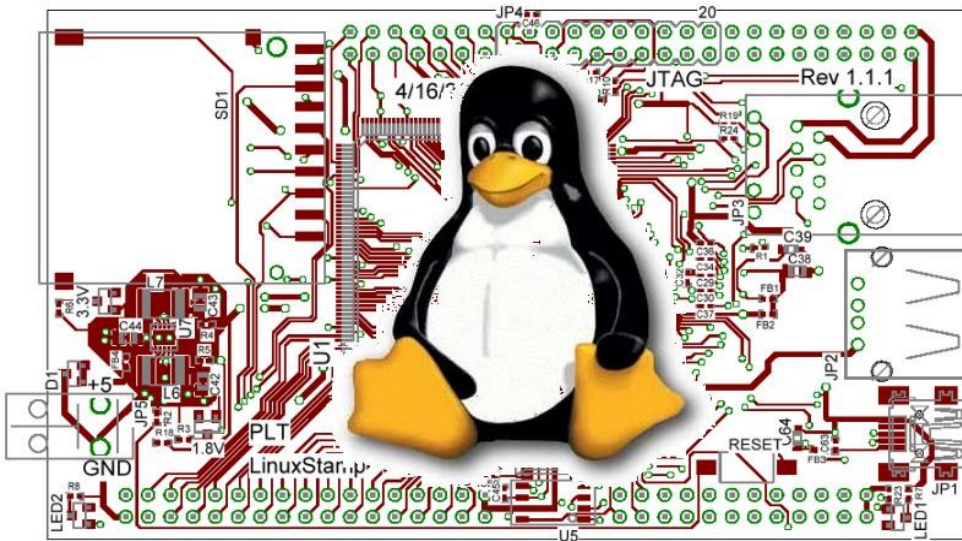


# Gömülü Sistemler ve Linux

Örnek Uygulama:  
Navigasyon Cihazı Tasarımı



Serdar KÖYLÜ

Özgür Yazılım ve Linux Günleri 2011

## Gömülü Sistem Nedir ?

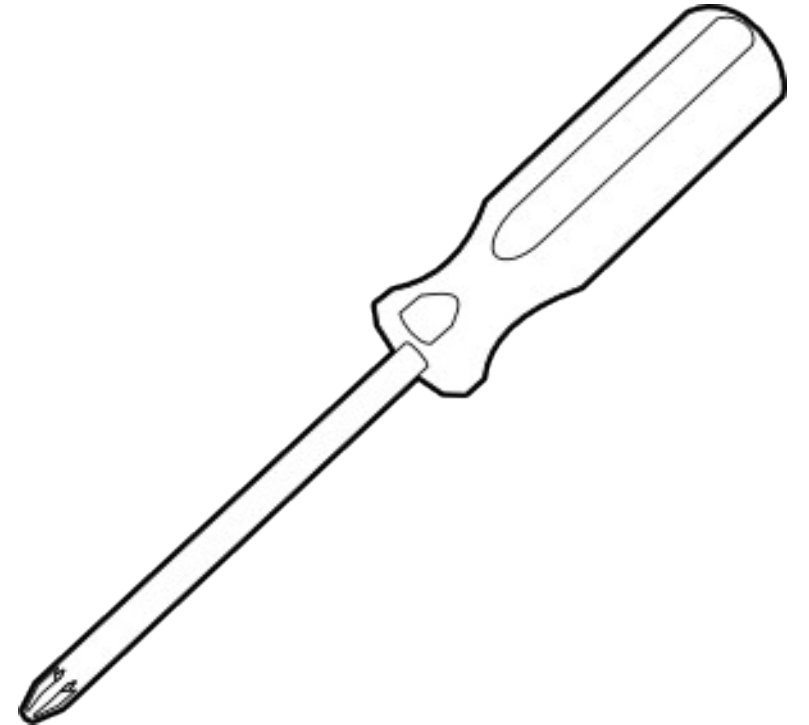
---

Gömülü sistem, bilgisayarın kendisini kontrol eden cihaz tarafından içerildiđi özel amaçlı bir sistemdir.

Genel maksatlı, örneđin **kişisel bilgisayar** gibi bir bilgisayardan farklı olarak, gömülü bir sistem kendisi için önceden özel olarak tanımlanmış görevleri yerine getirir.

# Gömülü Sistemlerin Avantaj / Dezavantajları

---



# Gömülü Sistemlerin Avantaj / Dezavantajları

---



# Ne yapacağız?

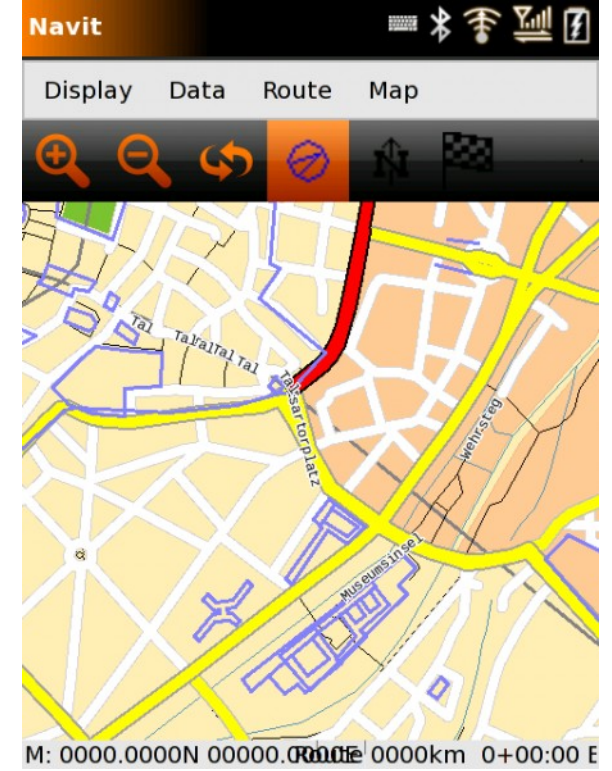
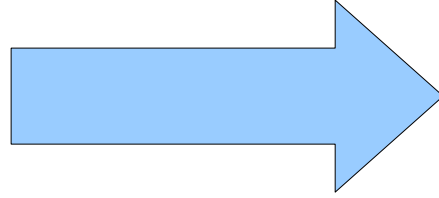
Aradığımız yeri bulan, gösteren bir alet. Bir ekranı olsun, ekranda yerimizi ve yolumuzu gösterebilir.

Navigasyon Cihazı..

# İçin Neler Gerekir?

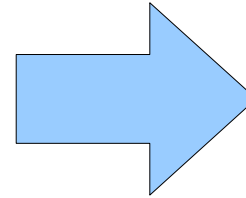
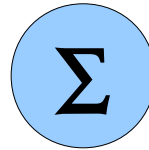
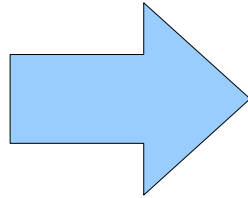


Konum Bilgisi  
İstikamet bilgisi  
Coğrafya  
???



Sensörler

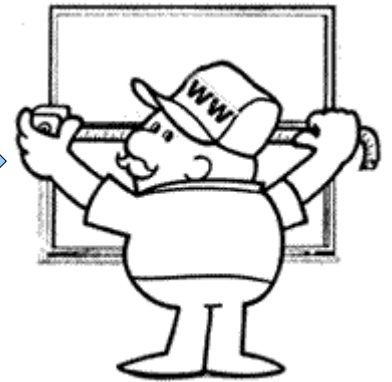
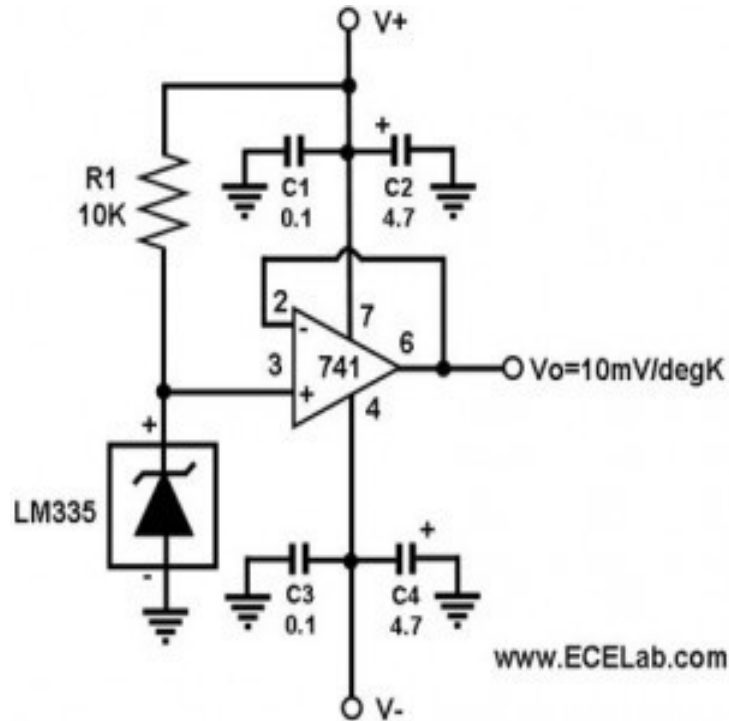
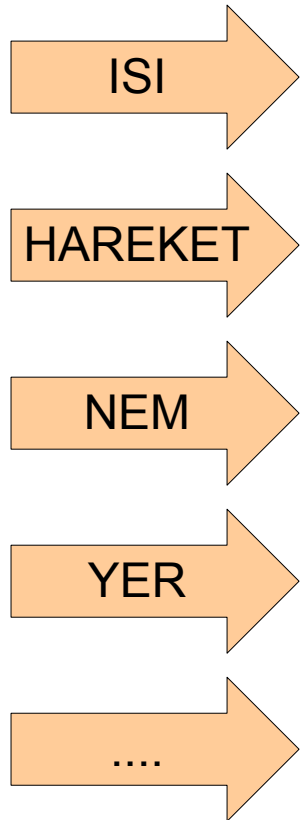
Bilinenler



