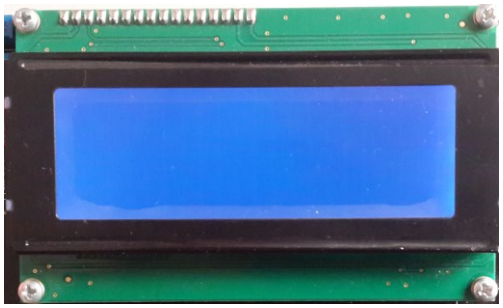
LCD EKRANIN SÜRÜLMESİ VE FLASH/EE HAFIZASININ KULLANILMASI

1. Deneyin Amacı

a. Kullanıcı ve mikrokontrolör arasında bir arayüz elamanı olan LCD'lerin sürülmesi
b. ADUC841 mikrokontrolör mimarisinde bulunan 4 kByte lık Flash/EE veri hafızanın kullanılması
c. Veri tabanı (Database, DB) kullanılması
d. Timer0 kesmesi kullanılması

2. LCD (Liquid Crystal Display) Ekranın Tanıtılması



Şekil 1: Deneyde kullanılacak olan LCD 2x20

Şekil 2: Grafik LCD

LCD bir görüntüleme teknolojisidir. Bu teknolojiyi kullanan cihazlar ise LCD gösterge olarak adlandırılmaktadır. 7-Segment (7 Parçalı Gösterge) göstergelerin fazla akım çekmesi ve kullanım zorluğu nedeniyle, son yıllarda LCD göstergelerin kullanımı popüler hale gelmiştir.

Bu deneyde kullanacağımız LCD (ve çoğu LCD), aslında sadece bir göstergeden ibaret değildir. Bünyesinde,

1. Kendisine ait bir **mikroişlemcisi**,
2. RAM ve ROM'LARI,
3. Giriş çıkışları,

olan bir cihazdır. Yani bir nevi **bütünleşik mikrodenetleyici+LCD** cihazdır.

LCD'ler 1 satırdan 4 satıra kadar, 16 karakterden 80 karaktere kadar ve 5X7, 5X10 nokta font gibi değişik ölçülerde üretilip satılmaktadır. Bazılarında ise tüm ekran tek bir karakter gibi yapılandırılmıştır, bu türlerine grafik LCD gösterge adı verilmektedir. LCD gösterge ile iletişim, TTL standardında 4 veya 8 veri hattı ile yapılır. 4 bit iletişim G/Ç hatlarının başka işler(görevler, amaçlar) için kullanımını kolaylaştırırken, iletişim süresini iki kat uzatmaktadır.

2.2. LCD Hafıza Haritası (Memory Map)

LCD göstergeler üzerinde kullanılan denetleyiciler, Hitachi firması tarafından üretilen **HD44780U** mikrodenetleyicisidir ve bu mikrodenetleyici standart bir hale gelmiştir.

LCD'lerin mimarisinde 3 adet hafıza yapısı bulunmaktadır. Bunlar DDRAM, CGROM ve CGRAM'dir.

DDRAM: LCD göstergeler, 40 karaktere ve 4 satıra kadar değişik seçenekler sunar. LCD göstergeler 80 adet karaktere kadar kodu saklayabilmek için dâhili bir RAM bulundururlar. Bu RAM'a Gösterge Veri RAM'i(Display Data RAM-DDRAM) denir. Örneğin bir satırında 16 karakteri olan iki satırlık bir göstergeyi, her birinde 40 karakteri olan 2 sanal satır olarak düşünebiliriz. 40 karakterlik bir satır bulunmaktadır ancak biz onu 16 karakterlik bir pencere ile görebilmekteyiz. Sanal satırdaki diğer karakterleri görebilmek için gösterge kaydırılmalıdır. Örneğin, bu göstergenin birinci satırına aşağıdaki 40 karakterli diziyi yazalım.

9876543210QWERTYUIOPLKJHGFDSAZXCVBNMsedn

Göstergede sadece ilk 16 tanesi gözükecektir.

9876543210QWERTY

Bu gösterimde, aşağıda bahsedileceği gibi Entry (Giriş) moduna bağlı olarak ekran kayabilir veya kaymayabilir. Burada ekran kaydırılmamıştır. “U” karakterini görüntülemek istediğimizde, aşağıdaki gibi ekran bir karakter sağa kayacak ve “9”karakteri gizlenecektir.

876543210QWERTYU

CGROM: LCD göstergede önceden programlanmış veya kullanıcı tarafından karakterleri tanımlanan gösterebilmektedir. LCD kontrolörü, 192 adet karakter içeren bir Karakter Üretici ROM’a (Character Generator ROM –CGROM) sahiptir. Karakterler belirli kodlarla seçilirler. Bu karakterlerden 96 tanesi ASCII karakter (ASCII kodlarla seçilir) , 64 adeti japon karakterleri (Kana Alfabesi) ve 32 adeti Yunan harfleri gibi özel karakterlerdir.

CGRAM: LCD kontrolörü aynı zamanda, Karakter Üretici RAM (Character Generator RAM – CGRAM) olarak adlandırılan ve kullanıcı tarafından tanımlanabilen 8 karakter içerebilen bir hafızaya sahiptir. Bu karakter kullanılmadan önce tanımlanmalı, CGRAM’a yüklenmeli ve gösterilmek için gereken yerlerde çağrılmalıdır.

