ·   tezlik siqnalları 7-120 Hers ·   birdəfəlik komandalar (gərginlik 7 - 33 V) ·   ardıcıl kodlar | 1 3 6 2 |
| Müraciət tezliyi (proqramlaşdırıla bilər) АС/РК | 1, 2, 4, 8, 16, 32 Hers |
| Xidməti verilənlərin qeydiyyatı:  ·         cari zaman ·         qoşulma nömrəsi ·         uçuş aparatının nömrəsi | (saat, dəqiqə, saniyə) 0 - 4095 0 - 99999 |
| Yazının saxlanma müddəti: |  |
| ·         son 32 saat ·         son 16 saat  ·         son 8 saat | 256 söz/san 1 söz = 12 bit 512 söz/san 1 söz = 12 bit 1024 söz/san 1 söz = 12 bit |
| İnformasiyanın təsir altında saxlanması |  |
| ·         zərbə yükü 1000 ədəd. ·         2260 kq statik yükün paylanması  ·         temperatur + 1100°С ·         dəniz suyunda qalma  ·         kerosində qalma | 5 ms 5 dəqiqə 30 dəqiqə, 100% səth 30 sutka 2 sutka, hər biri |
| İnformasiyanın oxunması İnformasiyanın emalı | «Topaz-M» və digər vasitələr |
| İstismar şərtləri: |  |
| ·         temperatur ·         nəmlik | 55°С - +70°С 98% при +35°С |
| İmtina müddəti | 20 000 ч |

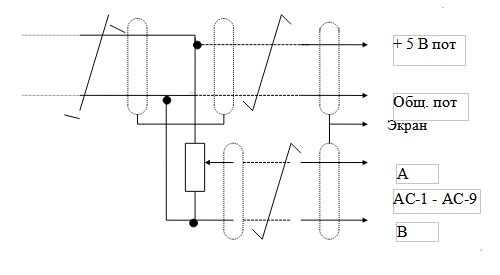
**BUR-SL-1 struktur sxemi**



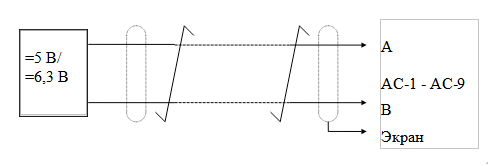
**Şək.1.2 BUR-SL-1 struktur sxemi.**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| KontKontKont | Dövrə | |  | KontKont | Dövrə | |  | Kont | Dövrə | | |
| 1 | А | АС-1 |  | 25 | А | АС-12 Тп2 |  | 51 | +27 В б/с | | |
| 2 | В |  | 26 | В |  | 52 | - 27 В б/с | | |
| 3 | А | АС-2 |  | 28 | РК1 | |  | 53 | В | ДПК1 | |
| 4 | В |  | 29 | РК2 | |  | 54 | А |
| 5 | А | АС-3 |  | 30 | РК3 | |  | 55 | В | ДПК2 | |
| 6 | В |  | 31 | РК4 | |  | 56 | А |
| 7 | А | АС-4 |  | 32 | РК5 | |  | 57 | А | RXD | |
| 8 | В |  | 33 | РК6 | |  | 58 | В |
| 9 | А | АС-5 |  | 34 | РК7 | |  | 59 | А | TXD | |
| 10 | В |  | 35 | РК8 | |  | 60 | В |
| 11 | А | АС-6 |  | 36 | РК9 | |  | 61 | DOĞRULUQ | | |
| 12 | В |  | 37 | РК10 | |  |  |  | |  |
| 13 | А | АС-7 |  | 38 | РК11 | |  |  |  | |  |
| 14 | В |  | 39 | РК12 | |  |  |  | |  |
| 15 | А | АС-8 |  | 40 | РК13 | |  |  |  | |  |
| 16 | В |  | 41 | РК14 | |  |  |  | |  |
| 17 | А | АС-9 |  | 42 | РК15 | |  |  |  | |  |
| 18 | В |  | 43 | РК16/Пвкл | |  |  |  | |  |
| 19 | + 5 V POT | |  | 44 | РК17 “Qəza” | |  |  |  | |  |
| 20 | ÜMUMİ.POT | |  | 45 | -27 V корп НУО | |  |  |  | |  |
| 21 | EKRAN | |  | 46 | А | АС-13 F1 |  |  |  | |  |
| 22 | Ekran Тп | |  | 47 | В |  |  |  | |  |
| 23 | А | АС-11 Тп1 |  | 48 | А | АС-14 F2 |  |  |  | |  |
| 24 | В |  | 49 | В |  |  |  | |  |
| 27 | ÜMUMİ.АН | |  | 50 | XİDMƏT REJİMİ | |  |  |  | |  |

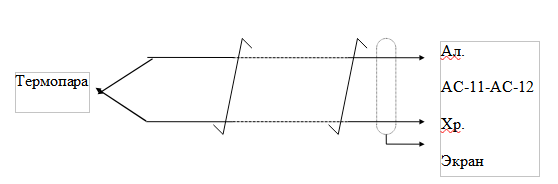
**Şək.1.3 BUR-SL-1 siqnalların paylanması**



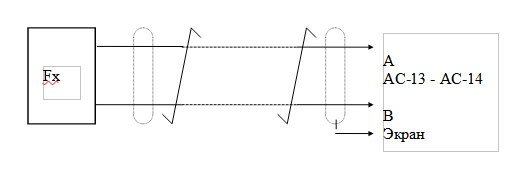
**Şək.1.4 Potensiometrik tipli vericilərin qoşulma sxemi .**



**Şək.1.5 Sabit cərəyan siqnal mənbəyinin qoşulma sxemi .**



**Şək. 1.6 Termocütün qoşulma sxemi**



**Şək. 1.7 Tezlik vericisinin qoşulma sxemi .**

**§1.3 BUR-SL-1 informasiya kadrı.**

Alınan və çevrilən məlumatların BUR-SL-1 sistemində qeydiyyatı üçün ardıcıl kod istifadə olunur. Konvertasiya edilmiş məlumat on iki kiçik biti informasiya daşıyıcısı olan on altı bitli sözlərə çevrilir. Sözlər kiçik bitlər əvvədə gəlməklə bir-birini izləyir.

BUR-SL-1 məlumat kadrı 256 sözdən (mövqedən) ibarətdir.

Kadr, hər biri 64 bitlik 12 sözdən ibarət olan 4 alt kadra bölünür.

Hər alt kadrın ilk sözü sinxronizasiya sözdür və alt kadrı müəyyənləşdirməyə xidmət edir.

Alt kadrların sinxronizasiya sözləri səkkizlik və ikilik kodlarda aşağıdakı qiymətlərə malikdir (kiçik bitlər solda):

* birinci alt kadr - 1107 – 001 001 000 111;
* ikinci alt kadr - 2670 - 010 110 111 000;
* üçüncü alt kadr - 5107 – 101 001 000 111;
* dördüncü alt kadr - 6670 – 110 110 111 000.

Birinci alt kadrln on yeddinci sözü, dövri olaraq 0-dan 4095-ə qədər dəyişən çərçivənin sıra nömrəsini qeyd etmək üçün nəzərdə tutulmuşdur.

Bütün alt kadrların otuz üçüncü sözləri tanınma verilənlərini qeyd etmək üçündür.

İlk üç alt kadrın otuz yeddinci sözləri Cədvəl 4-ə uyğun olaraq vaxt qeyd etmək üçün (saniyələr - birinci alt kadrda, dəqiqələr - ikinci alt kadrda, saatlar - üçüncü alt kadrda) nəzərdə tutulmuşdur.

Dördüncü alt kadrın otuz yeddinci sözü BUR-SL-1 status sözünü qeyd etmək üçün nəzərdə tutulmuşdur.

Parametrləri yazmaq üçün nəzərdə tutulmuş blokdakı sözlərin sayı 243-dür.  
Hər analog parametr üçün 12 və ya sözün böyük 10 biti ayrılır.

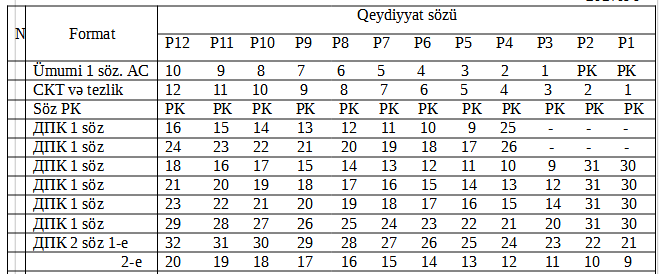
RK qeyd etmək üçün, 10-bit AC və ya tam 12-bitli bir söz olan hər hansı bir sözdən iki alt sərbəst bit ayrılır. Bu vəziyyətdə, hər bir RK və ya biti qeyd etmək üçün bir bit ayrılır.

AC girişlərində və PC sözləri şəklində alınan məlumatların qeydiyyatı hər bir konkret obyekt üçün qarşılıqlı əlaqə protokollarına və əməliyyat proqramına uyğun olaraq həyata keçirilir.