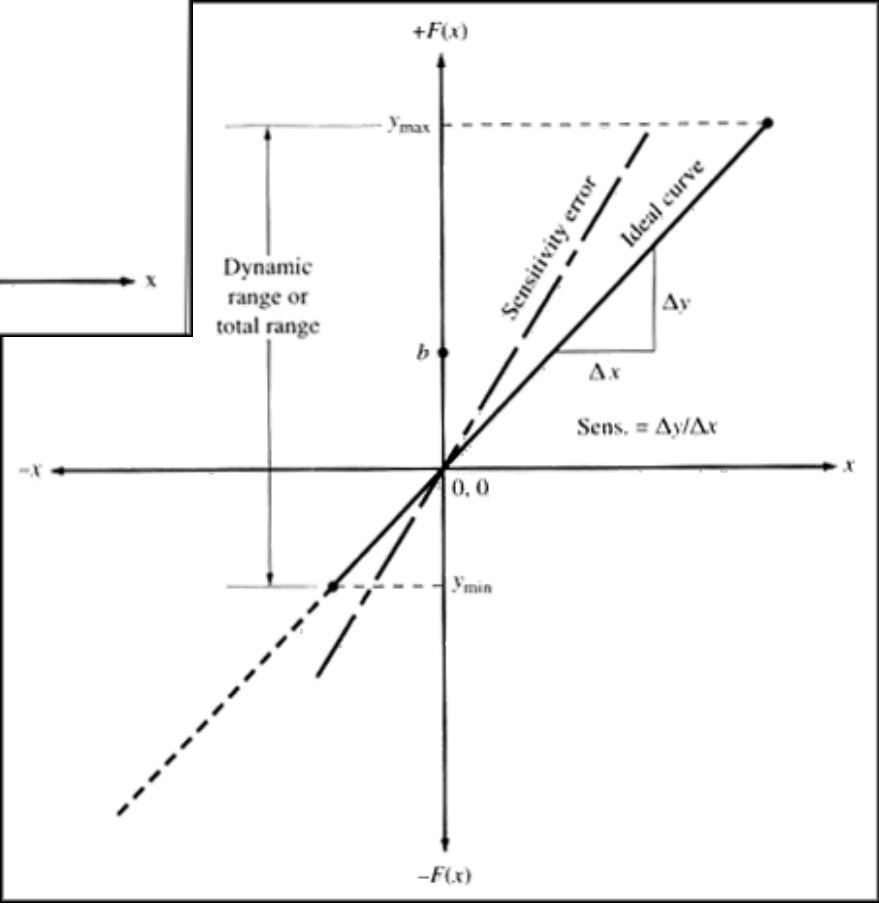
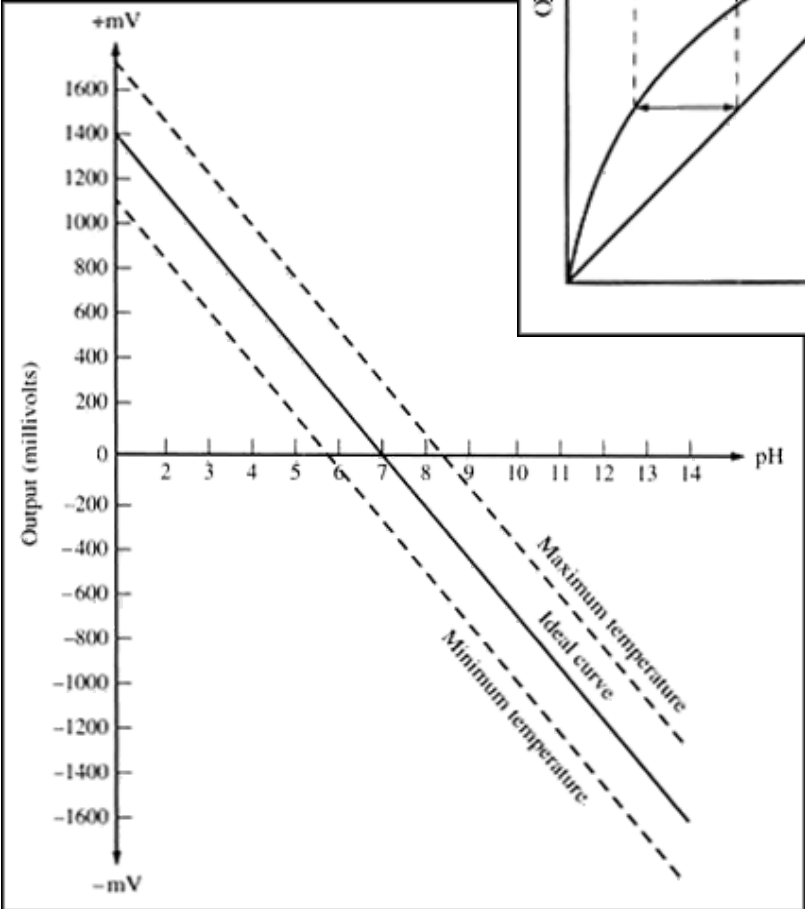
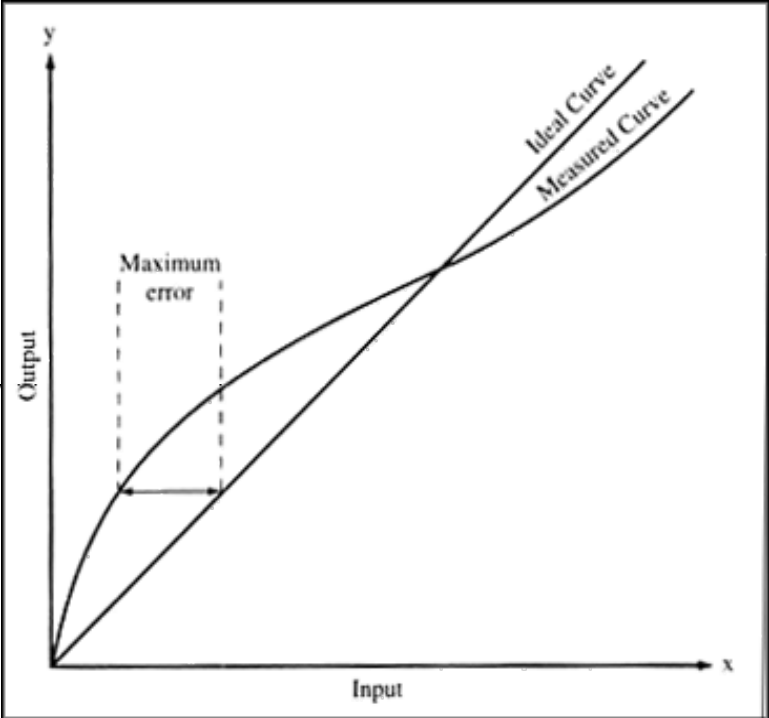
Sonuçlar

Sunum

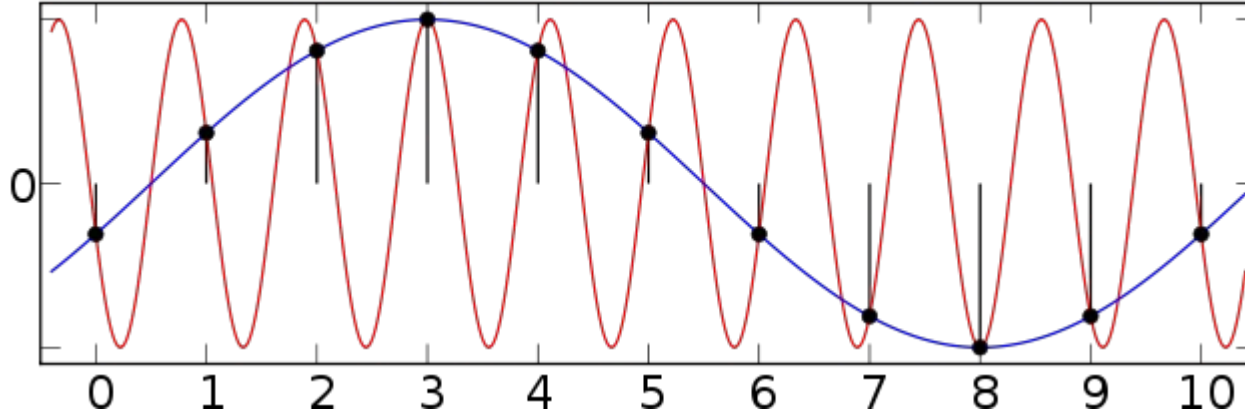
# Gömülü Sistem Tasarımı – Sensörler



# Doğrusallık (Linearity)



## Örnekleme hızı



## Veri iletim yöntemi

**Sorgulama (Polling), Kesme (Interrupt)**

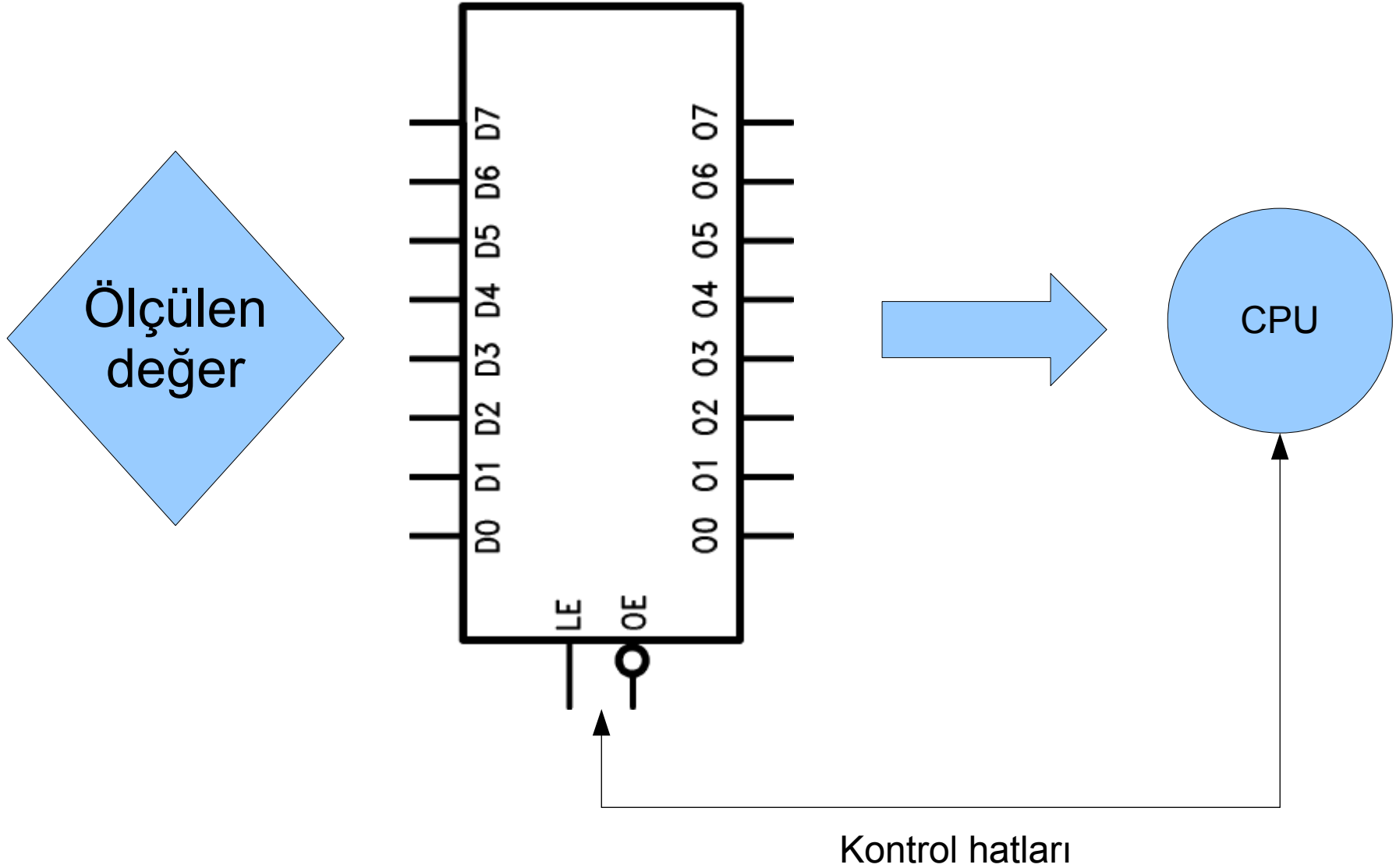
**Arabirim (Interface)**

Paralel Arabirimler

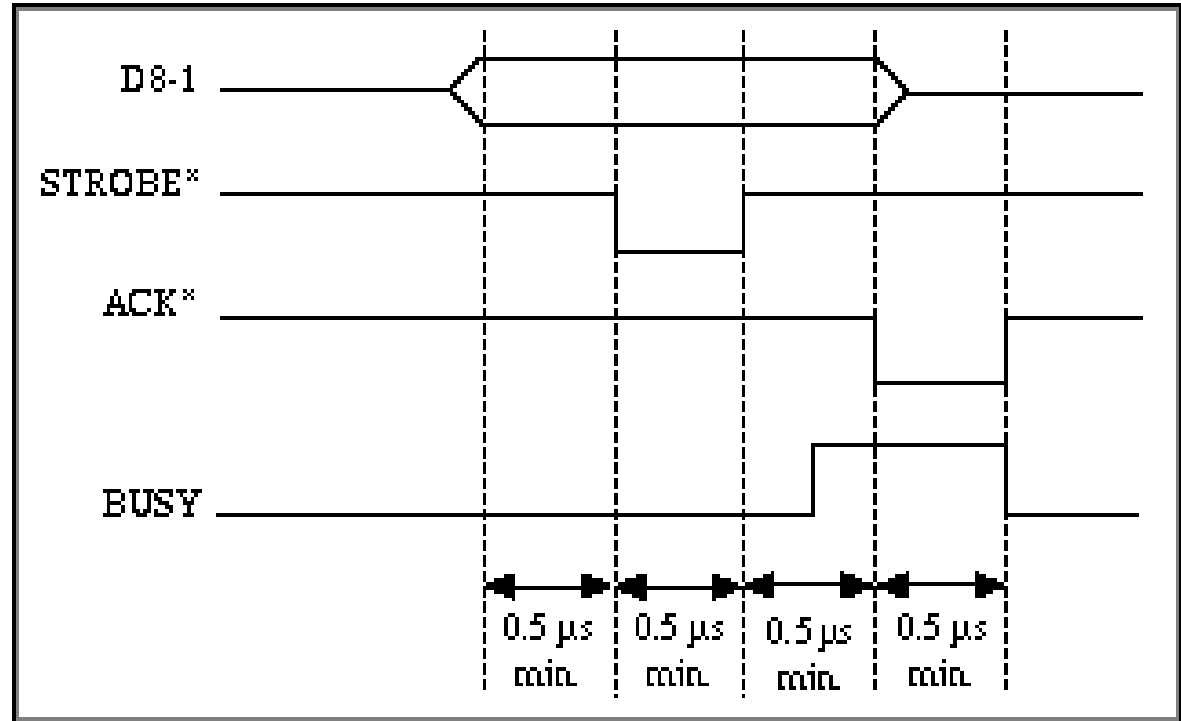
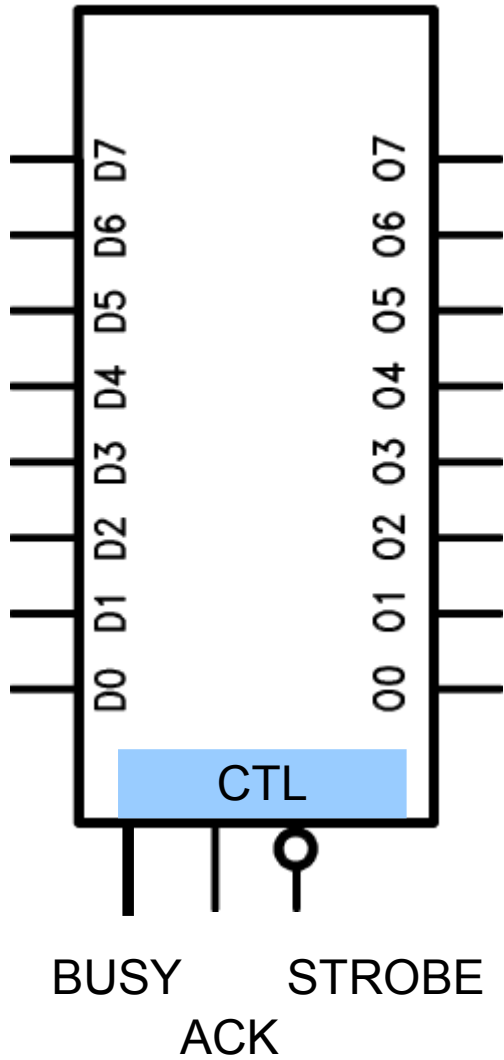
Seri Arabirimler