ASCII KOD TABLOSU: CGROM hafızasında dâhili olarak bulunur. Latin alfabesi üzerine kurulu 7 bitlik bir karakter setidir. ASCII'de 33 tane basılmayan (ekranda görülmeyen) kontrol karakteri ve 95 tane basılan (ekranda görülen) karakter bulunur. Kontrol karakterleri metnin akışını kontrol eden, ekranda çıkmayan karakterlerdir. Basılan karakterler ise ekranda görünen, okuduğumuz metni oluşturan karakterlerdir.

Tablo 1: ASCII Kod Tablosu

*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

Tablo 1’den bir karakterin ASCII kodunu bulmak için önce karakterin bulunduğu satır numarası sonrada sütün numarasına bakılır. Aşağıda örnek olarak çeşitli karakterlerin ASCII kodları verilmiştir.

Karakter	ASCII Kodu
A	41’H
a	61’H
5	35’H
=	3D’H

LCD Ekranın Adreslenmesi

(LCD Göstergeye bu dokümanda bundan sonra kısaca LCD denilecektir.)

LCD’lerde aşağıda gösterildiği gibi her karakterin ayrı bir adresi vardır. Aşağıdaki resimlerde 2x16 ve 4x20 karakter boyutundaki LCD’lerin her karakter için belirlenen adresleri verilmiştir.

2x16

80	81	·	·	·	·	·	8E	8F
C0	C1	·	·	·	·	·	CE	CF

4x20

80	81	·	·	·	·	·	·	·	92	93
C0	C1	·	·	·	·	·	·	·	D2	D3
94	95	·	·	·	·	·	·	·	A6	A7
D4	D5	·	·	·	·	·	·	·	E6	E7

Değişik firmalar tarafından üretilen LCD’lerde, LCD ekranın sürücü devresini de içerir. Dolayısıyla ilave devre kurmadan doğrudan mikrokontrolör ile LCD modüller sürülebilir. LCD’ler farklı firmalar tarafından üretilmelerine rağmen erişim protokolleri çoğunlukla aynıdır. LCD’lerin 3 kontrol hattı (RS, R/W, E) ve 8 veri hattı (DB0…DB7) vardır.

2.2. LCD Bağlantı Uçları

Şekil 2'de görüldüğü gibi günümüzde üretilen LCD panellerin çoğunda tek sıra halinde 16 pin bulunur. Bazı LCD 'lerde kontrol için kullanılan 14 pin 2 adet 7'li sıra halinde de bulunabilir. LCD bağlantı uçları Tablo 2'de verilmiştir. Bağlantı uçlarını besleme, denetim ve veri olmak üzere üç grupta inceleyebiliriz.

Tablo 2: Pinlerin Görevleri

Pinlerin Görevleri					
No	**Sembol**	**Fonksiyon**	**No**	**Sembol**	**Fonksiyon**
1	Vss	GND	**9**	DB2	Komut veya Veri Yolu
2	Vdd	+5V	**10**	DB3	Komut veya Veri Yolu
3	Vee	LCD Sürmek için	**11**	DB4	Veri Yolu
4	RS	Fonksiyon Seçimi	**12**	DB5	Veri Yolu
5	R/W	Okuma/Yazma	**13**	DB6	Veri Yolu
6	E	Sinyal Aktifleş.	**14**	DB7	Veri Yolu
7	DB0	Komut veya Veri Yolu	**15**	LEDA	Işık +5v
8	DB1	Komut veya Veri Yolu	**16**	LEDA	Işık 0V

2.1.1. Besleme Gerilimleri

HD44780U standardında besleme ile ilgili üç uç yer almaktadır. Bunlar **Vcc**, **Vee (V0)**, **Vss (GND)**'dir. Vss ve Vcc standart TTL gerilimi 0 ve 5 Volttur. Vee ise ekranın parlaklığını belirleyen bir gerilimdir ve değeri en çok Vcc'dir.

2.1.2. Kontrol ve Veri Pinleri

LCD'yi kontrol etmek amaçlı üç uç yer almaktadır. Bunlar **RS**, **R/W**, **E** uçlarıdır. Ayrıca 8 tane de veri ucu bulunmaktadır. Bunlar;

Tablo 3: Kontrol Uçları

Sembol	**Görevi**
RS	LCD'ye komut mu yoksa veri mi gönderileceğini belirler. LCD ekrana veri aktarılacaksa **RS= 1**, komut gönderilecekse **RS= 0** LCD ekranı silme, kursör on/off, kursör başa dön, yazma başlangıç adresinin belirtilmesi gibi işlemler komut olarak adlandırılır. LCD'lerde kullanılan komutlar ve ilgili komutlar için pin değerleri aşağıda tablo halinde verilmiştir. LCD ekrana yazılan (örneğin "SAU","12+3=15", vb.) değerler ise veri olarak adlandırılır.
R/W	Lcd den okuma mı yoksa lcd ye yazma yapılacağını belirler. Lojik **R/W=1** seviyesi LCD'lerden okuma, lojik **R/W=0** ise LCD'ye yazma işlemini gösterir. Deneylerde LCD'den okuma işlemi **yapılmayacağı** için bu pin Şekil 1 de gösterildiği gibi donanımsal olarak GND pinine bağlanarak Lojik 0 seviyesinde tutulmuştur.
E	Enable (Aktifleştirme) ucu LCD ve pinler arasındaki gerçek veri alışverişini sağlayan bacaktır. Bu girişi mikrodenetleyiciye program aracılığıyla tanıttıktan sonra mikrodenetleyici kendisi veri gönderileceği zaman bu bacağa enable (aktifleştir) darbesi gönderir. Yani bu uca 0-1-0 darbesi üretilir.
DB0-DB7	Data hattı olan bu pinler doğrudan mikrodenetleyicinin bir portuna bağlanır. Veri 4 ya da 8 bitlik veri yolu ile gönderilebilir.