Hər hansı bir analog parametr və ya diskret komanda ilə kadrda tutulan mövqelərin sayı, 2 Hz-dən yuxarı tezliklərlə alınan parametrlər kadrında qeydin vahidliyini müşahidə edərkən qəbul / qeyd etmə tezliyi ilə mütənasibdir.

**§1.4 Qeydiyyat parametrlərinin formatı cədvəli**

Cədvəl 1.1



**Fəsil II. Konkret BUR-SL-1 qurğusu,və onun quruluşu texniki xarakteristikaları.**

**§2.1 "BUR-SL-1 seriya 7" tipli BQQ-lərin məlumat interfeysləri**

"BUR-SL-1 seriya 7" BQQ-lərində yığılmış uçuş məlumatlarının köçürülməsi üçün "RS232" tipli ardıcıl ötürmə interfeysindən istifadə olunur. [1,2] "RS232" fiziki interfeys olmaqla küylərə böyük davamlılığı təmin etmək üçün ±12V-luq siqnal səviyyələrindən istifadə edir. Bu BQQ-lərdə məlumat axını üçün 57600 Bod sürəti istifadə olunur. Məlumat mübadiləsi sorğu-cavab protokolu əsasında təşkil olunur və əsas komandaların siyahısı cədvəl 1-də verilmişdir. Bu komandalar BQQ-nin yaddaş massivinin quruluşu və uçuş məlumatlarının orada yerləşdirilməsi konsepsiyasına uyğun təşkil olunmuşlar. Bu konsepsiyanın əsas məqsədi uçuş məlumatlarının yaddaş massivinə mümkün qədər sadə sxemotexniki yollarla cəld yazılıb-oxunmasıdır. Nəticədə bəzi yaddaş blokları bu massivin ünvan klasterində əlçatmaz olmuşdur və komandalardan istifadə edərkən bu məqamlar nəzərə alınmalıdır.

"BUR-SL-1 seriya 7" BQQ-lərində cari zaman uçuş məlumatları axınına nəzarət etmək üçün "RS422" interfeysi də nəzərdə tutulmuşdur. Bu interfeysin köməyi ilə bortdakı ölçü-nəzarət sisteminin işçi vəziyyəti yoxlanıla bilər.

Hər iki interfeys üzrə uçuş məlumatlarının ötürülmə ardıcıllığı BQQ-də qəbul olunmuş məlumat kadrına uyğundur və xüsusi sinxronlaşdırıcı baytlar vasitəsi ilə tənzimlənir.

**§2.2 "USI-TM" tipli ÇİC-in xüsusiyyətləri**

"USI-TM" müxtəlif tip BQQ-lərə qoşulmaq üçün nəzərdə tutulmuş daşınan cihazdır (şək. 1). Elektron hissəsi müasir mikroelektron ciplərdən istifadə edilməklə hazırlanır. Membranlı klaviatura və 3(sətir) x 8 simvolluq LED displey ilə təchiz olunur. Avtonom yaddaş modulu kiçik voltajlı "flash" çipləri əsasında hazırlanır və cihaza optik interfeys vasitəsi ilə qoşulur. BQQ-lərə qoşulmaq üçün müxtəlif kabel-adapterlərdən istifadə olunur. Onlarda qısa qapanma, izafi (80 V-a qədər) potensial fərqindən qorunma elementləri, elektrik küyünü azaltmaq üçün filtrlər quraşdırıla bilər. Cihaz uçuş aparatının 27V-luq sabit cərəyan elektrik şəbəkəsindən qidalandırılır.

"USI-TM" cihazının "BUR-SL-1 seriya 7" BQQ-lərinə qoşulmasını təmin edən kabel-adapterin özəlliyi onda MAX221 çipindən istifadə edilməsidir. Bu, avtomatik sönmə/oyanma funksiyaları və yüksək məlumat ötürmə sürəti olan tək +5V qida gərginlikli "RS-232" (V.28) interfeys ötürücü-qəbuledicisidir. Daxilində +5V qida gərginliyini interfeysin tələb etdiyi +12/-12V gərginliklərinə çevirən inverter sxemi vardır. MAX221 çipinin ölçüləri və güc sərfiyyatı çox kiçikdir, ±15 кV-luq elektrostatik boşalmalara dözümlü ötürücü çıxışları 3-vəziyyətlidir və kabelin BQQ-yə qoşulub-ayrılmasına avtomatik reaksiya verə bilirlər. [3]

"USI-TM" cihazı işə salındıqda test rejiminə keçir. 15 saniyəlik müddətdə cihazın konfiqurasiyası, elektron modulların vəziyyəti və BQQ-nün məlumat mübadiləsinə hazırlığı təyin olunur. Müsbət nəticələr alındıqda işçi rejim qərarlaşır və cihazın displeyində uyğun məlumatlar təqdim olunur.

**§2.3 Yeni məlumat köçürmə cihazının sxemotexnikası**

Yeni məlumat köçürmə cihazı geniş yayılmış STM32F103 mikrokontrolleri və MAX3232 interfeys çipi əsasında layihələndirilmişdir [4,5] və aşağıdakı imkanlara malikdır (şək. 2):

- BQQ-də qeyd olunmuş uçuş məlumatlarının siyahısının çıxarılması,

- BQQ-də qeyd olunmuş uçuş məlumatlarının seçilməsi,

- seçilmiş uçuş məlumatlarının yaddaş elementinə köçürülməsi,

- yaddaş elementindəki uçuş məlumatlarının emalı üçün Yerüstü Emal Sisteminə ötürülməsi.

Mikrokontroller uçuş məlumatlarının qəbulunu və DB9 tipli konnektorun gövdəsində yerləşdirilmiş SDCard yaddaş elementinə qeyd edilməsini təmin edir. Yaddaş elementindəki parametrik uçüş məlumatları kart oxuyucu vasitəsi ilə Yerüstü Emal Sisteminə köçürülür və emal olunur. BQQ ilə mübadilə protokoluna və məlumat axınına nəzarət üçün 2(sətir) x 16 simvolluq "LCD" modulundan istifadə olunmuşdur. İdarə funksiyaları 5-düyməli klaviatura vasitəsilə həyata keçirilir. Cari vaxtı təyin etmək üçün taymer modulu, bortdan gələn 27V qida gərginliyini cihazın işçi 5V və 3V gərginliklərinə çevirmək üçün impuls gərginlik çevirici modulu istifadə olunmuşdur. Mikrokontrollerin digər elektron modullarla əlaqəsi standart "SPI", "I2C", "GPIO", "UART" portları üzərindən təşkil olunmuşdur.

MAX3232 çipi kabeldə deyil, cihazın daxilində yerləşdirilmiş və "RS232" interfeysində yaranan gərginlik sıçrayışlarından diod körpüsü vasitəsi ilə qorunmuşdur. Diodlar məlumat xətlərini çipin ±12V-luq "V+", "V-" çıxışları ilə əlaqələndirir. Diod körpüsündən istifadə MAX3232 çipinin dözümlülüyünü nəzərə çarpacaq dərəcədə artırmış, kabelin borta qoşulması və qida gərginliyinin verilməsi anlarında məlumat xətlərində yaranan zərərli sıçrayışlar nəticəsində sıradan çıxma ehtimalnı xeyli azaltmışdır.

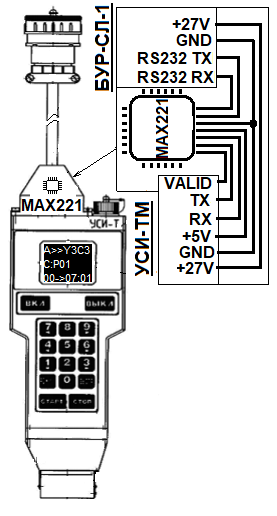
**§2.4 Yeni məlumat köçürmə cihazının proqram təminatı**

Yeni məlumat köçürmə cihazının iş alqoritmi şəkil 3 və 4-də verilmişdir. Cihazın proqram təminatı strukturlu proqramlaşdırma metodikasına uyğun olaraq məsələlər toplusu kimi təşkil olunmuşdur və hər bir məsələ ayrıca funksiya kimi tərtib olunmuşdur. Funksiyalar növbə ilə ardıcıl olaraq işə buraxılır.

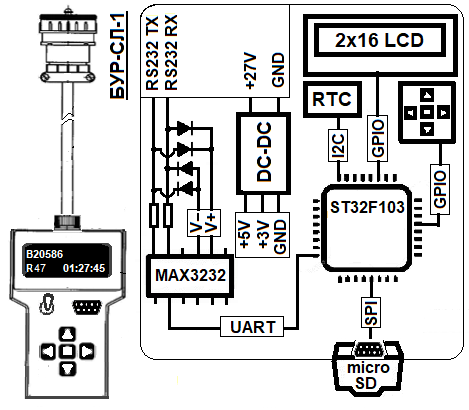
Cihaz qida mənbəyinə qoşulan kimi mikrokontroller və digər elektron modullar standart kitabxana altproqramlarının köməyi ilə standart konfiqurasiya vəziyyətinə sazlanırlar. Sonra BQQ-yə "Başlangıc Vəziyyət" komandası göndərilir və onun məlumat mübadiləsinə hazırlığı yoxlanılır. Əgər BQQ hazırdısa bortun nömrəsi oxunur və cari təqvim məlumatları ilə birlikdə cihazın displeyinə çıxarılır.