*USB, UART, SPI, I<sup>2</sup>C*

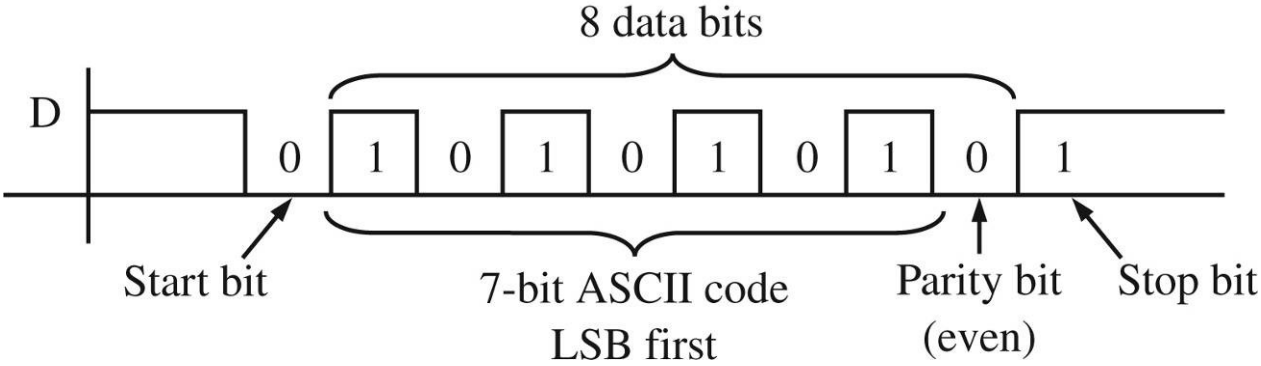
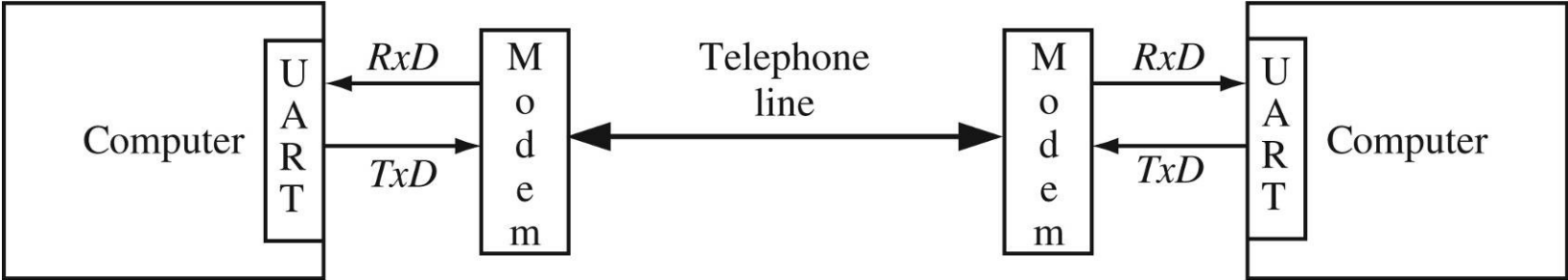
## Paralel Arabirimler



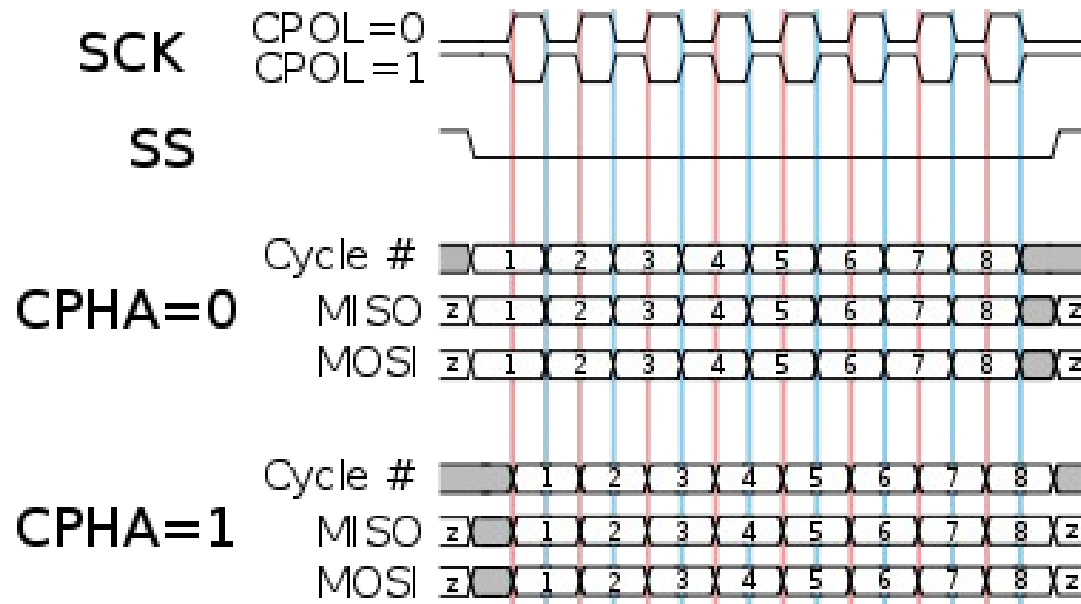
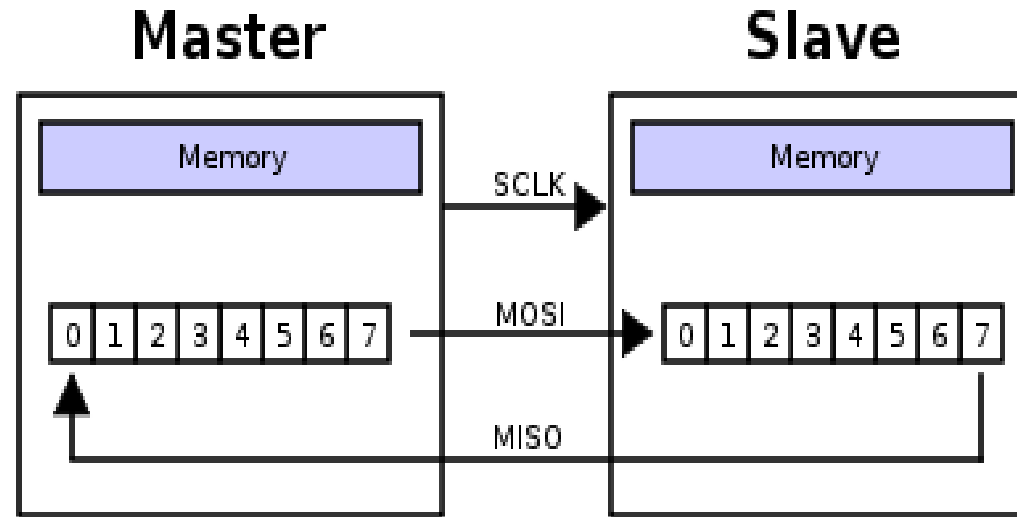
## Paralel Arabirimler



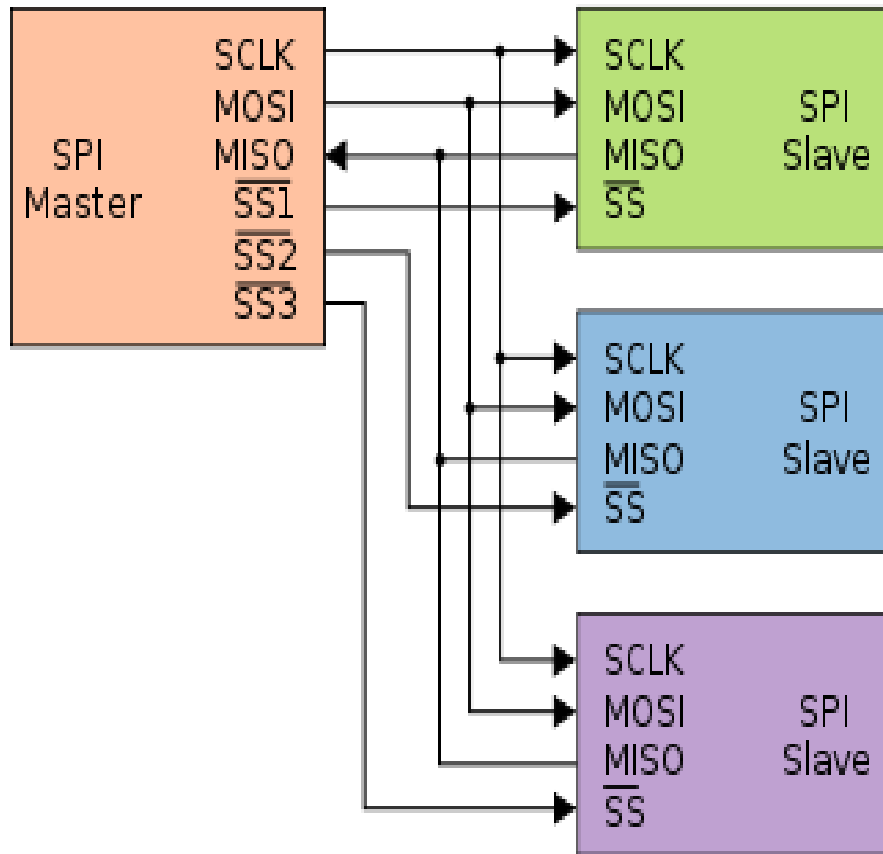
# USB, UART



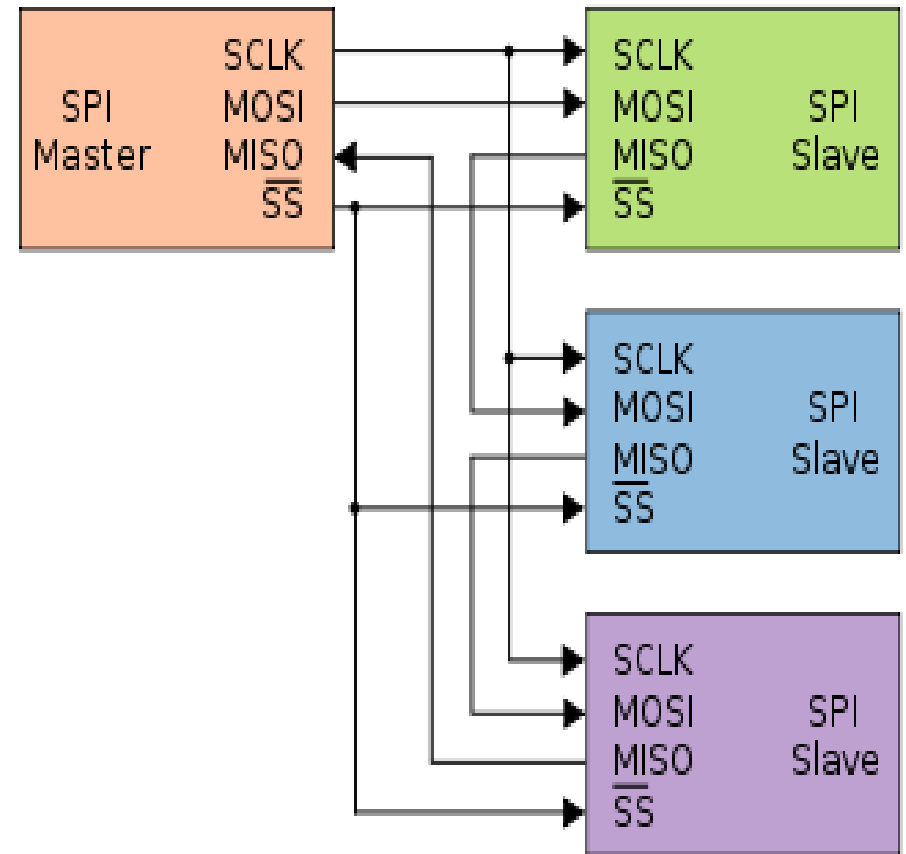
# SPI (SSP) - Serial Peripheral Interface