2.2. LCD Komut Tipleri ve Zaman Çizelgesi

Bir LCD işlemi, ya kontrol ya da veri işlemidir. Kontrol işlemleri, ya LCD'ye gönderilen komutlardır, ya da LCD'den okunan bir hafıza adresidir (kaydedici-register). RS (Register Select) ucu lojik 0'a çekilirse yapılacak işlem, kontrol işlemidir. Bütün komutlar ve sinyaller, harici bir mikrodenetleyici, bilgisayar vb. tarafından üretilir. Komutlar LCD'ye 8-bit olarak yazılır. Bunun için R/W lojik 0'da tutulmalıdır. Bu uç lojik 1'e çekilirse, LCD'den veri okunabilir. Komut çeşitleri ve ayar bitleri Tablo 4'te verilmiştir.

Tablo 4: LCD Komut Gönderimi

KOMUT	KOD										İŞLEM SÜRESİ
	RS	**R/W**	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Ekranı Sil	**0**	**0**	0	0	0	0	0	0	0	1	1,64 ms
Kursör basa don	**0**	**0**	0	0	0	0	0	0	1	*	1,64 ms
Giriş kipini seç	**0**	**0**	0	0	0	0	0	1	I/D	S	40µs
Ekran aç/kapa	**0**	**0**	0	0	0	0	1	D	C	B	40µs
Kursör kaydır	**0**	**0**	0	0	0	1	S/C	R/L	*	*	40µs
Fonksiyon seç	**0**	**0**	0	0	1	DL	N	F	*	*	40µs
Meşgul bayrağını oku	**0**	**1**	**BF**	DDRAM ADRESİ							0µs
Veri yaz	**1**	**0**	VERİ YAZ								40µs
Veri oku	**1**	**1**	VERİ OKU								40µs

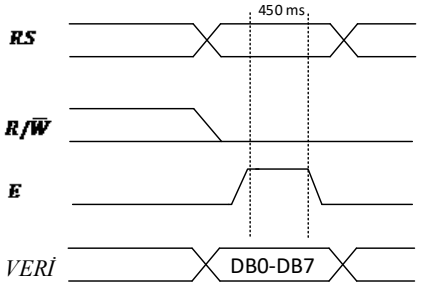
Komut seti; (* : Aldığı değer (1-0) önemsiz, DDRAM: Ekran Veri Belleği) Not: Tabloda belirtildiği gibi LCD modüle yazma işlemi minimum 40us sürmektedir. Dolayısı ile ard arda yapılan yazma işlemlerinde bir önceki verinin yazılabilmesi için en az belirtilen süre kadar beklenmelidir.

Tablo 5: Kontrol bitlerinin görevleri

Kod		AÇIKLAMA	
I/D		0 = Her yazma işleminden sonra kursör pozisyonunu azalt	1= Her yazma işleminden sonra kursör pozisyonunu arttır
S		0 = Ekran kaydırma modu kapalı	1 = Ekran kaydırma modu açık
D		0 = Ekran kapalı	1 = Ekran açık
C		0 = Kursör kapalı	1 = Kursör açık
B		0 = Kursör blink kapalı	1 = Kursör blink açık
S/C		0 = Kursör taşınması gerekir. Manuel.	1 = Kursör kaydır
R/L		0 = Sola kaydır	1 = Sağa kaydır
DL		0 = Veri hattı 4 bit	1 = Veri hattı 8 bit
N		0 = 1 satır	1 = 2 satır
F		0 = 5x7 pixel	1 = 5x10 pixel
BF		0 = Komut kabul edebilir	1 = LCD Meşgul

2.2.1. Komut/Veri Gönderim Zaman Çizelgesi

Tablo 3'teki LCD kontrol pinlerinin zamanlama çizelgesi aşağıda verilmiştir.