BQQ-nin yaddaş blokları dövri "FIFO" *(first in first out)* sxemi üzrə qoşulduqlarından ilk növbədə sonuncu uçuş məlumatlarının qeyd olunduğu blokun nömrəsini təyin etmək zərurəti yaranır. Daha sonra sonuncu yaddaş blokundan başlayaraq bütün bloklarda qeyd olunmuş uçuşların nömrəsi, müddəti, həcmi barədə məlumat cihazın displeyinə çıxarılır və köçürülməsi üçün uçuşun seçilməsi təklif olunur.

Son mərhələdə seçilmiş uçuş məlumatları "DB9" tipli konnektorun gövdəsində yerləşdirilmiş SDcard yaddaşına fayl şəklində köçürülür. Proqram təminatı bir neçə uçuş məlumatını birgə seçərək köçürməyə imkan verir. Bu halda hər bir uçuşa aid məlumatlar ayrı-ayrı fayllara qeyd olunur.

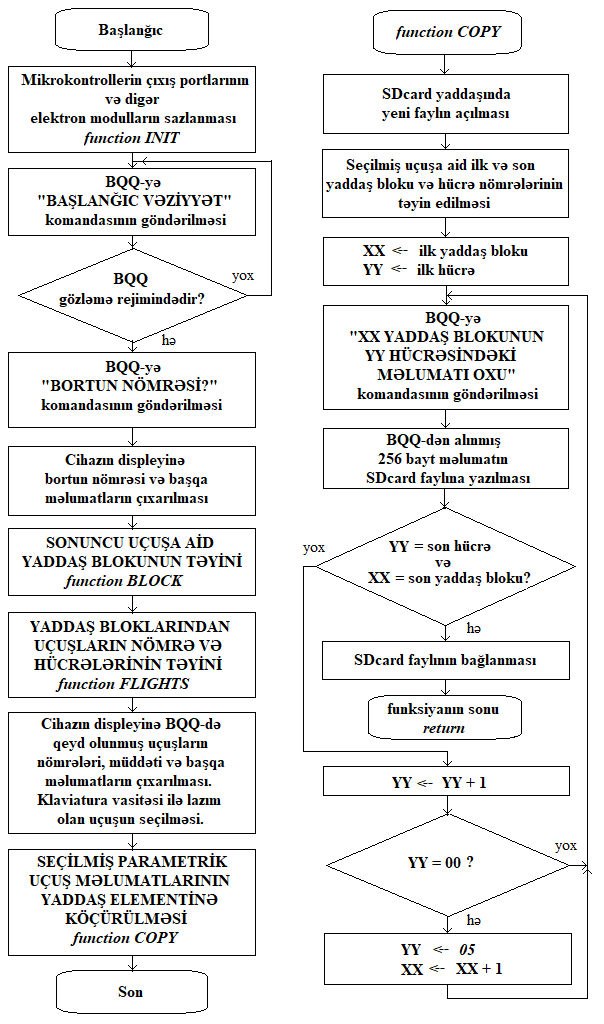


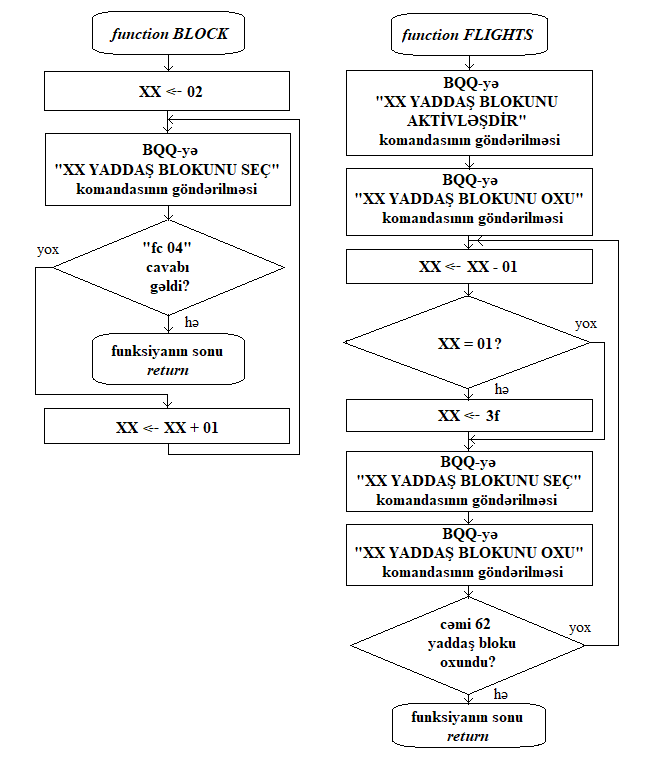
**Şək.2.1 "BUR-SL-1 seriya 7" tipli Bort Qeydiyyat Qurğusuna qoşula bilən "USI-TM" tipli Çoxtəyinatlı İnterfeys Cihazı və onun "RS232" kabeli.**



**Şək.2.2 "BUR-SL-1 seriya 7" tipli Bort Qeydiyyat Qurğusuna qoşula bilən yeni Məlumat Köçürmə Cihazı və onun struktur sxemi.**

**Şək. 2.3 Məlumat Köçürmə Cihazının iş alqoritminin ümumiləşdirilmiş blok-sxemi.**



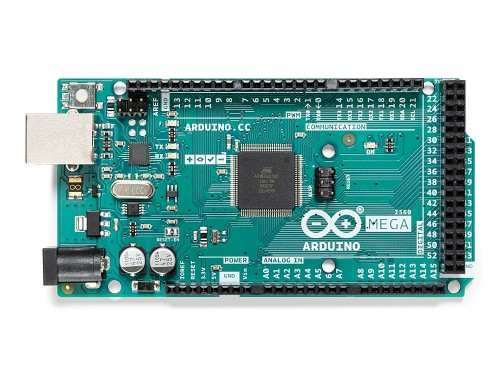
**Şəkil.2.4 Məlumat Köçürmə Cihazının iş alqoritminin ümumiləşdirilmiş blok-sxemi(ardı).**

Cədvəl 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | **KOMANDA** | **BQQ-dən CAVAB** | **FUNKSİYASI** |
| (komandaya uyğun) | (istənilən komanda) | 80 8e | Sorğu Komandası (BQQ komandanı qəbul etdi) İcraya Hazır |
| 8080 | İCRA ET | (komandaya uyğun) | Komandaya Cavab (BQQ komandanı cavablayır) İcra Olunur |
| 80 | BİTİR | 8e | Komandaya Son (BQQ növbəti komandaya hazırdı) İcra Olundu |
| növbəti komandalar şablona uyğun olaraq "İCRA ET" (8080) komandasına cavab alırlar | | | |
| **80 01 XX 00 00 01 00 CRC** | **XX yaddaş blokunu seç** | **f8 08** yaxud **fc 04** | **"fc 04"** cavabı XX (02 ~ 3f) nömrəli yaddaş blokunda sonuncu uçuşun olduğunu təsdiqləyir. |
| **80 01 XX 93 00 01 00 CRC** | **XX yaddaş blokunu aktivləşdir** | **04 fc** | XX yaddaş blokunu aktivləşdirir (əvvəlki komandanın eynidir, yalnız sonuncu uçuşun qeyd olunduğu bloka tətbiq olunur) |
| **80 01 XX 00 01 00 01 CRC** | **XX yaddaş blokunu oxu**  (01 və 00 nömrələrini buraxmaqla, hər dəfə cədvəldə əvvəlki komandalarla birgə tam 62 bloka tətbiq olunur) | 4 dəfə (64 bayt **??** və **CRC**) | XX yaddaş blokundakı 256 hücrənin hansı uçuşa mənsub olduqlarını təyin edir. (XX yaddaş blokundakı 4\*64=256 baytın hər biri hansısa uçuşun nömrəsi ilə işarələnir, "**ff** "ilə işarələnmiş baytlar boş hücrələri işarə edir, 4saniyəlik vaxtı və 256 ölçü baytını əhatə edən hücrənin ünvanı kimi istifadə olunur, məsələn, bütün bloklarda cəmi 304 bayt **"de"** ilə işarələnibsə (30-cu uçuş), bu 304\*4=1216san, 304\*256=77824bayt uçuş məlumatı deməkdir) |
| **80 01 XX 00 YY 00 01 CRC** | **XX yaddaş blokunun YY hücrəsindəki məlumatı oxu** | 4 dəfə (64 bayt **??** və **CRC**) | YY hücrəsindən 256 bayt uçuş məlumatını oxuyur.  bu komandanı digərlərindən fərqləndirmək üçün hücrələr (YY) "**05~ff**"intervalında nömrələnirlər. bu səbəbdən hər bir XX blokunun 5 ədəd son bloku ünvanlana bilmədiyi üçün **"ff"** ilə işarələnir. hər bir uçuşa aid məlumatı oxumaq üçün bu komandanı həmin uçuşun nömrəsi qeyd olunmuş sıraca əvvəlki blokdan (XX) və hücrədən (YY) başlamaqla həmin uçuşa aid hər bir hücrə üçün göndərmək lazımdır. məsələn, 30-cu (**"de"**) uçuş üçün bu komanda 304 dəfə göndərilməlidir |
| **80 01 01 40 0a 0c 00 CRC** | **Bortun nömrəsi?** | **XX XX ff ff ff ff ff ff ff ff ff ff CRC** | bortun identifikasiya nömrəsini oxuyur (**9b 4e** - BORT 20123) |
| **80 0f 01 00 00 01 00 CRC** | **BAŞLANĞIC VƏZİYYƏT** | **80 8e** | BQQ-ni ilkin vəziyyətinə gətirir (şablondan fərqli olaraq **"İCRA ET"** komandasından istifadə etmir) **"BİTİR"** komandasından sonra **"8e"** və dövri təkrarlanan **"80"** cavabları alır (BQQ gözləmə rejimindədir) |