# SPI (SSP) - Serial Peripheral Interface



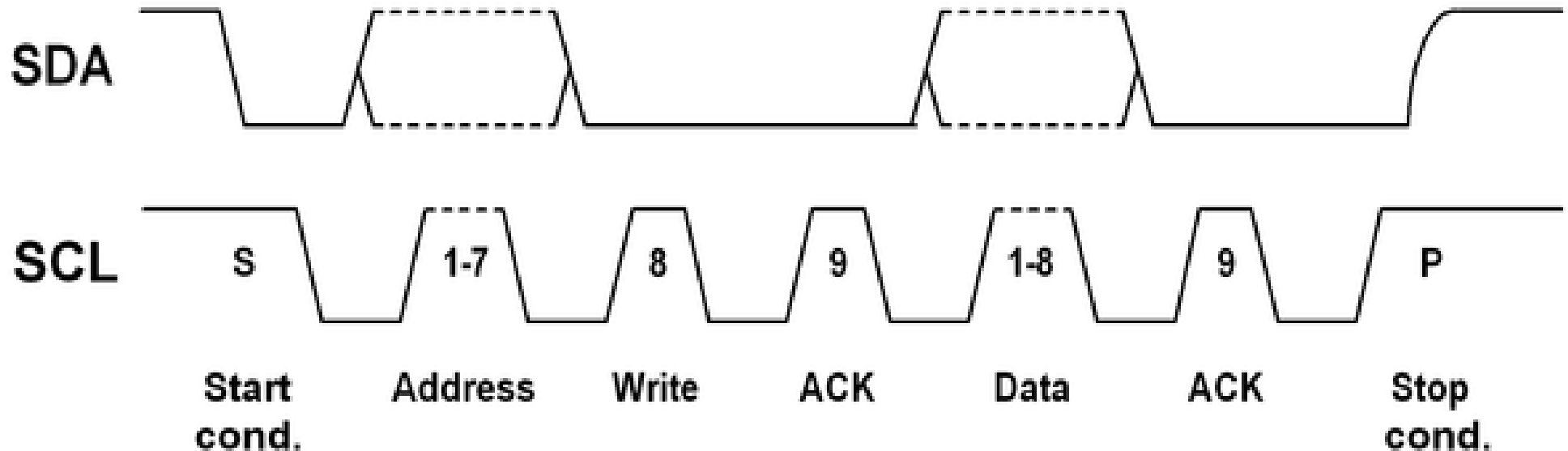
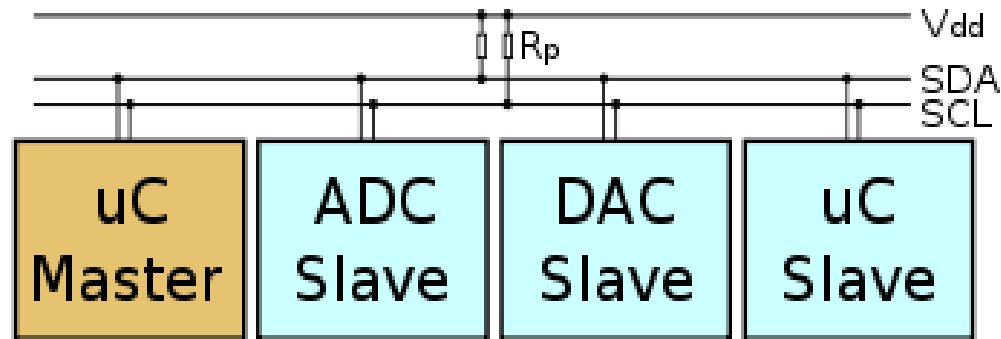
Independent



Daisy Chain

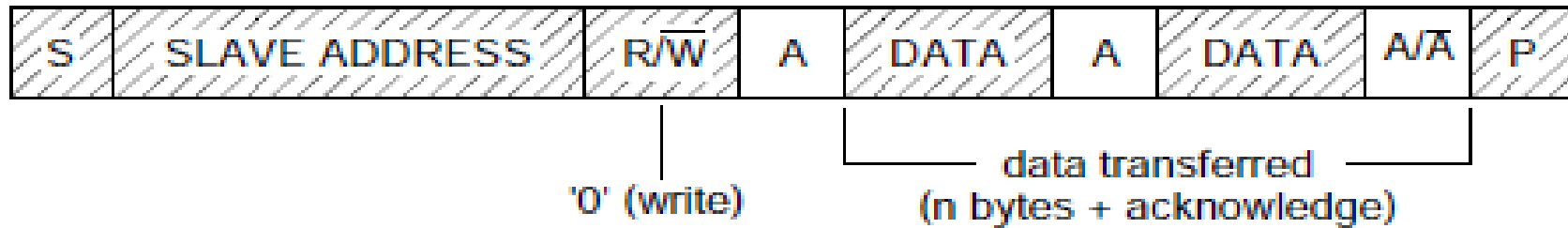


## I<sup>2</sup>C Inter – Integrated Circuit (2-Wire)





## I<sup>2</sup>C Inter – Integrated Circuit (2-Wire)



 from master to slave

 from slave to master

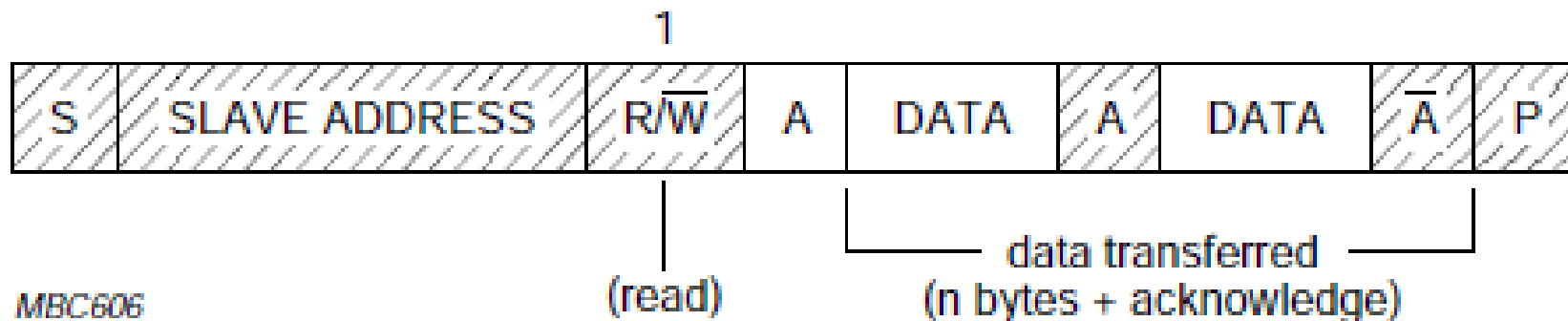
A = acknowledge (SDA LOW)

$\bar{A}$  = not acknowledge (SDA HIGH)

S = START condition

P = STOP condition

MBC605



MBC606

## Diğer hususlar

- *Maliyet*
- *Boyutlar*
- *Güç Tüketimi*
- *Uygulama kolaylığı*
- *Sensöre özel kriterler*

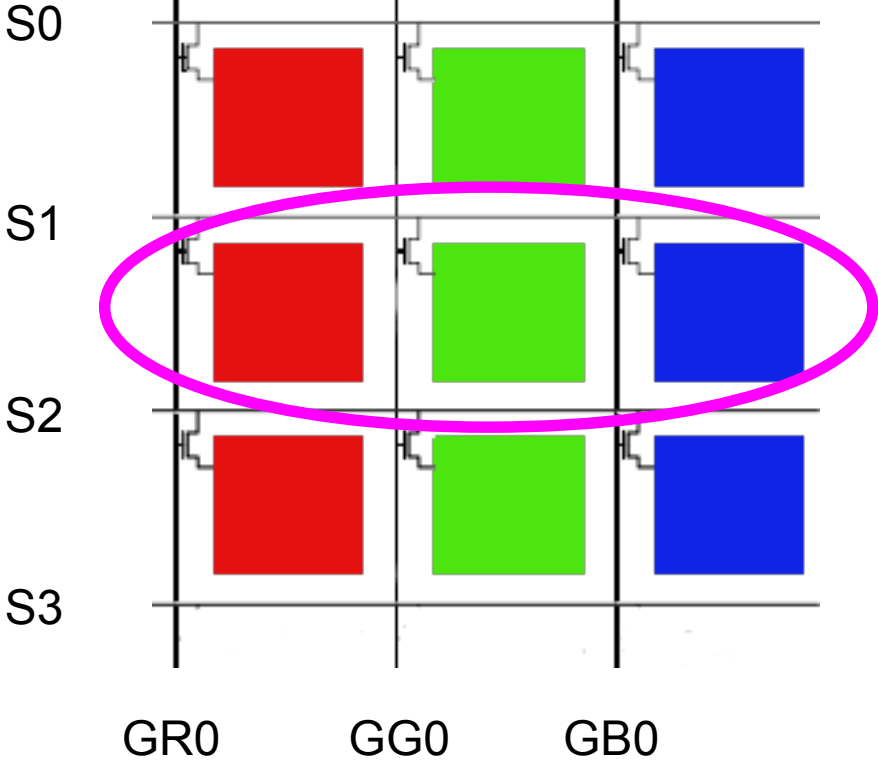
## LCD Paneller



- ✓ Hitachi HD44780 Uyumlu Paralel arabirim
- ✓ RS232/UART uyumlu harici bağlantılar
- ✓ SPI, I2C