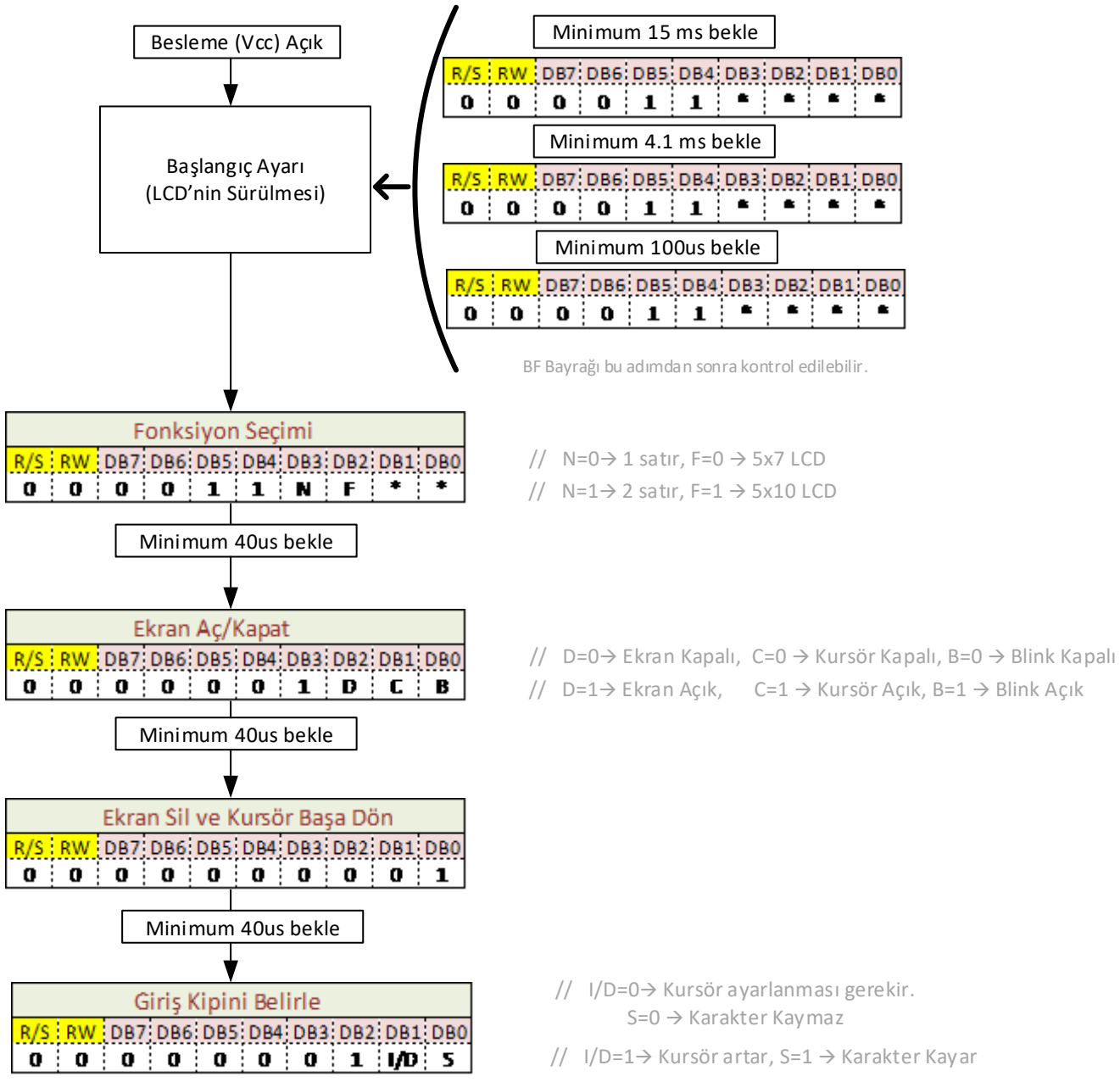
Şekil 3: LCD'ye veri yazmak için kontrol pinlerinin zamanlanması

2.3. LCD'nin Başlangıç Ayarlarının Yapılması

LCD'ye ilk enerji verilmesinden ardından, hazırlanan program ile arka arkaya 3 defa 30h komutu LCD'ye gönderilir. Burada bu komutun düşük dörtlüsü (son dört biti) ihmal edilir; DB7 veDB6=0, DB5 ve DB4=1 olarak atanır.

Yukarıdaki başlangıç ayarının ardından LCD için **fonksiyon belirleme** işlemi Tablo 4 ve Tablo 5 ile gerçekleştirilir yani veri yolunun büyüklüğü (4 bit veya 8 bit), göstergedeki satır sayısı (1-2-3-4) ve font büyüklüğü (5x7 veya 5x10 gibi) belirlenir. Ardından sırası önemli olmamak üzere giriş kipi, ekranın, kursörün, blink'in (imlecin) açık/kapalı ayarları yapılır. Giriş kipi; her karakter okuma veya yazma işlemini takiben imlecin veya göstergenin yerini belirler.

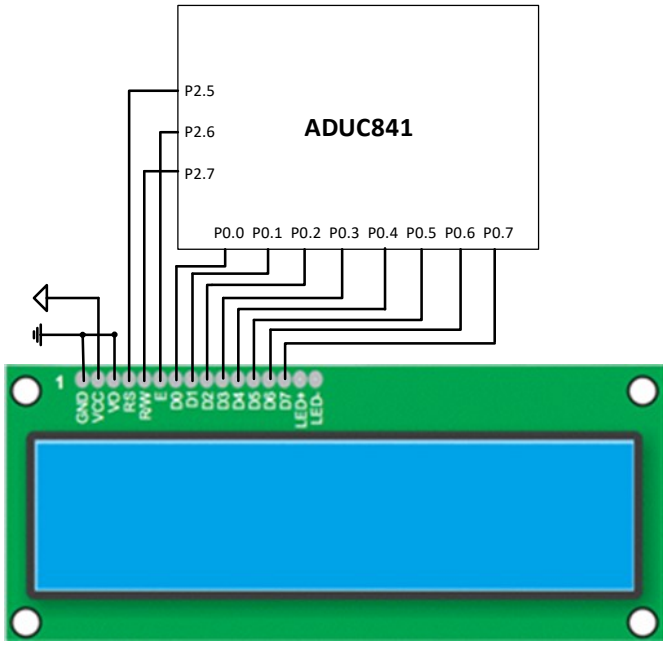
LCD ilk başlangıç ayarı aşağıdaki şekilde verilmiştir.



Şekil 4: LCD Başlangıç Ayarı

2.4. Deney Setindeki Aduc841 İle Lcd Bağlantı Şeması Ve Pinleri

ADUC841 deney setindeki mikrodenetleyici ile LCD arasında aşağıdaki şekilde bağlantılar gerçekleştirilmiştir.



3. FLASH VERİ HAFIZASI

ADUC841 mikrokontrolöründe bulunan 4 kByte Flash/EE veri hafızası her biri 4 Byte lık alana sahip 1024 sayfadan oluşur (4*1024 = 4096 Byte). Flash veri hafızası için 1 tane kontrol (ECON), 2 tane adres (EADRH/ EADRL) 4 tane veri tutmak (EDATA1-4) amacı ile kullanılan 7 adet SFR vardır.