**Fəsil III. İnformasiyanın oxunması üçün vasitələr,o cümlədən arduino platforması və onun serial portu haqqinda məlumat.**

**§3.1 Arduino mega 2560 platfornası haqqında ümumi məlumat.**

Yuxarıda Arduino Meganın ön və arxa tərəfdən görünüşləridir. Kartda Atmega 2560 mikrokontrollerindən istifadə olunmuşdur. Bu kart ilə istənilən layihəni hazırlamaq və çipi (MCU) proqramladıqdan sonra çıxarmadan yerindəcə istifadə etmək olar. Bunun üçün kart üzərindəki çıxışlardan istifadə edilir. Belə ki, bu çıxışlar birbaşa çipin pinlərinə qoşulmuş vəziyyətdədir.

Kartın 54 giriş-çıxış pini var, bunlardan 14-nü PWM çıxışı olaraq istifadə etmək olar, 16-sı analog giriş, 4-ü UART (serial port), qalanlarını digital (rəqəmsal) giriş-çıxış kimi istifadə etmək olar. Bundan başqa kartda 16 MHz-lik kristal, USB əlaqə, adaptor girişi, ICSP çıxışı və reset düyməsi vardır.

§3.2 Mikrokontrollerin xüsusiyyətləri.

Mikrokontroller -ATmega2560

Çalışma gərginliyi -5 Volt

Qidalandırma gərginliyi -(7-12) Volt (məsləhətli)

Qidalandırma gərginliyi -(6-20) Volt (Limit)

Digital İ/O - 54 (14-ü PWM çıxışı)

Analog giriş pinləri - 16

İ/O pin axınları -40 mAmper

3,3 Volt pin axını - 50 mAmper

Fləş yaddaş - 256 KB (8 kb bootloader istifadə edir.)

SRAM - 8 KB

EEPROM - 4 KB

Saat tezliyi -16 MHz

Arduino Mega 2560-ı enerji ilə təmin etməyin iki yoldu var. Birinci USB ilə qidalanma, ikincisi adaptor ilə qidalanma. USB qidalanmada artıq yükləmədən qoruyucu açar var. Yəni məsələn Arduinoya servo mator qoşulsa onu idarə etməyə başlayan kimi USB əlaqəsi kəsilir, və deməli enerji əlaqəsidə kəsilmiş olur. Kartın normal işləməsi üçün sabit 5 Volt tələb olunur. Əgər adaptordan istifadə edilsə 7-12 Volt ilə qidalandırılır. 12 voltdan yuxarı enerji verildikdə tənzimləyici element çox qızar və kartın tez sıradan çıxmasına səbəb olar. Əksinə 7 Voltdan az enerji verilsə tənzimləyici tam olaraq 5 Volt verə bilməyəcək və dövrədə tarazlıq pozulacaq.

Giriş-çıxış pinləri, 54-ə qədərdir. Bunların hər birinipin Mode(), digitalWrite(), və digitalRead() funksiyalarından istifadə edərək giriş-çıxış kimi istifadə etmək olar. Pinlərin hər biri 5 voltla işləyir və maksimum 40 mA cərəyan axınına dözür. Tranzistorlar və relelərdən istifadə etməklə istənilən gücdə cihazı idarə idarə etmək olar.

Bundan başqa bu pinlərin bəzilərinin xüsusi funksiyalarıda var.

Ardıcıl: 0 (RX) və 1 (TX); Serial 1: 19 (RX) ve 18 (TX); Serial 2: 17 (RX) ve 16 (TX); Serial 3: 15 (RX) ve 14 (TX). Ardıcıl data almaq (RX) və göndərmək (TX) üçün istifadə olunur. 0 and 1 pinləri eyni vaxtda  Atmega16U2 USB-to-TTL ardıcıl mikrosxeminin müvafiq pinlərinə bağlıdır.

Xarici kəsmə: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). Bu pinlər xarici kəsmələr üçün istifadə olunur. attachInterrupt()funksiyasından istifadə etməklə hansı interruptın ne şəkildə istifadə olunacağı bildirilir.

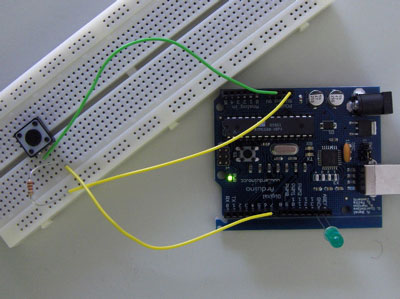
PWM: 0 to 13. 8-bit PWM çıxış verir. analogWrite()funksiyası istifadə olunur.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).Bu pinlerSPIitabxanası istifadə olunaraq SPI əlaqə yaratmağa imkan verir. SPI pinləri eyni zamanda ICSP headerdan da istifadə olunur.

LED: 13.13 nömrəli pində LED vardır. Çıxış HIGH edildikdə LED yanır, LOW edildikdə sönür.

TWI: 20 (SDA) and 21 (SCL).WIRE kitabxanasından yararlanaraq TWI əlaqə qurula bilər

Arduino Mega 2560, ATmega2560 bazalı mikrokontrolör lövhəsidir. 54 rəqəmsal giriş və çıxış pininə (bunlardan 15-i PWM çıxışları kimi istifadə edilə bilər), 16 analoq girişə, 4 UART-a (hardware serial portları), 16 MHz kristal osilatora, USB bağlantısına, elektrik prizinə, ICSP başlığına, və sıfırlama düyməsini basın. Mikro nəzarətçini dəstəkləmək üçün lazım olan hər şeyi ehtiva edir; başlamaq üçün sadəcə USB kabeli ilə bir kompüterə qoşun və ya AC-to-DC adapteri və ya batareyası ilə gücləndirin.

Düymə.Düymə basdığınız zaman bir dövrədə iki nöqtəni birləşdirən bir komponentdir. Düyməni basdığınız zaman nümunə bir LED yandırır. Arduino lövhəsinə üç naqil bağlayırıq. Birincisi, düymənin bir ayağından bir çəkmə müqavimətindən (burada 2,2 KOhms) 5 voltlu bir təchizatı keçər. İkincisi, düymənin uyğun ayağından yerə gedir. Üçüncüsü, düymənin vəziyyətini oxuyan rəqəmsal bir giriş / çıxış pininə (burada pin 7) qoşulur.Düymə açıq olduqda (basılmamış) düymənin iki ayağı arasında heç bir əlaqə yoxdur, buna görə də pin 5 volt (çəkmə müqaviməti vasitəsilə) bağlanır və YÜKSƏK oxuyuruq. Düymə bağlandıqda (basıldıqda), iki ayağı arasında bir əlaqə qurur, pimi yerə bağlayır, belə ki, DÜŞÜK oxuyuruq. (Pin hələ də 5 volta bağlıdır, lakin aralarındakı müqavimət, sancağın yerə "dahayaxın"olmasıdeməkdir.)**Şəkil.3.1**

\* Basic Digital Read

\* ------------------

\*

\* turns on and off a light emitting diode(LED) connected to digital

\* pin 13, when pressing a pushbutton attached to pin 7. It illustrates the

\* concept of Active-Low, which consists in connecting buttons using a

\* 1K to 10K pull-up resistor.