# LCD Paneller - TFT

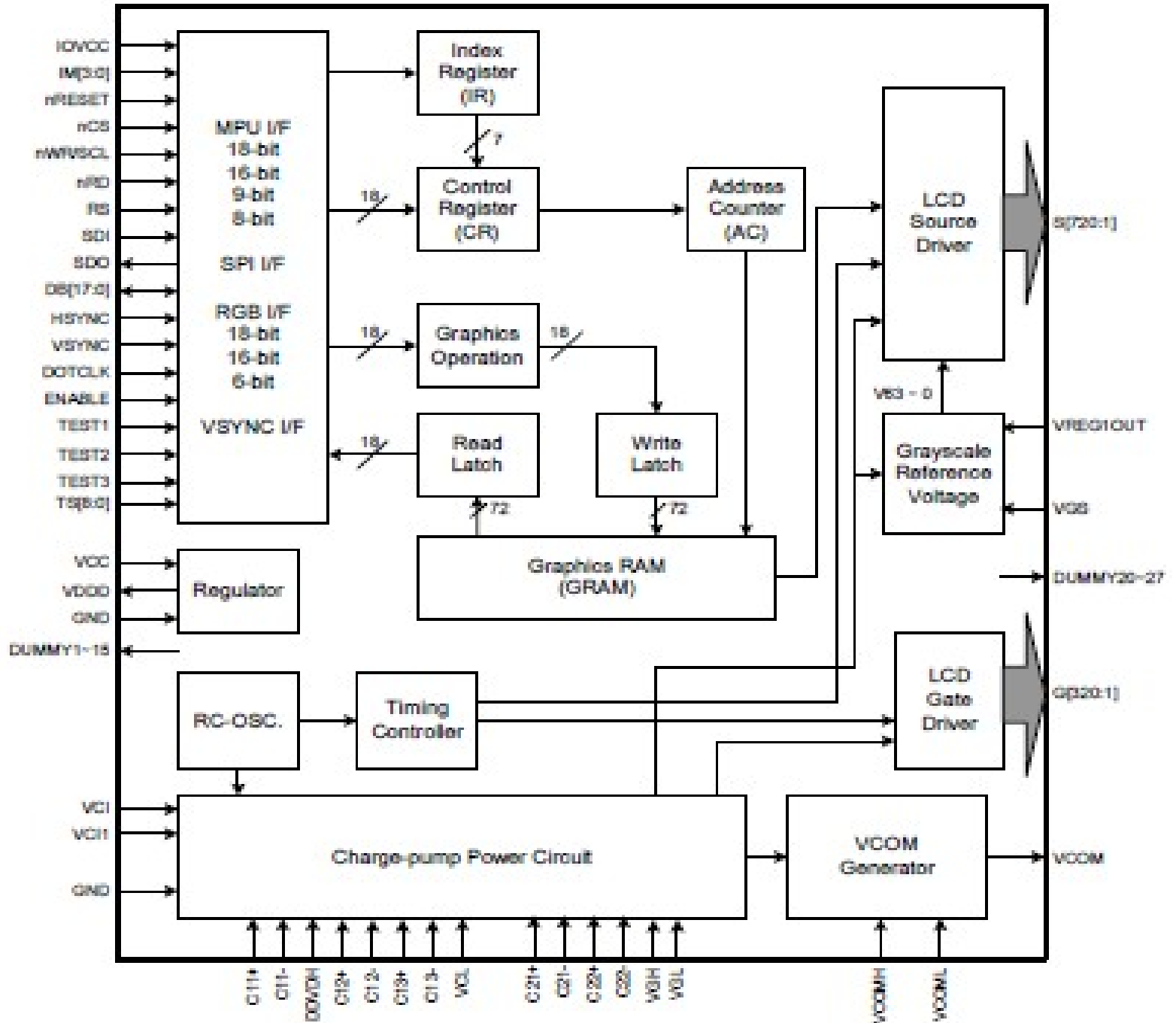
320 Source Hattı



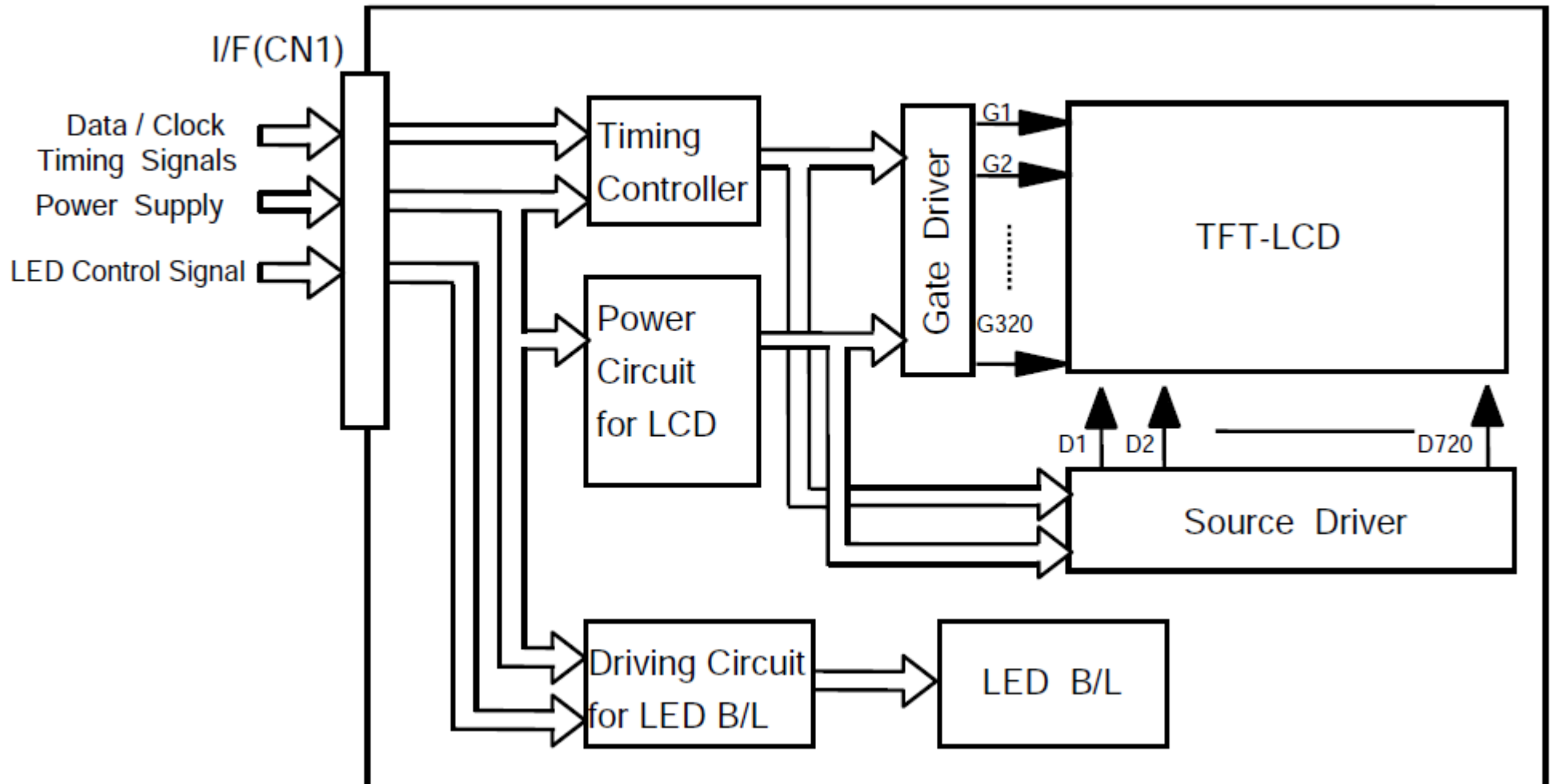
240 x 3 = 720 Gate Hattı

# Gömülü Sistem Tasarımı – LCD w/ Controller

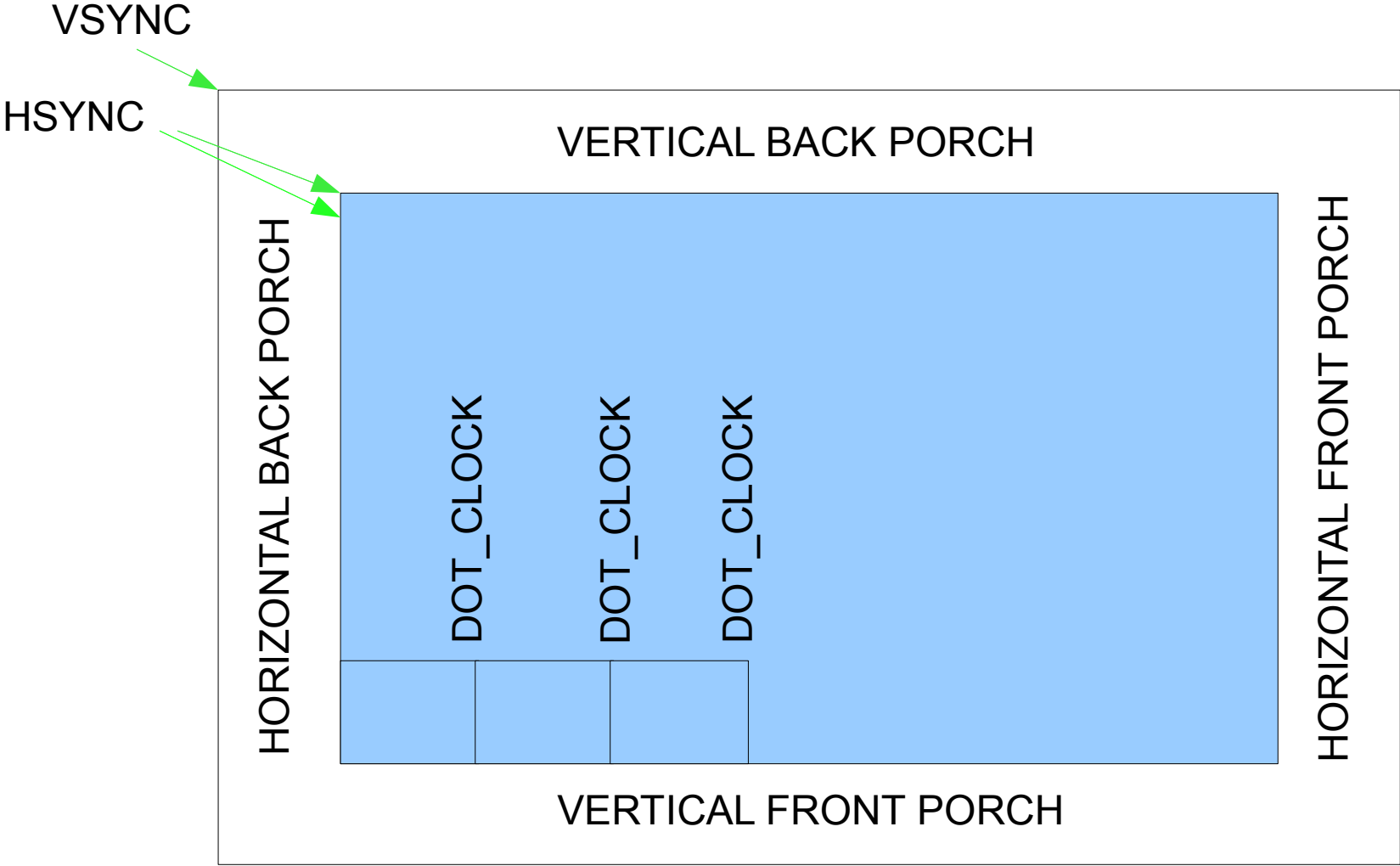
## ILI9325 320xRGBx240 LCD kontrolcüsü



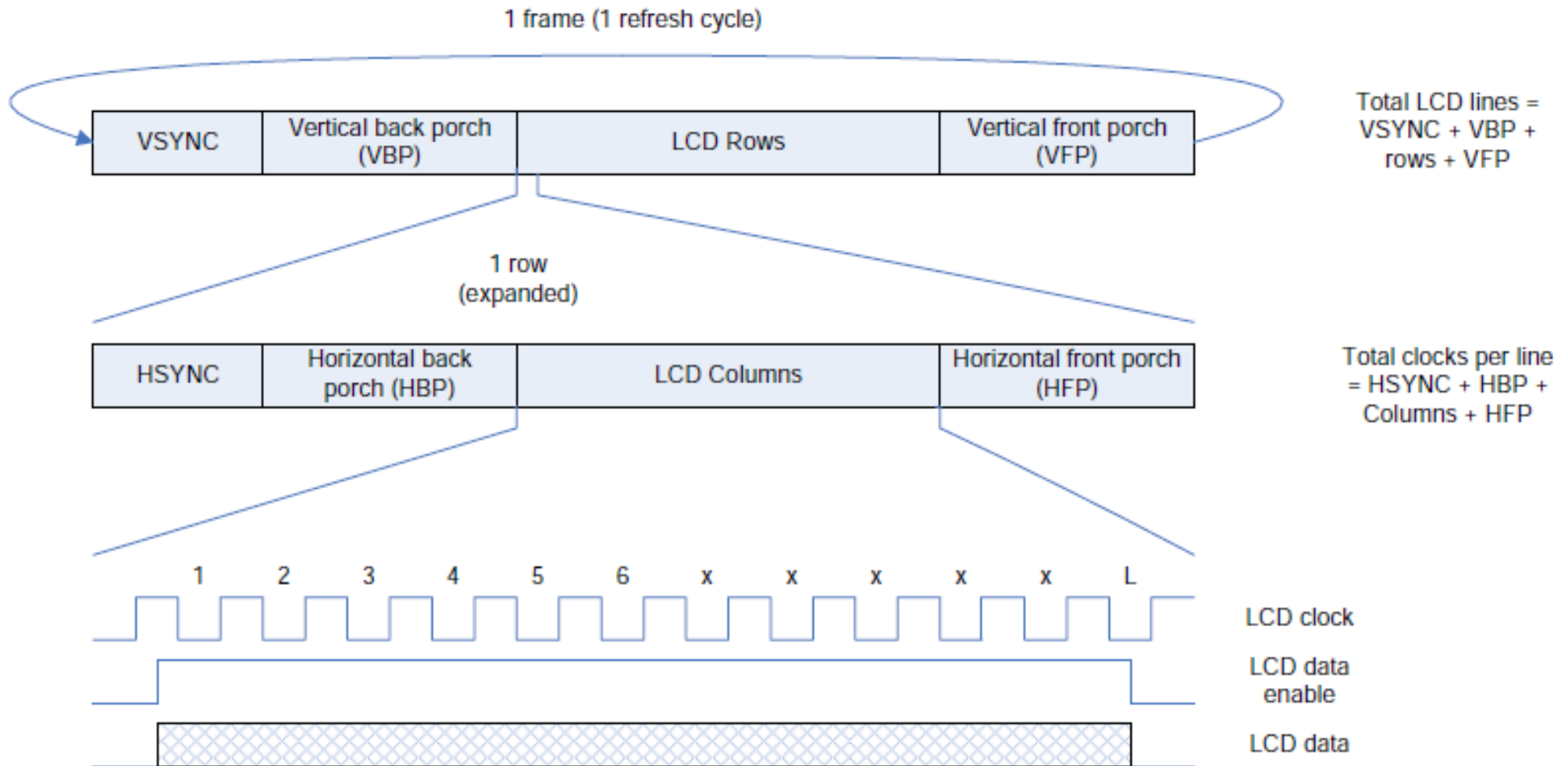
## LCD Paneller - TFT



# LCD Panel Zamanlaması

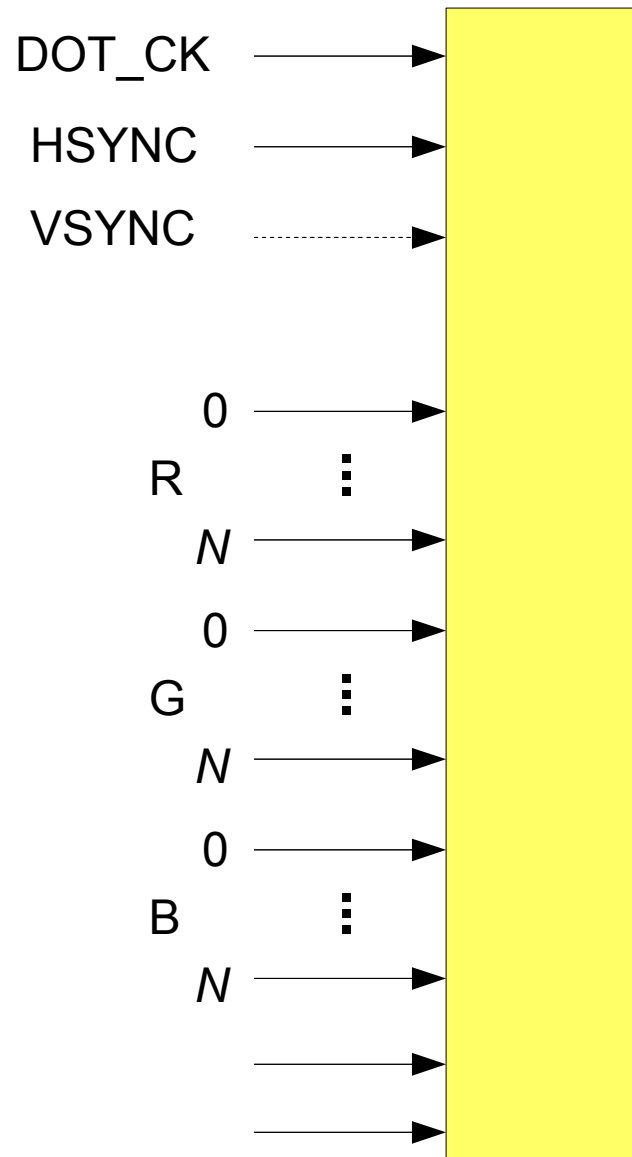


# LCD Panel Zamanlaması



$$\text{Total clocks per frame} = \text{Total LCD lines} * \text{Total clocks per line}$$