Sayfa adreslenmesi (seçilmesi) EADRH/EADRL saklayıcıları kullanılarak gerçekleştirilir. 1024 adet sayfa için 0 ≤ EADRH/L < H'0400 olmalıdır. EADRH/L saklayıcıları Byte adresleme içinde kullanılır. Bu durumda 0 ≤ EADRH/L ≤ H'0FFF olur.

EDATA1-4 saklayıcıları okuma işlemi esnasında seçili olan sayfaya ait verileri tutar. Seçili sayfaya yazma işlemi de bu saklayıcılar üzerinden yapılır.

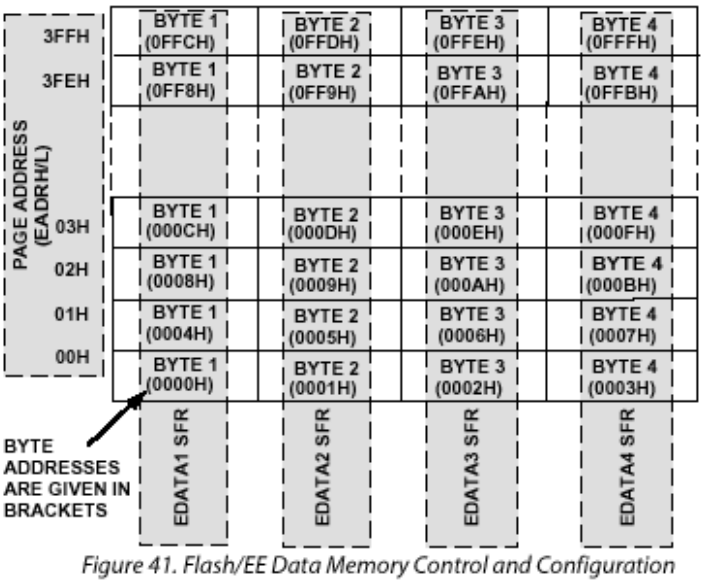


Figure 41. Flash/EE Data Memory Control and Configuration

Veri hafızasında yapılacak işlem (yazma, okuma, silme, doğrulama) 8-bitlik ECON SFR ile belirlenir.

ECON saklayıcısının aldığı değerlere göre yapılan işlemler şu şekildedir;

ECON = 01'H: READ = EADRH/L kullanılarak seçilen sayfadaki veriler EDATA1-4 saklayıcılarına aktarılır.
ECON = 02'H: WRITE = EDATA1-4 saklayıcılarındaki veriler seçilen sayfaya yazılır. (Not: Yazma işleminden önce ilgili sayfanın silinmesi gerekir.)
ECON = 04'H: VERIFY = Adreslenen sayfadaki değerlerin EDATA1-4 saklayıcılarındaki değerlerle aynı olup olmadığını kontrol eder. Eğer aynı iseler ECON SFR sinin değeri 0 olur, herhangi biri farklı ise ECON SFR si 0 dan dan farklı bir değer alır.
ECON = 05'H: ERASE PAGE = Adreslenen sayfayı siler, (temizler).
ECON = 06'H: ERASE ALL = 4 kByte lık veri hafızasının tamamını siler.
ECON = 81'H: READ BYTE = EADRH/L ile adreslenen Byte EDATA1 saklayıcısına aktarılır.
ECON = 82'H: WRITE BYTE = EDATA1 saklayıcısında tutulan veri EADRH/L ile adreslenen Byte' a yazılır.
ECON = 0F'H: EXULOAD = ECON SFR sinin Flash/EE veri hafızasının kontrolünde kullanılmasını sağlar.
ECON = F0'H: ULOAD = ECON SFR sinin Flash/EE program hafızasının kontrolünde kullanılmasını sağlar.

Flash/EE veri hafızasına yazma ve hafızadan okuma işlem süreleri aşağıda tabloda verilmiştir.

READPAGE (4 byte)	22 makine çevrimi
WRITEPAGE (4 byte)	380 µs
VERIFYPAGE (4 byte)	22 makine çevrimi
ERASEPAGE (4 bytes)	2 ms
ERASEALL (4 kByte)	2 ms
READBYTE (1 byte)	9 makine çevrimi
WRITEBYTE (1 byte)	200 µs

Örnek 1: Veri hafızasının H'203 sayfasına sayfanın 1.Byte ' tından başlayarak sırasıyla H'65, H'2B, H'3E, H'7F sabit değerlerini yazmak için yapılması gereken işlemler sırasıyla şu şekildedir;

1- Sayfa seçimi EADRH/L kullanılarak gerçekleştirilir.
2- ECON saklayıcısı kullanılarak (ECON =05'H) seçilen sayfa temizlenir (silinir).
3- Silme işleminden sonra ilgili saklayıcılara yazma değerleri aktarılır.
4- ECON saklayıcısına yazma komutu (ECON =02'H) yüklenir.
5- Yazma işleminin doğruluğu kontrol edilir. *(zaruri değil, isteğe bağlı)*

```
   MOV  EADRH, #02          ; sayfa seçimi
   MOV  EADRL,  #03
ERROR:
   MOV  ECON,#05            ; seçili olan sayfa silindi

   MOV  EDATA1, #H'65       ; birinci byte'a istenen değer yazıldı
   MOV  EDATA2, #H'2B       ; ikinci byte
   MOV  EDATA3, #H'3E       ; üçüncü byte
   MOV  EDATA4, #H'7F       ; dördüncü byte

   MOV  ECON,#02            ; EDATA1-EDATA4 saklayıcılarındaki veriler seçilen  sayfaya aktarıldı.
   MOV  ECON,#04            ; yazma işleminde hata var mı?
   MOV  A,ECON
   JNZ  ERROR               ; hata varsa yazma işlemini tekrarla
```