\*

\* Created 1 December 2005

\* copyleft 2005 DojoDave <http://www.0j0.org>

\* http://arduino.berlios.de

\*

\*/

int ledPin = 13; // choose the pin for the LED

int inPin = 7; // choose the input pin (for a pushbutton)

int val = 0; // variable for reading the pin status

void setup() {

pinMode(ledPin, OUTPUT); // declare LED as output

pinMode(inPin, INPUT); // declare pushbutton as input

}

void loop(){

val = digitalRead(inPin); // read input value

if (val == HIGH) { // check if the input is HIGH (button released)

digitalWrite(ledPin, LOW); // turn LED OFF

} else {

digitalWrite(ledPin, HIGH); // turn LED ON

}

}

**§3.3.Arduino mega 2560 platfornasında BUR-SL1 tipli Bort qeydiyyat qurğusundan uçuş informasiyasının oxunması üçün proqram təminatı(C++)**

#include "LiquidCrystal.h"

#include "SdFat.h"

SdFat sd; SdFile dataFile; SdFile listFile;

const int ledPin = 13;

unsigned long tt;

byte w8[8]{ 0x80, 0x01, 0x02, 0x00, 0x00, 0x01, 0x00, 0xfc };

byte w2[]{ 0x80, 0x80 }; byte w1[]{ 0x80 };

byte r2[2]; byte r13[13]; byte r3[3]; byte r6[6];

byte r1[1]; byte r65[65]; byte r256[256];

int totalN; int totalT; int ind = 0; byte numarr[64];

char maze[64][16] = {}; int lcd\_start = 0;

LiquidCrystal lcd(43, 39, 29, 27, 25, 23);

int DownButtonState = 0; int UpButtonState = 0; int SaveButtonState = 0;

int BortNumber = 0; byte Tbayt, Cbayt; int sanS; int mudt[64];

byte nom[64];byte ktlq[64]; byte fb[]{ 0x80, 0x01, 0x02, 0x00, 0x00, 0x01, 0x00, 0xfc };

byte LastRecord = 0xff; boolean first = false; unsigned long t;

void w8\_r2()

{

Serial1.write(&w8[0], 8);

while (Serial1.available() < 2) {};

Serial1.readBytes(r2, 2);

}

void w1\_r1()

{

while (Serial1.read()>=0) {};

Serial1.write(w1[0]);

while (Serial1.available() < 1) {};

r1[0] = Serial1.read();

}

void w2\_r2()

{

while (Serial1.read()>=0) {};

Serial1.write(&w2[0], 2);

while (Serial1.available() < 2) {};

Serial1.readBytes(r2, 2);

}

void w2\_r13()

{

while (Serial1.read()>=0) {};

Serial1.write(&w2[0], 2);

while (Serial1.available() < 13) {};

Serial.print("w2\_r13 after millis="); Serial.print(millis(), DEC); Serial.print("available=");Serial.print(Serial1.available(), DEC); Serial.println("\n");

Serial1.readBytes(r13, 13);

}

void w2\_r65(int i)

{

while (Serial1.read()>=0) {};

Serial1.write(&w2[0], 2);

while (Serial1.available() < 65) {};

for (int j = 0; j < 65; j++) {

r65[j] = Serial1.read();

if (j < 64) r256[j + 64 \* i] = r65[j];

}

}

void GetLastFlight(byte ui)

{

mudt[ui] = sanS;

if (LastRecord == 0xff) LastRecord = ui;

char hexstrbt2[3]; char hexstrbt4[3];

sprintf(hexstrbt2, "%.2x", ktlq[ui]);

sprintf(hexstrbt4, "%.2x", (nom[ui] + 5));

numarr[ind] = ui;

if (ind == lcd\_start)

sprintf(maze[ind], "R=%.2i t=%.2i:%.2i:%.2i", ui, mudt[ui] \* 4 / 3600, ((mudt[ui] \* 4) % 3600) / 60, ((mudt[ui] \* 4) % 3600) % 60);

else

sprintf(maze[ind], "r=%.2i t=%.2i:%.2i:%.2i", ui, mudt[ui] \* 4 / 3600, ((mudt[ui] \* 4) % 3600) / 60, ((mudt[ui] \* 4) % 3600) % 60);

lcd.setCursor(0, 1);

lcd.print(maze[ind]);

ind++;

sanS = 0;

totalN++;

totalT += mudt[ui];

}

void GetFlight()

{

for (int j = 250; j >= 0; j--)

{

if (r256[j] == 0xff) continue;

if (Tbayt != r256[j] && !first)

{

Tbayt = r256[j];

first = true;

continue;

}

if (Tbayt == r256[j]) sanS++;

else

{

Tbayt = r256[j];

sanS++;

byte ui = (byte)(Cbayt - 192);

nom[ui] = (byte)(j + 1);

ktlq[ui] = (byte)(w8[2]);

mudt[ui] = sanS;

if (LastRecord == 0xff) LastRecord = ui;

char hexstrbt2[3]; char hexstrbt4[3];

sprintf(hexstrbt2, "%.2x", ktlq[ui]);

sprintf(hexstrbt4, "%.2x", (nom[ui] + 5));

numarr[ind] = ui;

if (ind == lcd\_start)

sprintf(maze[ind], "R=%.2i t=%.2i:%.2i:%.2i", ui, mudt[ui] \* 4 / 3600, ((mudt[ui] \* 4) % 3600) / 60, ((mudt[ui] \* 4) % 3600) % 60);

else

sprintf(maze[ind], "r=%.2i t=%.2i:%.2i:%.2i", ui, mudt[ui] \* 4 / 3600, ((mudt[ui] \* 4) % 3600) / 60, ((mudt[ui] \* 4) % 3600) % 60);

Serial.println(maze[ind]);

ind++;

sanS = 0;

totalN++;

totalT += mudt[ui];

}

Cbayt = r256[j];

}

}

void GetFlights()

{

w1\_r1();

w8[0] = 0x80; w8[1] = 0x01; w8[2] = 0x02; w8[3] = 0x00; w8[4] = 0x00; w8[5] = 0x01; w8[6] = 0x00; w8[7] = 0xfc;

while (1)

{

w8\_r2();

w8[2]++; w8[7]--;

w2\_r2();

if (r2[0] == 0xfc && r2[1] == 0x04) { break; }

w1\_r1();

}

w1\_r1();

fb[3] = w8[3]; fb[4] = w8[4]; fb[5] = w8[5]; fb[6] = w8[6]; fb[7] = w8[7];

w8[2]--; w8[3] = 0x93; w8[7] = (byte)(w8[7] + 1 - w8[3]);

fb[2] = w8[2];

w8\_r2();

w2\_r2();

w1\_r1();

w8[3] = 0x0; w8[4] = 0x1; w8[5] = 0x0; w8[6] = 0x1; w8[7] = fb[7]--;

while (1)

{

w8\_r2();

for (int i = 0; i < 4; i++)

{

w2\_r65(i);

}

GetFlight();

w1\_r1();

if (w8[2] == (fb[2] + 1))

{

Tbayt = r256[0];

sanS++;

byte ui = (byte)(Cbayt - 192);

nom[ui] = (byte)(0);

ktlq[ui] = (byte)(w8[2]);

GetLastFlight(ui);

break;

}

if (w8[2] == 0x02) { w8[2] = 0x3f + 1; w8[7] = 0xbf - 2; }

w8[2]--; w8[4] = 0; w8[5] = 1; w8[6] = 0; w8[7] += 2;

w8\_r2();

w2\_r2();

w1\_r1();

w8[4] = 1; w8[5] = 0; w8[6] = 1; w8[7]--;

if (r2[0] != 0xf8 && r2[1] != 0x08)

{

Tbayt = r256[0];

sanS++;

byte ui = (byte)(Cbayt - 192);

nom[ui] = (byte)(0);

if (w8[2] == 0x3f)

ktlq[ui] = (byte)(0x02);

else

ktlq[ui] = (byte)(w8[2] - 1);

GetLastFlight(ui);

break;

}

}

lcd.setCursor(0, 1);

lcd.print(maze[lcd\_start]);

}

void WriteFlight()

{

byte inn = LastRecord;

w8[0] = 0x80; w8[1] = 0x01; w8[3] = 0x00; w8[5] = 0x00; w8[6] = 0x01;

w8[2] = ktlq[inn]; w8[4] = (byte)(nom[inn] + 5); w8[7] = (byte)(0xff - w8[2] - w8[4]);

int ms = mudt[inn];

for (int bb = 0; bb < ms; bb++)

{

if (w8[4] == 0)

{

w8[4] = 5;

w8[2]++;

w8[7] = (byte)(255 - w8[2] - 6);

}

w8[7]--;

w8\_r2();

for (byte i = 0; i < 4; i++)

{

w2\_r65(i);

}

w1\_r1();

w8[4]++;

}

}

void Init()

{

t = 0;

lcd.begin(16, 2);

lcd\_start = 0;

w8[0] = 0x80; w8[1] = 0x01; w8[2] = 0x02; w8[3] = 0x00; w8[4] = 0x00; w8[5] = 0x01; w8[6] = 0x00; w8[7] = 0xfc;

w2[0] = 0x80; w2[1] = 0x80;

w1[0] = 0x80;

ind = 0;