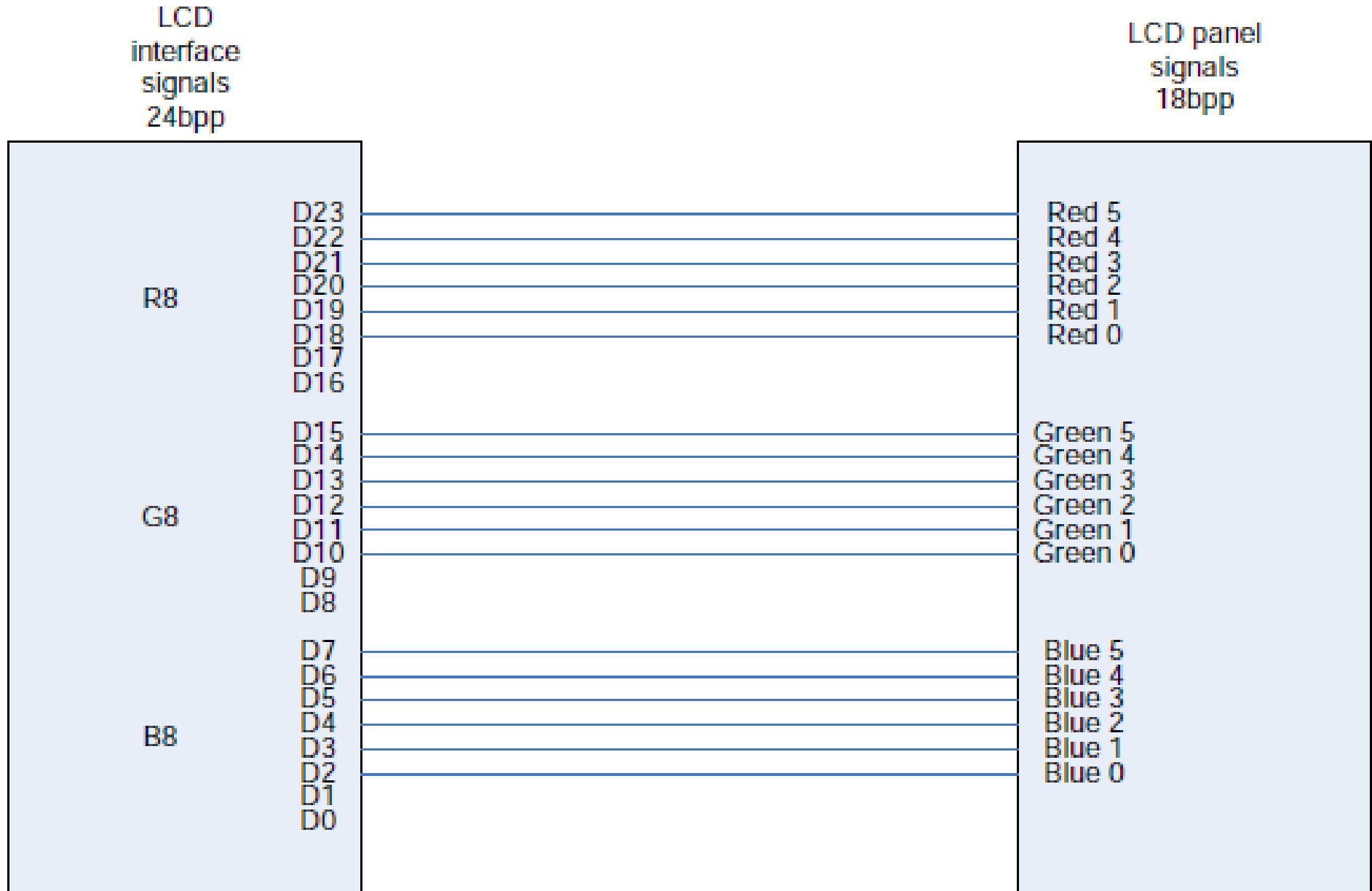
## LCD Panel Arabirimi - RGB



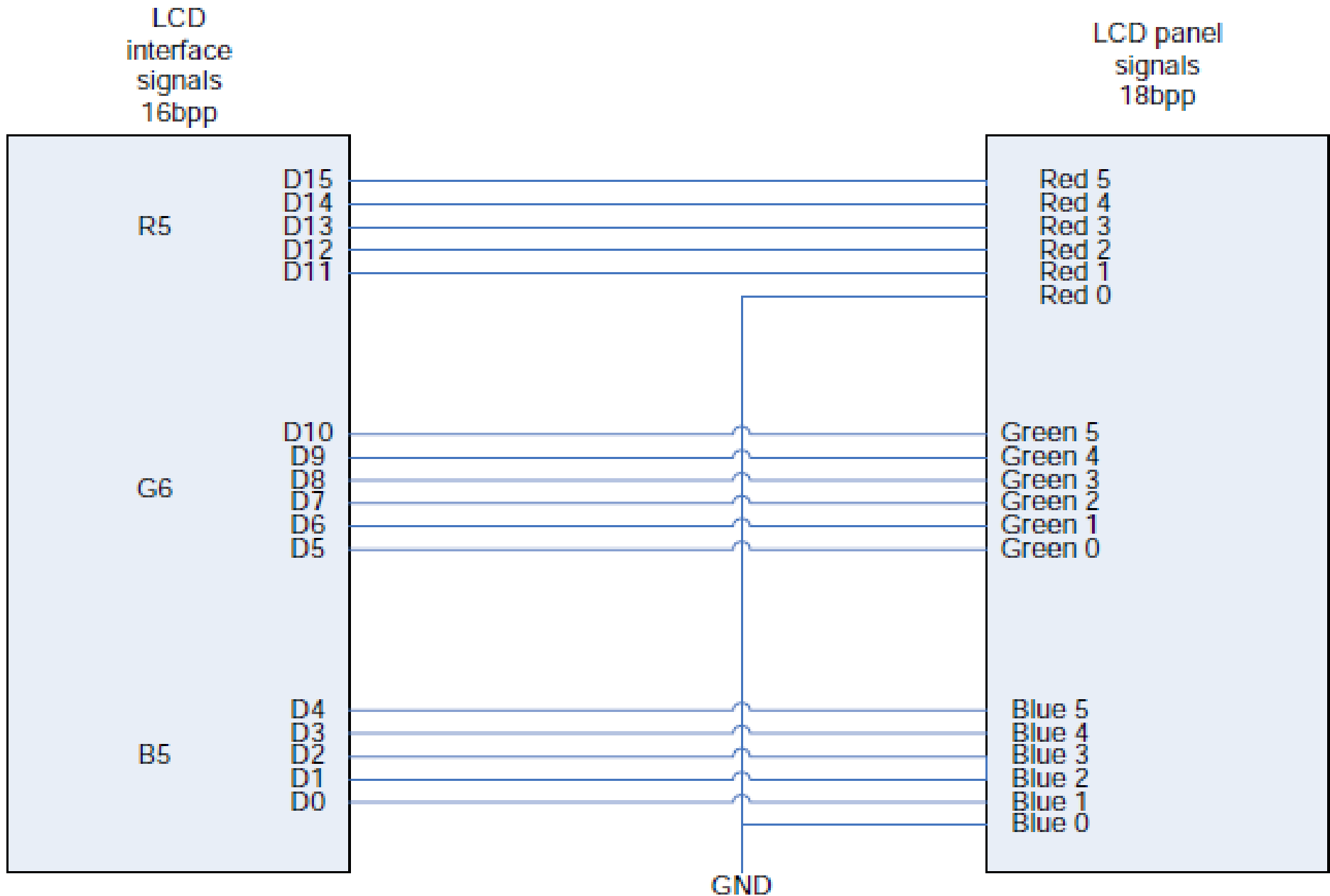
$RGBR_nG_nB_n$

- RGB555 - 15bpp
- RGB565 - 16bpp
- RGB666 - 18bpp
- RGB888 - 24bpp

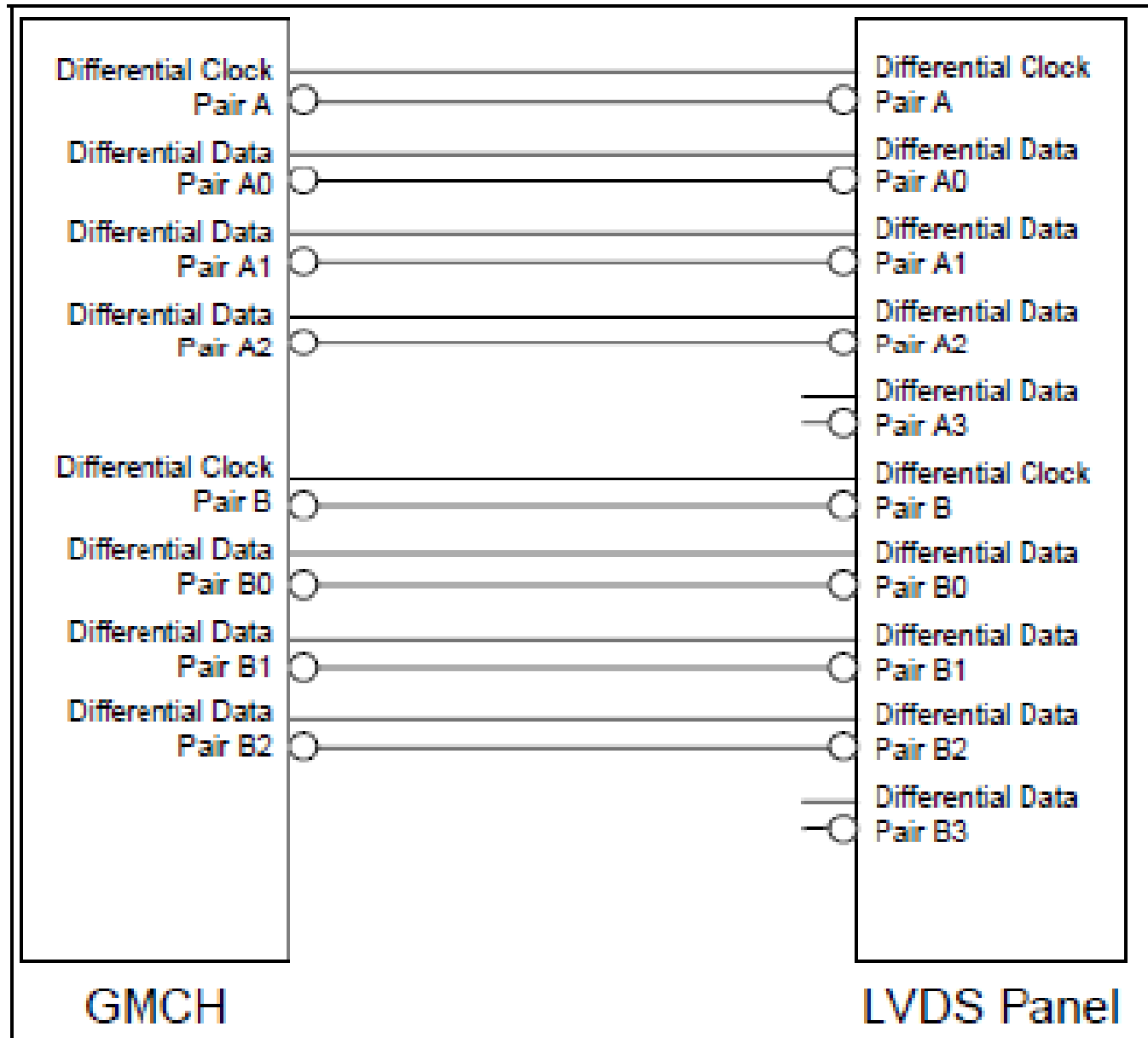
# LCD Panel Arabirimi



# LCD Panel Arabirimi

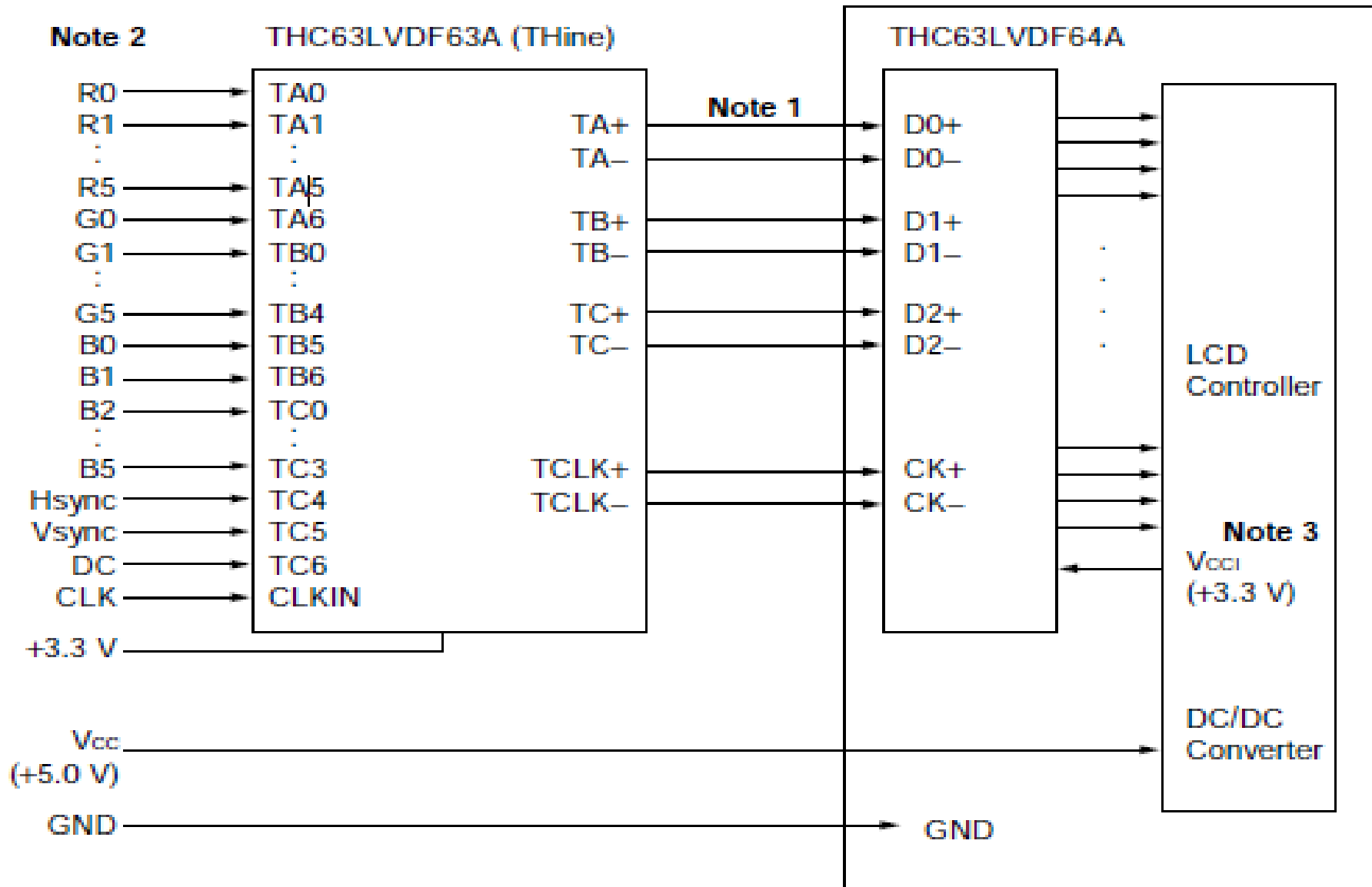


## LCD Panel Arabirimi - LVDS

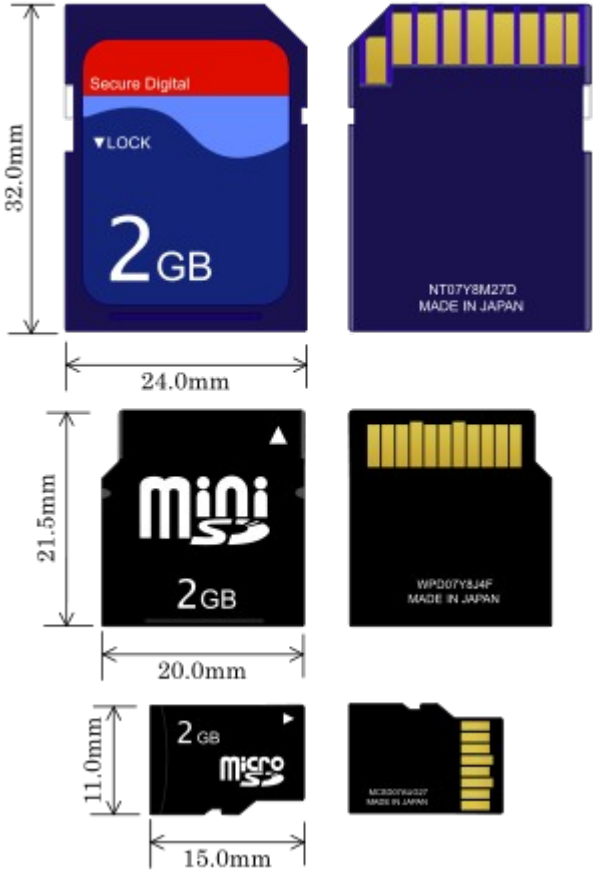


# LCD Panel Arabirimi - LVDS

NL10276BC30-04D



# MTD – Memory Technology Devices



Flash ROM

## Flash ROM Karakteristikleri

- ✓ Üretimde tüm bitler “1” olarak hazırlanabilir.
- ✓ Yazma işlemi sadece “1” olan bitleri “0” olarak değiştirmedir.
- ✓ “0” olan bitler tümü birden özel silme işlemiyle “1” yapılabilir.
- ✓ Yazma sayısı sınırlıdır, 10.000, 100.000 ....