Veri hafızadan yapılacak bir okuma işlemi için, EADRH/L' saklayıcıları kullanılarak sayfa (veya byte) seçimi yapılır. Sayfa seçildikten sonra ECON SFR sine okuma komutu yazıldığında seçili sayfadaki (Byte'taki) değerler otomatik olarak EDATA1-4 (EDATA1) saklayıcılarına aktarılır. İstenen değerler bu saklayıcılardan, EDATA1-4, okunur.

Örnek 2: Veri hafızanın H'153 sayfasındaki 1. ve 2. Byte 'ı okuyup bu iki değeri çarpan program parçası şu şekildedir;

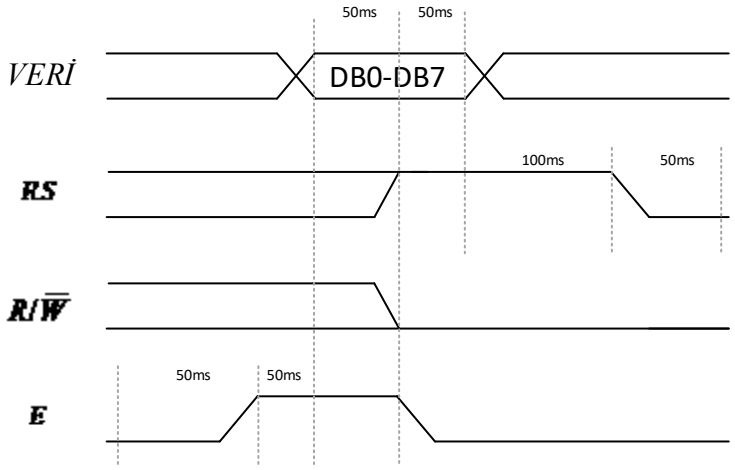
1- EADRH/L' saklayıcıları kullanılarak sayfa seçimi yapılır.
2- Sayfa seçildikten sonra ECON saklayıcısı ile okuma işlemi yapılacağı belirtilir. ECON' okuma komutu yazıldığında seçili sayfadaki değerler otomatik olarak EDATA1-4 saklayıcılarına aktarılır.
3- İstenen değerler bu saklayıcılardan, EDATA1-2, okunur.
4- Çarpma işlemi "mul AB" şeklinde olduğundan okunan değerler bu saklayıcılara aktarılır.

```
MOV   EADRH, #01
MOV   EADRL, #53H        ; sayfa seçimi

MOV   ECON,#01           ; seçili olan sayfadan okuma işlemi seçildi

MOV   A,EDATA1           ; H'53 sayfasındaki 1. Byte "A" ya
MOV   B,EDATA2           ; 2. Byte "B" ye  aktarıldı.
MUL   AB                 ; okunan iki değer birbiri ile çarpıldı..
```

ÖN ÇALIŞMA:



Şekil 5: Veri yazma zamanlama diyagramı

Yukarıdaki şekilde, bir cihaza ait kontrol pinlerinin veri yazma zamanlama diyagramı verilmiştir.

Şekil 5'i kullanarak **sadece** kontrol uçları için (3 pin için) asm kodunu yazınız. Yazılan asm kodunun kendini tekrarlanması istenmektedir. İstenilen aduc841 port ve pinleri kullanılabilir. Bekleme süreleri timer ya da alt program ile yapabilirsiniz.

UYGULAMA 1:

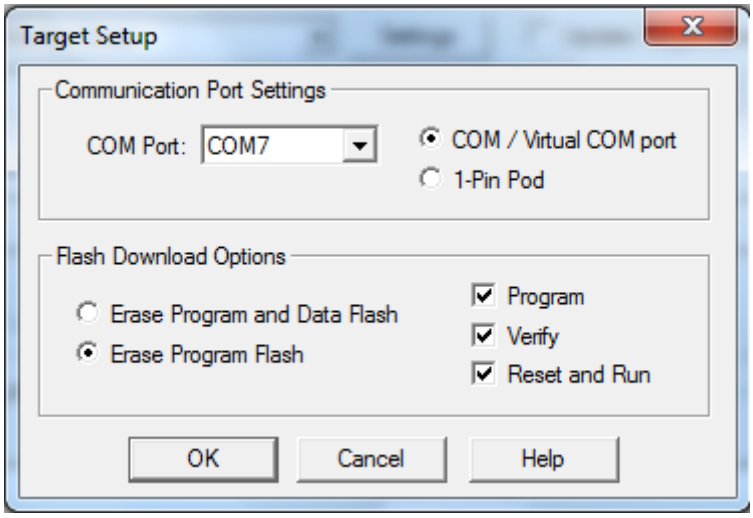
Deneyde aşağıdaki şekilde gözüktüğü gibi LCD ekranın ilk satırına 4. kolandan itibaren “KONTROL LAB. 2” ikinci satırına ise 2. kolandan itibaren “LCD ve Flash Deneyi” yazdırılacaktır.

UYGULAMA 2:

Veri hafızasının 3.sayfasına 0A’H sabiti yazılacaktır.