BortNumber = 0;

fb[0] = 0x80; fb[0] = 0x01; fb[0] = 0x02; fb[0] = 0x00; fb[0] = 0x00; fb[0] = 0x01; fb[0] = 0x00; fb[0] = 0xfc;

LastRecord = 0xff;

first = false;

DownButtonState = 0;//DownButton 16

UpButtonState = 0;//UpButton 15

SaveButtonState = 0;//SaveButton 14

Serial1.begin(57600, SERIAL\_8N2);

while (!Serial1) { SysCall::yield(); }

pinMode(13, OUTPUT);

pinMode(6, INPUT);

pinMode(7, INPUT);

pinMode(8, INPUT);

lcd.print("SD card OK!");

lcd.setCursor(0, 1);

lcd.print("Ready to receive");

Serial1.flush();

}

void setup() {

Init();

myStart();

lcd.blink();

GetFlights();

lcd.noBlink();

}

void myStart()

{

tt = millis();

w8[0] = 0x80; w8[1] = 0x01; w8[2] = 0x01; w8[3] = 0x40; w8[4] = 0x0a; w8[5] = 0x0c; w8[6] = 0x00; w8[7] = 0xa8;

Serial1.flush();

Serial.flush();

w8\_r2();

w2\_r13();

BortNumber = ((r13[1] << 8) + r13[0]);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Bort "); lcd.print(BortNumber);

}

void loop()

{

DownButtonState = digitalRead(6);

UpButtonState = digitalRead(7);

SaveButtonState = digitalRead(8);

if (DownButtonState == HIGH) {

if (lcd\_start < totalN)

{

lcd\_start++;

if (lcd\_start >= totalN)lcd\_start = 0;

lcd.setCursor(0, 1);

lcd.print(maze[lcd\_start]);

delay(200);

}

}

if (UpButtonState == HIGH) {

if (lcd\_start >= 0)

{

lcd\_start--;

if (lcd\_start < 0)lcd\_start = totalN - 1;

lcd.setCursor(0, 1);

lcd.print(maze[lcd\_start]);

Serial.print(lcd\_start); Serial.print(" ");

Serial.println(maze[lcd\_start]);

delay(200);

}

}

if (SaveButtonState == HIGH) {

digitalWrite(13, HIGH);

char filename[10] = "log08.dat";

char strBortNumber[6];

sprintf(strBortNumber, "%.3i", BortNumber);

filename[0] = strBortNumber[2];

filename[1] = strBortNumber[3];

filename[2] = strBortNumber[4];

LastRecord = numarr[lcd\_start];

char strLastRecord[3];

sprintf(strLastRecord, "%.2i", LastRecord);

filename[3] = strLastRecord[0];

filename[4] = strLastRecord[1];

sd.remove(filename);

if (!dataFile.open(filename, O\_CREAT | O\_APPEND | O\_WRITE)) return;

sanS = 0; Tbayt = 0xff; Cbayt = 0xff;

lcd.setCursor(0, 0);

lcd.print(" ");

lcd.setCursor(0, 0);

lcd.print(filename);

lcd.blink();

tt = millis();

WriteFlight();

lcd.noBlink();

int b\_data = (millis() - tt) / 1000;

lcd.print(" ");

lcd.print(b\_data);

lcd.print("s");

dataFile.close();

}

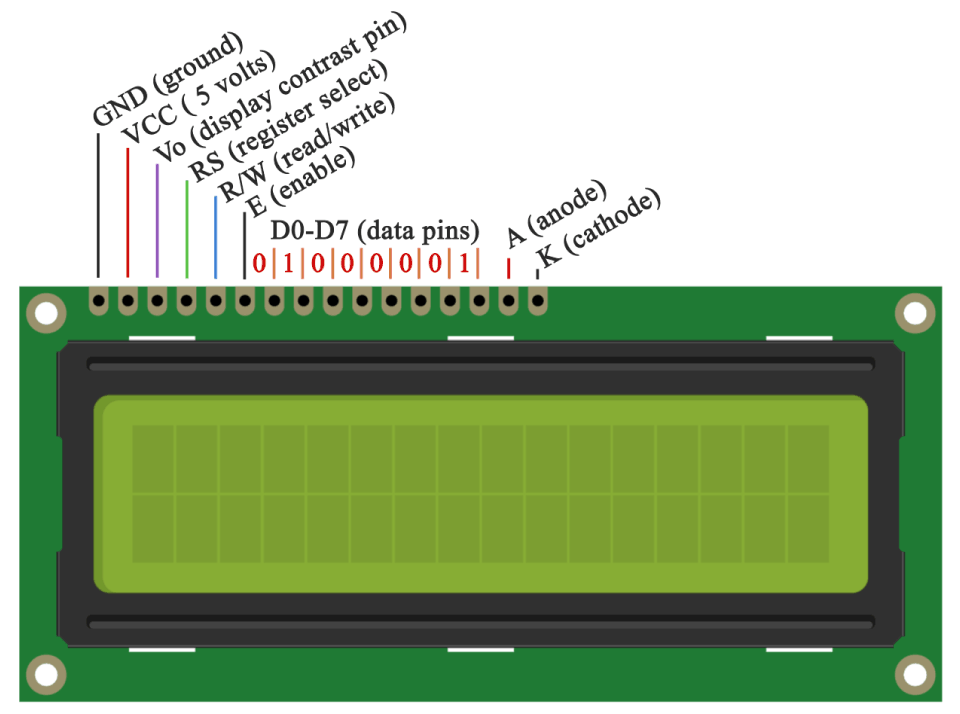
delay(50);

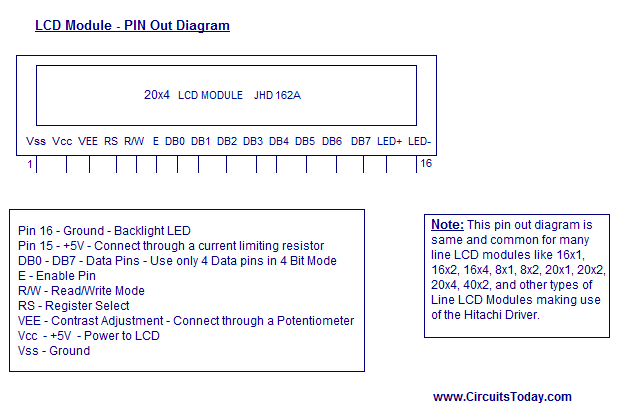
}

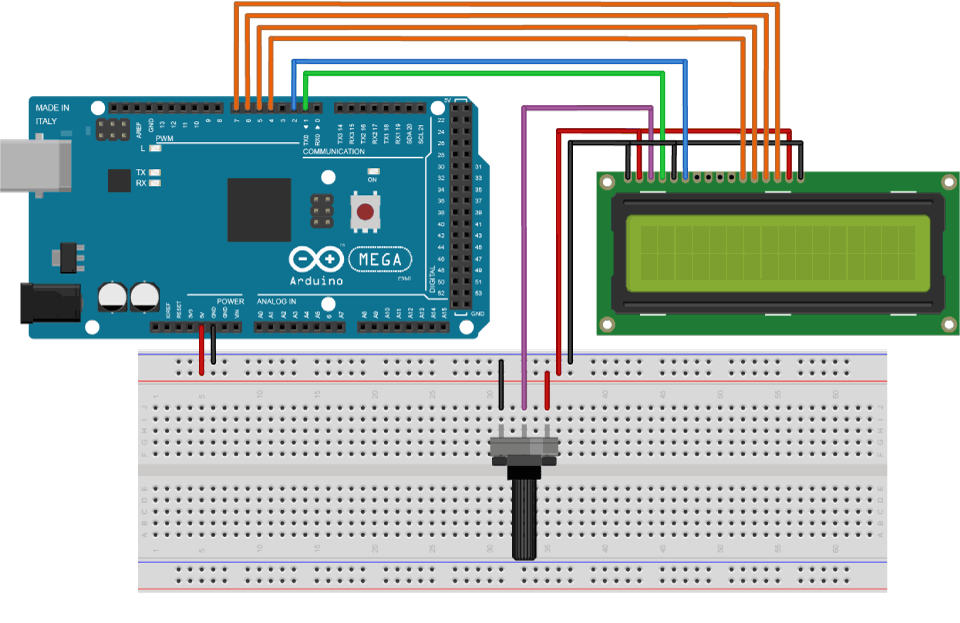
**Fəsil IV.Alqoritim və proqramların təsviri və simulyatorda test edilməsi.**