- ✓ Bir süre sonra kendiliğinden bozulur.

## NOR Flash

Kelime (Byte, Word) bazında okuma/yazma erişimi  
Bloklar halinde silme işlemi.

## NAND Flash

Sayfa ve blok bazında okuma/yazma/silme işlemleri  
Hata tespit ve giderme için ayrılmış alanlar.

## Karşılaştırma

	<b>NOR</b>	<b>NAND</b>
Arabirim	Bus	I/O
Yoğunluk	Düşük	Yüksek
Fiyat	Yüksek	Düşük
Okuma	Hızlı	Yavaş
Tek kelime yazma	Hızlı	Yavaş
Blok yazma	Yavaş	Hızlı
Silme	Yavaş	Hızlı
Güç tüketimi	Yüksek	Düşük
Lüzumsuz işyükü.	Yok gibi.	Fazla.
0 km.de bozuk alanlar	Yok	Olabilir.

## Diğer türevler

SDCARD, MMC, SDHC, SDX.....

*Kompakt yapı, taşınabilir.*

OneNAND

*NOR benzeri NAND Flash.*

*Senkron erişim, farklı hardware, NOR software*

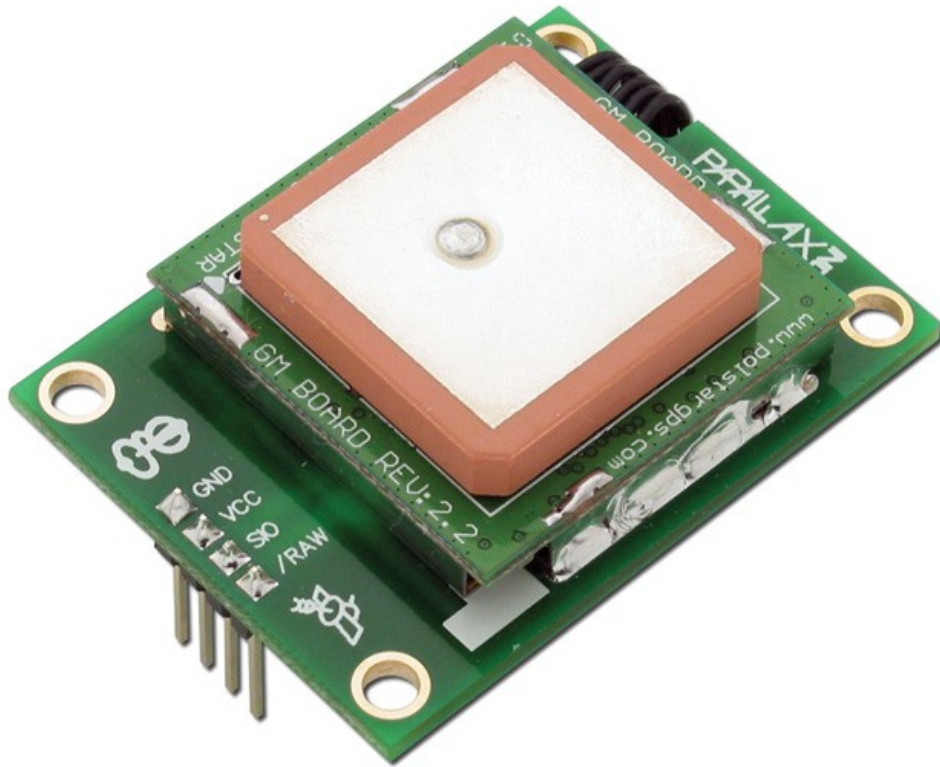
*Özel mimari, yüksek hız, dahili ECC vs.*