UYGULAMA 3:

Bir önceki deneyde veri hafızasının 3.sayfasına yazılan sayı (0Ah) yazıldığı adresten okunup LCD ekranın 1.satır 16.sütununda başlanarak, her 500ms de değeri bir azaltılarak (0 olana kadar) bir önceki karakter alanına yazılacaktır. Derlenen kodun hex dosyası Aduc841’e gönderilirken aşağıda verilen şekilde mikroişlemciye gönderilmesi gerekmektedir. (Data Flash’ın programlanmaması gerekmektedir.) Aksi takdirde bir önceki deneyde flash’a yazılan veri silinir. uVision ayarı aşağıdaki şekilde verilmiştir.



A	9	8	7	6	5	4	3	2	1	0					

(Not: Uygulama 2 de veri hafızaya yazma işlemi yapıldıktan sonra işlemcinin beslemesi kapatılıp tekrar açılır. Ardından uygulama 3 gerçekleştirilir. Uygulama 3'te veri hafızaya yazma işlemi yapmadan uygulama 2 de veri hafızaya yazılan veri okunup yukarıda belirtildiği gibi LCD ekrana aktarılacaktır.)

Uygulamalara Ait Kodlar

Uygulama 1:

```
#include<aduc841.h>

LRS    EQU   P2.5
LEE    EQU   P2.6
LCD    EQU   P0
RW     EQU   P2.7

       ORG    00H
       sjmp   BASLA
BASLA:
       clr    RW                ; LCD'den okuma modunu iptal et.
       mov    r0,    #07fH      ; Temizleme
temiz: mov    @r0,   #00H       ; Temizleme
       djnz   r0,    temiz      ; Temizleme

       lcall  lcd_ayar          ; LCD'nin Başlangıç Ayarları Yapılıyor.
       mov    dptr,  #Data1
;_____________________________________ 1. Satir ____
       clr    LRS               ;komut girişi
       mov    a,     #83H       ;birinci satır
       lcall  yaz               ;baslangic adresi
       setb   LRS               ;veri girisi
       mov    r0,    #00H
str1:  mov    a,     r0
       movc   a,     @a+dptr
       cjne   a,     #'0',    go1
       sjmp   str2
go1:   lcall  yaz
       inc    r0
       sjmp   str1
;_____________________________________ 2. Satir ____
str2:  clr    LRS               ;komut girişi
       mov    a,#0c1H           ;ikinci satır
       lcall  yaz               ;baslangic adresi
       setb   LRS               ;veri girisi
       mov    dptr,  #Data2
       mov    r0,    #00H
go:    mov    a,     r0
       movc   a,     @a+dptr
       cjne   a,     #'0',    go2
       sjmp   dur
go2:   lcall  yaz
       inc    r0
       sjmp   go
DUR:   SJMP   DUR
;_________________LCD Ayarı_____________________
lcd_ayar:
       clr    LEE
```

```
        clr     LRS                     ;komut girişi
        ;-------------------------------------------------
        ; Minimum 15 ms bekleme
        lcall   gecik
        lcall   gecik
        lcall   gecik
        ;-------------------------------------------------
        ; -------------------------------------------------
        ; LCD'nin Sürülmesi için Gerekli Kod Parçacığı
        ; RS  RW  DB7   DB6   DB5   DB4   DB3   DB2   DB1   DB0
        ; 0   0    0     0     1     1     *     *     *     *   => 30H
        mov     a,      #30H
        lcall   yaz
        lcall   gecik
        mov     a,      #30H
        lcall   yaz
        lcall   gecik
        mov     a,      #30H
        lcall   yaz
        ;-------------------------------------------------
        ;-------------------------------------------------
        ; LCD Ayarları

        ; RS  RW  DB7   DB6   DB5   DB4   DB3   DB2   DB1   DB0
        ; 0   0    0     0     1     1     N     F     *     *
        ; 0   0    0     0     1     1     1     1     *     *   => 3CH =>  N=1 icin 2 satir, F=1 icin 5x10 LCD
        mov     a,      #3cH            ;2 satır, 5x10 pixel
        lcall   yaz

        ; RS  RW  DB7   DB6   DB5   DB4   DB3   DB2   DB1   DB0
        ; 0   0    0     0     0     0     1     D     C     B
        ; 0   0    0     0     0     0     1     1     1     1   => OFH =>  D=1 Ekran Açık, C=1 Kursör Açık, B=1 Blink Açık.
        mov     a,      #0fH            ;Ekran, Kursör ve Blink açık
        lcall   yaz
        mov     a,      #01H            ;Ekranı sil, kursör başa dön.
        lcall   yaz

        ; RS  RW  DB7   DB6   DB5   DB4   DB3   DB2   DB1   DB0
        ; 0   0    0     0     0     0     0     1     I/D   S
        ; 0   0    0     0     0     0     0     1     1     0   => O6H =>  I/D=1 Kursör Arttır, S=0 Ekran Kaydırma Kapalı
        mov     a,      #06H            ;Giriş Modu => Kursör pozisyonunu artır
        lcall   yaz
        ret
;_____________LCD Ayarı Bitti_____________________
yaz:    setb    LEE
        lcall   gecik
        mov     LCD,a     ; LCD = PortO (P0)
        clr     LEE
        ret
;_____________________________________________________
gecik:  mov     r3,     #4fh
w2:     mov     r4,     #0ffh
w1:     djnz    r4,     w1
        djnz    r3,     w2
        ret
;_____________________________________________________

Data1:  DB      'MIKRO LAB.0'
```

```
Data2:  DB     'SAU EEM BOLUMU0'
end
```