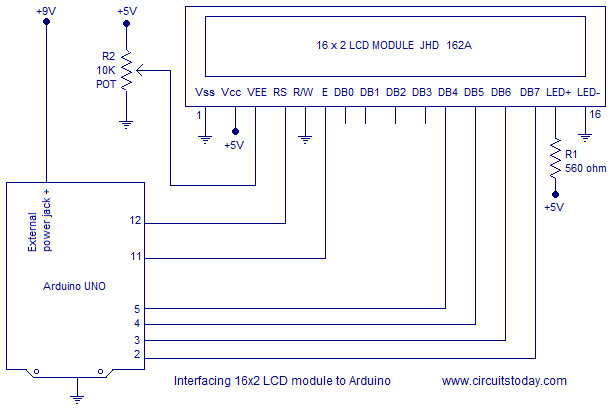
**§4.1 Arduinodan istifadə edərək mətni və simvolları LCD ekranda göstərmək.**

Bu Arduino LCD Təlimatında bir LCD (Maye Kristal Ekran) Arduino lövhəsinə necə bağlanacağını öyrənəcəyik. Bu kimi LCD-lər çox populyardır və elektron layihələrdə geniş istifadə olunur, çünki proyektinizdəki sensorlar məlumatları kimi məlumatları göstərmək üçün yaxşıdır və çox ucuzdur.16 sancaq var və soldan sağa birincisi Yer pinidir. İkinci pin, Arduino lövhəsindəki 5 volt pimi bağladığımız VCC-dir. Bundan sonra ekranın kontrastını idarə etmək üçün potensiometr əlavə edə biləcəyimiz Vo pin.Daha sonra, RS pin və ya qeyd seçmə pimi LCD-ə əmrlər və ya məlumat göndərəcəyimizi seçmək üçün istifadə olunur. Məsələn, RS pini aşağı vəziyyətdə və ya sıfır voltda quraşdırılıbsa, LCD-ə əmrlər göndəririk: kursoru müəyyən bir yerə qoyun, ekranı silin, ekranı söndürün və s. RS pin yüksək vəziyyətdə və ya 5 voltda qurulduqda, LCD-yə məlumat və ya simvol göndəririk

****

****

Ardından LCD-də oxuyub yazmayacağımızı seçən R / W pin gəlir. Burada yazma rejimi aydındır və əmrləri və məlumatları LCD-yə yazmaq və ya göndərmək üçün istifadə olunur. Oxu rejimi bu təlimatda bu barədə müzakirə etməyə ehtiyac duymadığımız proqramı icra edərkən LCD-nin özü tərəfindən istifadə olunur.Bundan sonra qeydlərə yazmağı təmin edən E pin və ya D0-dan D7-ə qədər növbəti 8 məlumat pinidir. Beləliklə, bu pinlər vasitəsilə qeydlərə yazarkən 8 bitlik məlumat göndəririk və ya məsələn sonuncu böyük A-nı ekranda görmək istəyiriksə, ASCII cədvəlinə əsasən qeydlərə 0100 0001 göndərəcəyik. Son iki pin A və K və ya anod və katot LED arxa işığı üçündür.Nəticədə, Liquid Crystal Library demək olar ki, hər şeyin qayğısına qaldığından, LCD-nin necə işləməsi barədə çox narahat olmamalıyıq. Arduinonun rəsmi saytından kitabxananın LCD-dən asanlıqla istifadə etməyə imkan verən funksiyalarını tapa və görə bilərsiniz. Kitabxananı 4 və ya 8 bit rejimində istifadə edə bilərik. Bu təlimatda 4 bit rejimində istifadə edəcəyik və ya yalnız 8 məlumat sancağından 4-nü istifadə edəcəyik.Arduino Board-dan yalnız 6 rəqəmsal giriş pinindən istifadə edəcəyik. LCD-nin D4-dən D7-yə qədər olan qeydləri Arduinonun rəqəmsal sancaqlarına 4-dən 7-ə qədər bağlanacaq. Enable pin 2 nömrəli, RS pinli 1 nömrəli pinlə birləşdiriləcək. R / W pin Ground və Vo pin potensiometrlə birləşdiriləcəkdir.**Şəkil.4.1**



**§4.2 Proqram kodları.**

Etməli olduğumuz ilk şey Maye Kristalı Kitabxanaya daxil etməkdir. Bunu belə edə bilərik: Eskiz> Kitabxana əlavə et> Maye Kristal. Sonra bir LC obyekti yaratmalıyıq. \Bu obyektin parametrləri, Arduino Kartının Rəqəmsal Giriş sancaqlar sırasıyla LCD düymələrinə aşağıdakı kimi olmalıdır: (RS, Enable, D4, D5, D6, D7). Quraşdırmada interfeysi LCD-yə başlamalıyıq və start () funksiyasından istifadə edərək ekranın ölçülərini təyin etməliyik.Döngədə əsas proqramımızı yazırıq. Print () funksiyasından istifadə edərək LCD-də yazdırırıq. SetCursor () funksiyası LCD-yə yazılmış sonrakı mətnin göstəriləcəyi yeri təyin etmək üçün istifadə olunur. Blink () funksiyası yanıb sönən bir imleci və söndürmək üçün noBlink () funksiyasını göstərmək üçün istifadə olunur. Cursor () funksiyası alt cizgi və noCursor () funksiyasını söndürmək üçün istifadə olunur. Clear () funksiyasından istifadə edərək LCD ekranı təmizləyə bilərik.

/\*

**\* Arduino LCD Tutorial**

**\***

**\* Crated by Dejan Nedelkovski,**

**\* www.HowToMechatronics.com**

**\***

**\*/**

**#include <LiquidCrystal.h> // includes the LiquidCrystal Library**

**LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // Creates an LC object. Parameters: (rs, enable, d4, d5, d6, d7)**

**void setup() {**

**lcd.begin(16,2); // Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display }**

**}**

**void loop() {**

**lcd.print("Arduino"); // Prints "Arduino" on the LCD**

**delay(3000); // 3 seconds delay**

**lcd.setCursor(2,1); // Sets the location at which subsequent text written to the LCD will be displayed**

**lcd.print("LCD Tutorial");**

**delay(3000);**

**lcd.clear(); // Clears the display**

**lcd.blink(); //Displays the blinking LCD cursor**

**delay(4000);**

**lcd.setCursor(7,1);**

**delay(3000);**

**lcd.noBlink(); // Turns off the blinking LCD cursor**

**lcd.cursor(); // Displays an underscore (line) at the position to which the next character will be written**

**delay(4000);**

**lcd.noCursor(); // Hides the LCD cursor**

**lcd.clear(); // Clears the LCD screen**

**}**

**LCD-yə xüsusi bir simvo**l yazmaq da mümkündür. 5 × 8 pikseldən 8 simvolu dəstəkləyir. Hər bir simvolun görünüşünü 8 baytlıq bir sıra ilə təyin edə bilərik. Aşağıdakı mənbə kodunda, 5 × 8 pikseli təmsil edən 0-u 1-ə dəyişdirərək xarakterin görünüşünü necə təyin edə biləcəyimizi görə bilərik. Quraşdırmada createChar () funksiyasından istifadə edərək xüsusi xarakter yaratmalıyıq. Bu funksiyadakı ilk parametr 0 ilə 7 arasındakı bir rəqəmdir və ya dəstəklənən 8 xüsusi simvoldan birini sifariş etməliyik. İkinci parametr bayt massivinin adıdır. Xüsusi işarəni write () funksiyasından istifadə edərək ekrana yazırıq və parametr olaraq simvol sayından istifadə edirik.

**#include <LiquidCrystal.h>**

**byte slash[8]= { // Array of bytes**

**B00001, // B stands for binary formatter and the 5 numbers are the pixels**

**B00010,**

**B00100,**

**B01000,**

**B10000,**

**B00000,**

**B00000,**

**B00000,**

**};**

**LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // Creates an LC object. Parameters: (rs, enable, d4, d5, d6, d7)**

**void setup() {**

**lcd.begin(16,2); // Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display**

**lcd.createChar(7, slash); // Create a custom character for use on the LCD. Up to eight characters of 5x8 pixels are supported**

**}**

**void loop() {**

**for(int i=0;i<=15;i++) {**

**lcd.setCursor(i,0); // Sets the location at which subsequent text written to the LCD will be displayed**

**lcd.write(7); // Writes a character to the LCD**

**delay(1000); // 1 second delay**

**lcd.clear(); // Write a character to the LCD**

**}**

}

/\*

\* Arduino LCD Tutorial

\*

\* Crated by Dejan Nedelkovski,

\* www.HowToMechatronics.com

\*

\*/

#include <LiquidCrystal.h> // includes the LiquidCrystal Library

LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // Creates an LC object. Parameters: (rs, enable, d4, d5, d6, d7)

void setup() {

lcd.begin(16,2); // Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display }

}

void loop() {

lcd.print("Arduino"); // Prints "Arduino" on the LCD

delay(3000); // 3 seconds delay

lcd.setCursor(2,1); // Sets the location at which subsequent text written to the LCD will be displayed

lcd.print("LCD Tutorial");

delay(3000);

lcd.clear(); // Clears the display

lcd.blink(); //Displays the blinking LCD cursor

delay(4000);

lcd.setCursor(7,1);

delay(3000);

lcd.noBlink(); // Turns off the blinking LCD cursor

lcd.cursor(); // Displays an underscore (line) at the position to which the next character will be written

delay(4000);

lcd.noCursor(); // Hides the LCD cursor

lcd.clear(); // Clears the LCD screen

}

LCD modulunun RS pimi arduinonun rəqəmsal pininə 12 bağlıdır. LCD-nin R / W pimi topraklanmışdır. LCD modulunun aktivləşdirmə pimi arduinonun 11 nömrəli pininə bağlıdır. Bu layihədə LCD modul və arduino 4-bit rejimində qarşı-qarşıya qoyulur. Bu, LCD-nin yalnız dörd rəqəmsal giriş xəttindən (DB4 - DB7) istifadə olunduğunu göstərir. Bu metod çox sadədir, daha az əlaqə tələb edir və demək olar ki, LCD modulunun bütün potensialından istifadə edə bilərsiniz. Rəqəmsal xətlər DB4, DB5, DB6 və DB7 Arduinonun 5, 4, 3 və 2 nömrəli sancaqları ilə əlaqələndirilir. 10K potensiometr ekranın kontrastını tənzimləmək üçün istifadə olunur. 560 ohm müqavimət R1, arxa işıq LED-i ilə cərəyanı məhdudlaşdırır. Arduino, lövhədə göstərilən xarici güc jakı vasitəsilə işləyə bilər. Devrenin bəzi digər hissələrində tələb olunan + 5V arduino lövhəsindəki 5V mənbəyindən vurula bilər. Arduino PC-dən USB portu ilə də təchiz edilə bilər. Arduinoya LCD interfeysi üçün tam proqram aşağıda göstərilmişdir.

/\*

\* Arduino LCD Tutorial

\*

\* Crated by Dejan Nedelkovski,

\* www.HowToMechatronics.com

\*

\*/

#include <LiquidCrystal.h> // includes the LiquidCrystal Library

LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // Creates an LC object. Parameters: (rs, enable, d4, d5, d6, d7)

void setup() {

lcd.begin(16,2); // Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display }

}

void loop() {

lcd.print("Arduino"); // Prints "Arduino" on the LCD

delay(3000); // 3 seconds delay

lcd.setCursor(2,1); // Sets the location at which subsequent text written to the LCD will be displayed

lcd.print("LCD Tutorial");

delay(3000);

lcd.clear(); // Clears the display

lcd.blink(); //Displays the blinking LCD cursor

delay(4000);

lcd.setCursor(7,1);

delay(3000);

lcd.noBlink(); // Turns off the blinking LCD cursor

lcd.cursor(); // Displays an underscore (line) at the position to which the next character will be written

delay(4000);

lcd.noCursor(); // Hides the LCD cursor

lcd.clear(); // Clears the LCD screen

}

##### **Program – Arduino to LCD**

#include<LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // sets the interfacing pins

void setup()

{

lcd.begin(16, 2); // initializes the 16x2 LCD

}

void loop()

{

lcd.setCursor(0,0); //sets the cursor at row 0 column 0

lcd.print("16x2 LCD MODULE"); // prints 16x2 LCD MODULE

lcd.setCursor(2,1); //sets the cursor at row 1 column 2

lcd.print("HELLO WORLD"); // prints HELLO WORLD

}

Arduino ilə LCD modulu arasında əlaqəni asanlaşdırmaq üçün, Hitachi HD44780 çipsetindən (və ya uyğun bir çipsetdən) istifadə edən LCD modulları üçün yazılmış Arduino <LiquidCrystal.h> içərisində quraşdırılmış kitabxanadan istifadə edirik. Bu kitabxana həm 4 bitli, həm də 8 bitlik LCD kabellə işləyə bilər.

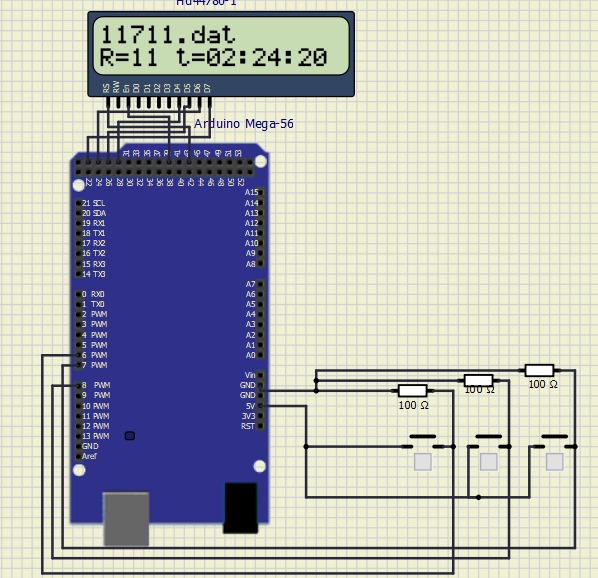
“LiquidCrystal.h” kitabxanası Arduino lövhəsini istifadə edərək LCD modulunu kitabxananın içərisində qurulmuş metodların köməyi ilə asanlıqla idarə etmək üçün istifadə olunur. Məsələn, lcd.print () metodunu çağıraraq məlumat modulu LCD moduluna yazdırıla bilər. . Sütun 3-dən başlayaraq 1-ci sətirdə “Salam Dünya” yazdırmaq istəyirsinizsə; əvvəlcə metodu lcd.setCursor (1,3) istifadə edərək kursoru istədiyiniz yerə qoyun və sonra simvolları lcd.print (“Dünyaya Salam”) şəklində yazdırmaq əmrini yazın; - anladım? Kitabxana Arduino IDE ilə (əvvəlcədən quraşdırılmış standart kitabxana kimi) hazırdır. Hər hansı bir kitabxanaya əsas menyu çubuğundakı "eskiz" sekmesindeki "Kitabxananı idxal et" vasitəsi ilə əl ilə daxil olmaq mümkündür. LiquidCrystal.h kitabxanası demək olar ki, simli çap etmək, imleci təyin etmək, LCD-ni işə salmaq, ekranı sürüşdürmək, avtomatik sürüşdürmək, LCD-ni təmizləmək, yanıb-sönən imlec və s. Kimi funksiyalar / metodlar təmin edir.

LiquidCrystal lcd () - növünün dəyişənini elan etmək üçün istifadə olunan bir konstruktordur. Burada ‘lcd’ konstruktordan istifadə edərək elan edilmiş dəyişəndir və kitabxana LiquidCrystal.h daxilində müəyyən edilmiş metodlara müraciət etmək üçün istifadə olunur (Nümunə - lcd.print (); lcd.setCursor () və digər metodlar)

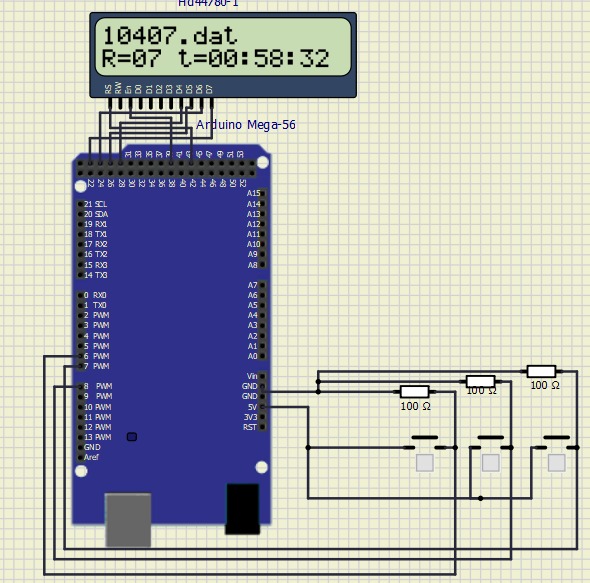
lcd.begin () - lcd ekranını işə salmaq və lcd ekranın ölçüsünü (sütunlar, sətirlər) çağırılan metodun parametrləri kimi keçirmək üçün çağırılır.

**§4.3 Arduino Mega 560 mikrokontrollerin və LCD ekranin köməyi ilə uçuşlarin oxunması.**

**117 saylı bortdan 11 saylı uçuşun oxunmas, uçuş müddəti 2 saat 24 dəq 20 san**

****

**104 saylı bortun 07 saylı uçuşun oxunması, müddət 58 dəq 32 san**



**ƏDƏBIYYAT**

1. Бортовое устройство регистрации БУР-СЛ-1 серия 1-8. Руководство по технической эксплуатации МИВЮ.794121.004 РЭ

2. ОСТ 1 01080-95 Устройства регистрации бортовые с защищенными накопителями. Общие технические требования.

https://pdf.standartgost.ru/catalog/Data2/1/4293814/4293814288.pdf

3. Single RS-232 Transceiver with AutoShutdown MAX221 Full Data Sheet.

https://datasheets.maximintegrated.com/en/ds/MAX221.pdf

4.*https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/*

5.*https://create.arduino.cc/projecthub/muhammad-aqib/arduino-button-tutorial-using-arduino-digitalread-function-08adb5*