Uygulama 2:

```
#include<aduc841.h>
WRITE    EQU    02h          ; 'write page'
VERIFY   EQU    04h          ; 'verify page'
ERASE    EQU    05h          ; 'erase page'
;_____________________________________________________________________________
        ORG     0000h
        JMP     MAIN
MAIN:
        MOV     EADRH,#0          ;3.sayfa seçildi
        MOV     EADRL,#3
        MOV     ECON,#ERASE       ;yazma yapmadan önce ilgili byte silinmeli

        MOV     EDATA1,#0AH       ;

        MOV     ECON,#WRITE       ;3. sayfadaki 1. Byte alanına H'0F yaz

        MOV     ECON,#VERIFY      ;yazma işlemini doğrula..
        MOV     A,ECON
        JNZ     MAIN
END
```

Uygulama 3:

```
#include<aduc841.h>
        LRS    EQU   P2.5
        LEE    EQU   P2.6
        LCD    EQU   P0
        RW     EQU   P2.7


        ORG   00H
        sjmp   BASLA
        ORG   0BH
        LJMP  TIMER0
BASLA:
        clr     RW
        mov     r0,     #07fH
temiz:  mov     @r0,    #00H
        djnz    r0,     temiz
        lcall   lcd_ayar
        mov     dptr,   #Sayi
;_____________________________________________
        MOV     EADRH,     #0h          ;veri hafızanın
        MOV     EADRL,     #3h          ;3.sayfası seçildi
        MOV     ECON,      #01          ;3.sayfadaki ilk Byte'in içeriği okundu
        MOV     R1,    EDATA1           ;ve R1 saklayıcısına aktarıldı..
;_____________________________________________
        clr     LRS                     ;LCD için komut girisi
        mov     a,      #80H            ;başlangıç adresi
```

```
        lcall   yaz
        setb    LRS                     ;LCD için veri girisi
;_____________________________________________________
        MOV  TMOD,      #081H
        MOV  TH0,  #00          ;Timer0 ayarlandı
        MOV  TL0,  #00
        MOV   R0,   #84d        ;500ms=5.9ms* 84
        SETB  EA
        SETB  ET0
        SETB  TR0               ;Timer0 start
;_____________________ _____________________________
        MOV     A,  R1          ;Veri hafizadan okunan deger R1 de tutuluyordu..
        MOVC    A,  @A+DPTR     ;ilk değeri
        LCALL   YAZ             ;başlangıç adresine yaz..

DUR:  CJNE    R0,  #0,     DUR  ;500ms doldumu?
        MOV     R0,  #84d       ;doldu..
        DEC     R1              ;değeri 1 azalt
        MOV     A,   R1
        MOVC  A,   @A+DPTR      ;LCD ya yaz..
        LCALL YAZ
        CJNE    R1,  #0,     DUR
STOP: SJMP  STOP
;_____________________________________________________
TIMER0:   dec       R0
          reti



;__________________LCD Ayarı_____________________
lcd_ayar:
        clr     LEE
        clr     LRS                     ;komut girişi
        ;-----------------------------------------------
        ; Minimum 15 ms bekleme
        lcall   gecik
        lcall   gecik
        lcall   gecik
        ;-----------------------------------------------
        ; -----------------------------------------------
        ; LCD'nin Sürülmesi için Gerekli Kod Parçacığı
        ; RS  RW  DB7   DB6   DB5   DB4   DB3   DB2   DB1   DB0
        ; 0   0    0      0     1     1     *     *     *     *   => 30H
        mov     a,      #30H
        lcall   yaz
        lcall   gecik
        mov     a,      #30H
        lcall   yaz
        lcall   gecik
        mov     a,      #30H
        lcall   yaz
        ;-----------------------------------------------
        ;-----------------------------------------------
        ; LCD Ayarları

        ; RS  RW  DB7   DB6   DB5   DB4   DB3   DB2   DB1   DB0
```

```
        ; 0   0    0     0     1    1     N    F    *    *
        ; 0   0    0     0     1    1     1    1    *    *   => 3CH =>  N=1 icin 2 satir, F=1 icin 5x10 LCD
        mov   a,      #3cH          ;2 satır, 5x10 pixel
        lcall yaz

        ; RS  RW  DB7  DB6  DB5  DB4  DB3  DB2  DB1  DB0
        ; 0   0    0     0     0    0     1    D    C    B
        ; 0   0    0     0     0    0     1    1    1    1  => OFH =>  D=1 Ekran Açık, C=1 Kursör Açık, B=1 Blink Açık.
        mov   a,      #0fH          ;Ekran, Kursör ve Blink açık
        lcall yaz
        mov   a,      #01H          ;Ekranı sil, kursör başa dön.
        lcall yaz

        ; RS  RW  DB7  DB6  DB5  DB4  DB3  DB2  DB1  DB0
        ; 0   0    0     0     0    0     0    1    I/D  S
        ; 0   0    0     0     0    0     0    1    1    0  => O6H =>  I/D=1 Kursör Arttır, S=0 Ekran Kaydırma Kapalı
        mov   a,      #06H          ;Giriş Modu => Kursör pozisyonunu artır
        lcall yaz
        ret
;_____________LCD Ayarı Bitti_____________________

yaz:    setb  LEE
        lcall gecik
        mov   LCD,a
        clr   LEE
        ret
;_____________________________________________
gecik:  mov   r3,     #4fh
w2:     mov   r4,     #0ffh
w1:     djnz  r4,     w1
        djnz  r3,     w2
        ret

;_____________________________________________
Sayi:   DB    '0123456789ABCDEF'

end
```