MovieNAND

*MMC Arabirimi, chip kılıfında.*

*Yüksek kapasite ve hız.*

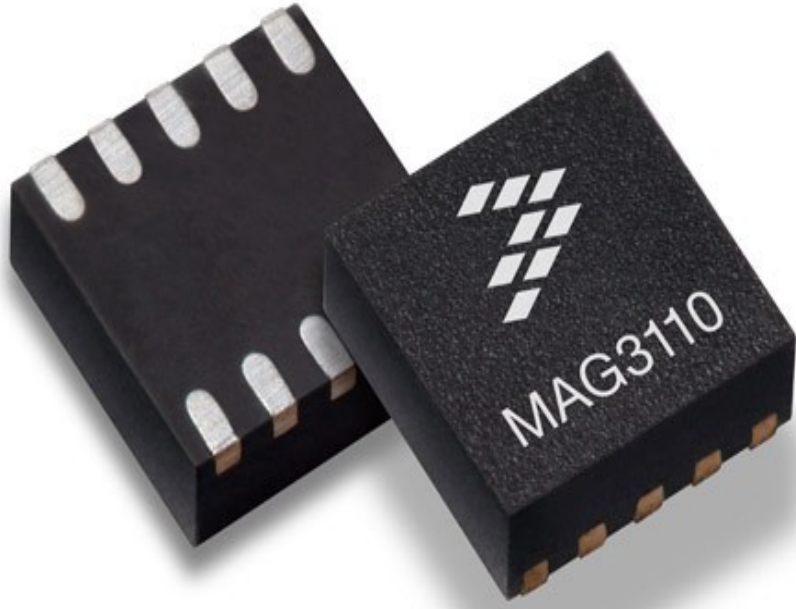
## GPS Modülü



SIRF StarII GPS chipset  
Passive patch antenna

Asynchronous serial,  
4800 bps @ TTL Level

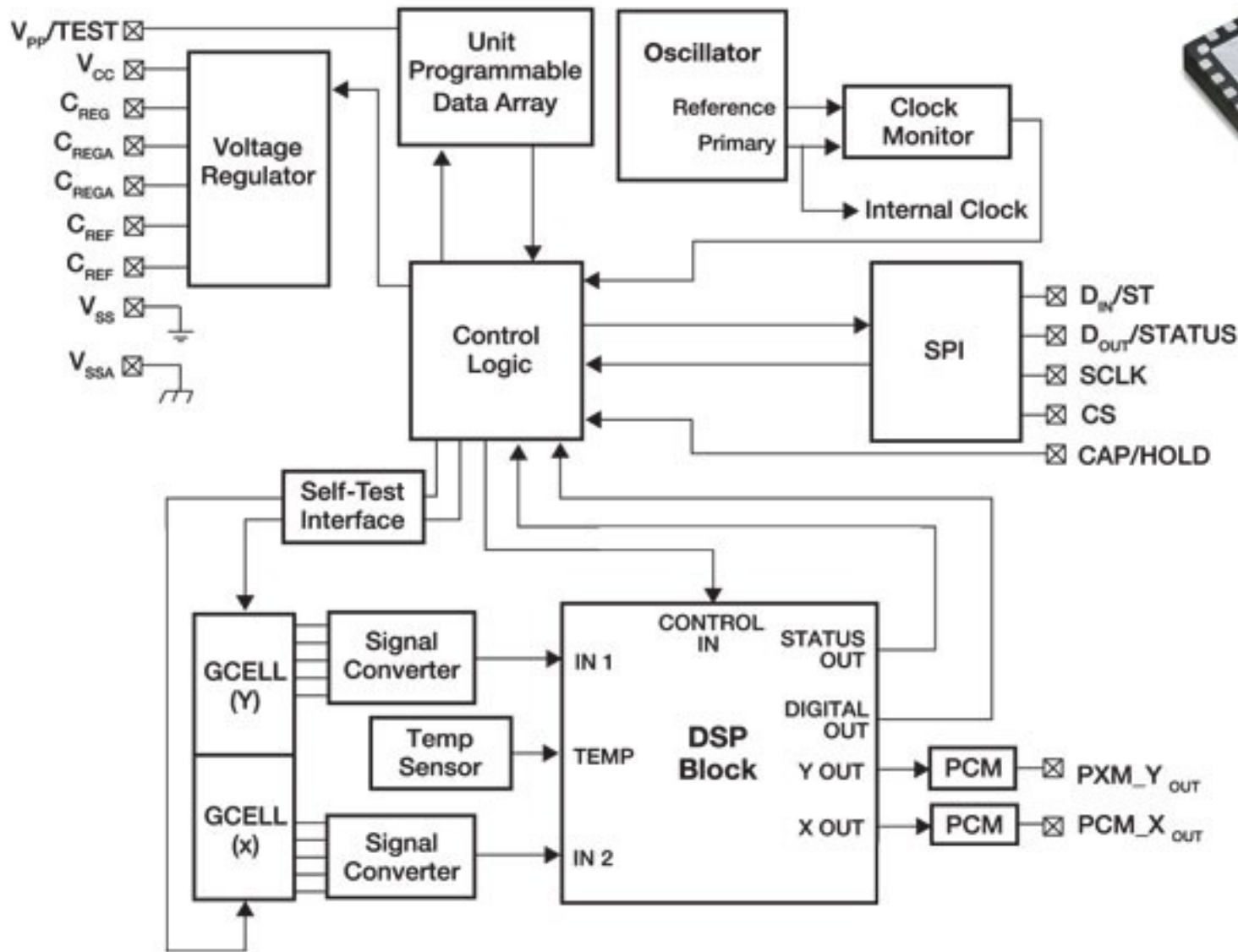
## Pusula için Manyetometre



I<sup>2</sup>C interface 400kHz

# Pusula için Hızlanma sensörü

MMA6900Q Block Diagram

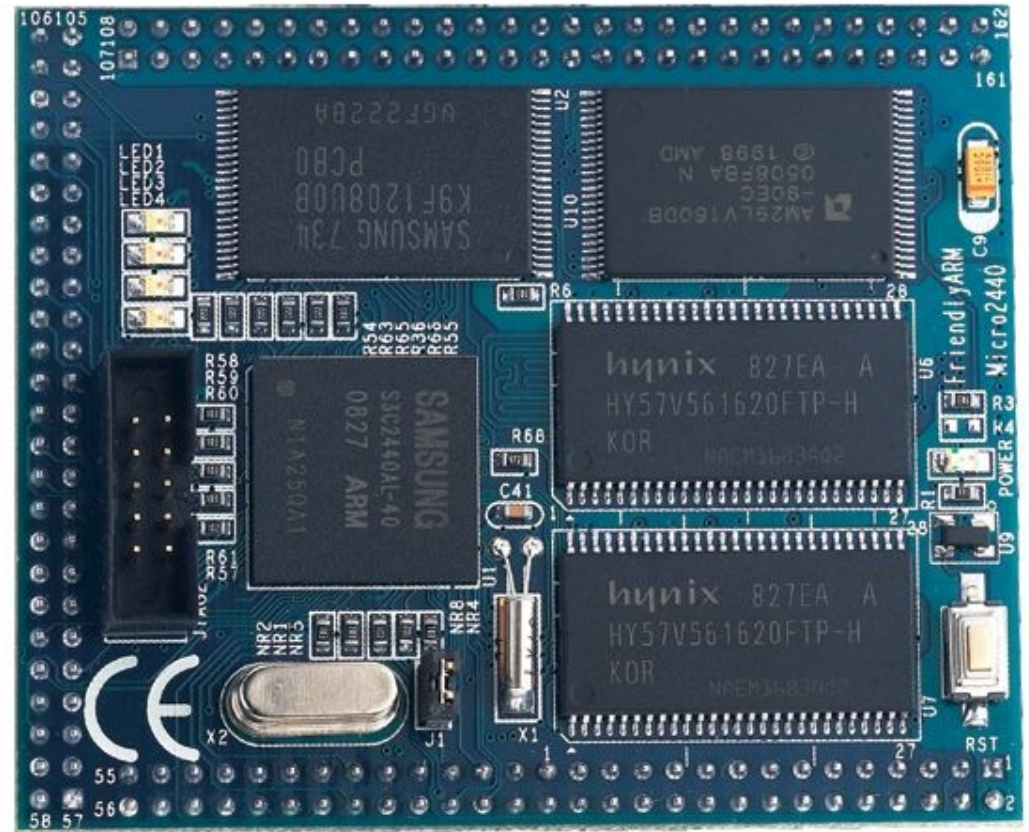


SPI  
Interface

## LCD Panel, CPU, Storage



3.5" RGB Ifc., TouchScreen  
NEC (compatible)

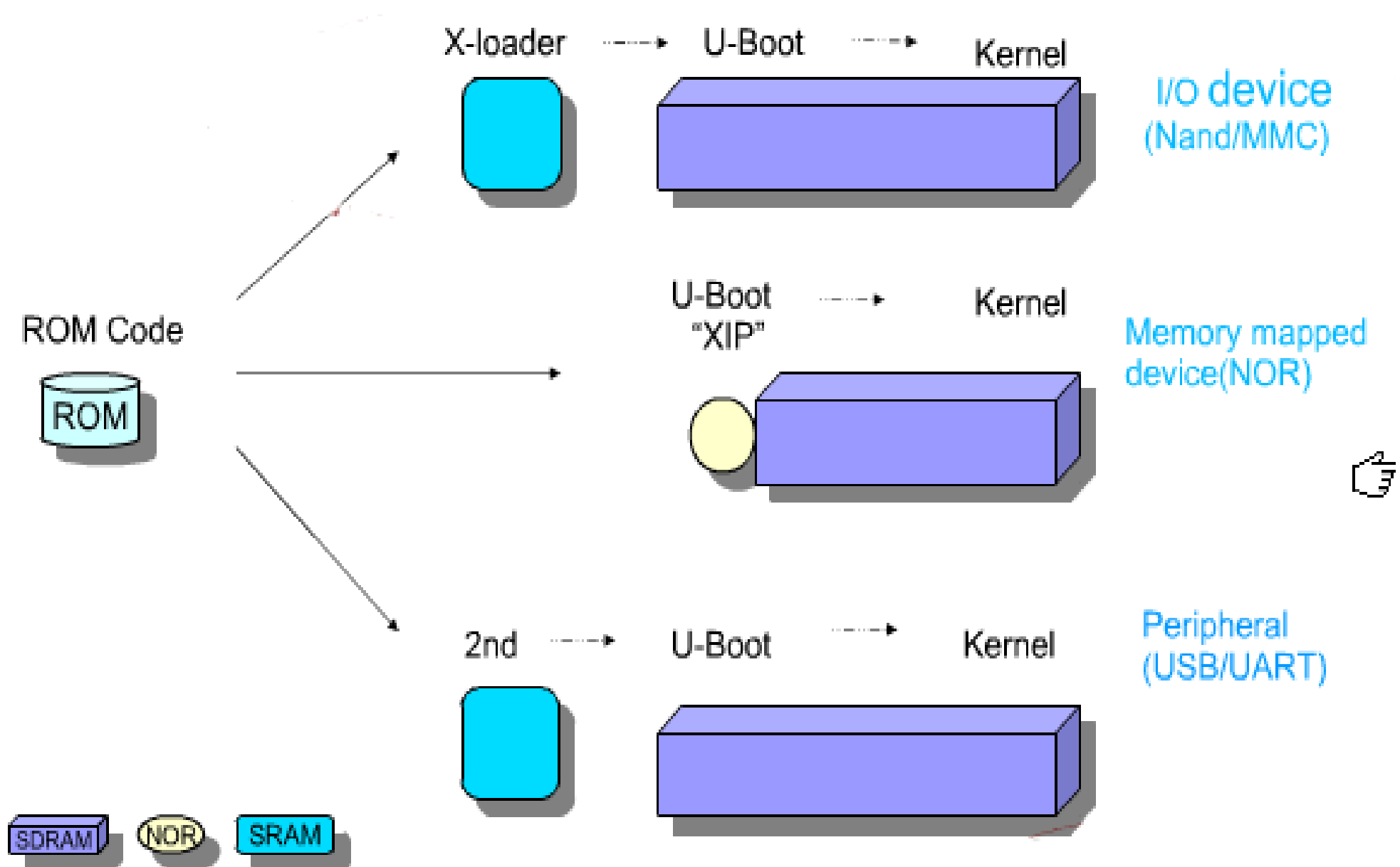


SAMSUNG S3C2440  
ARM9 Core @502 MHz  
LCD/2D

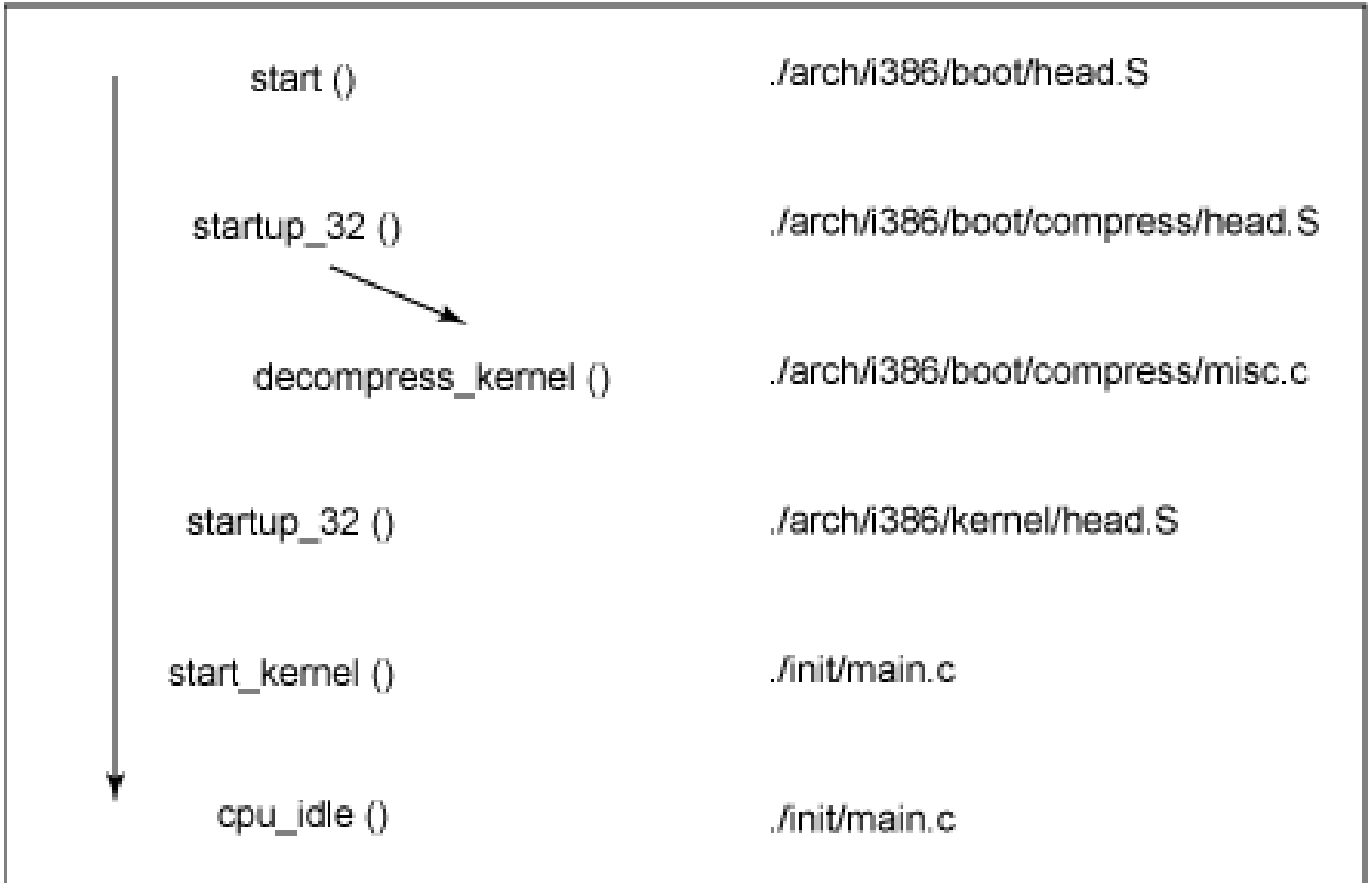
## BOOT Loader – ÖNYÜKLEYİCİ

- ✓ Kesmelerin durdurulması
- ✓ Sistem saatlerinin (CLOCK) düzenlenmesi.
- ✓ Sistem belleğinin düzenlenmesi.
- ✓ Çevre birimlerinin ilklendirilmesi.
- ✓ Asıl uygulamanın (OS) yüklenip işletilmesi.

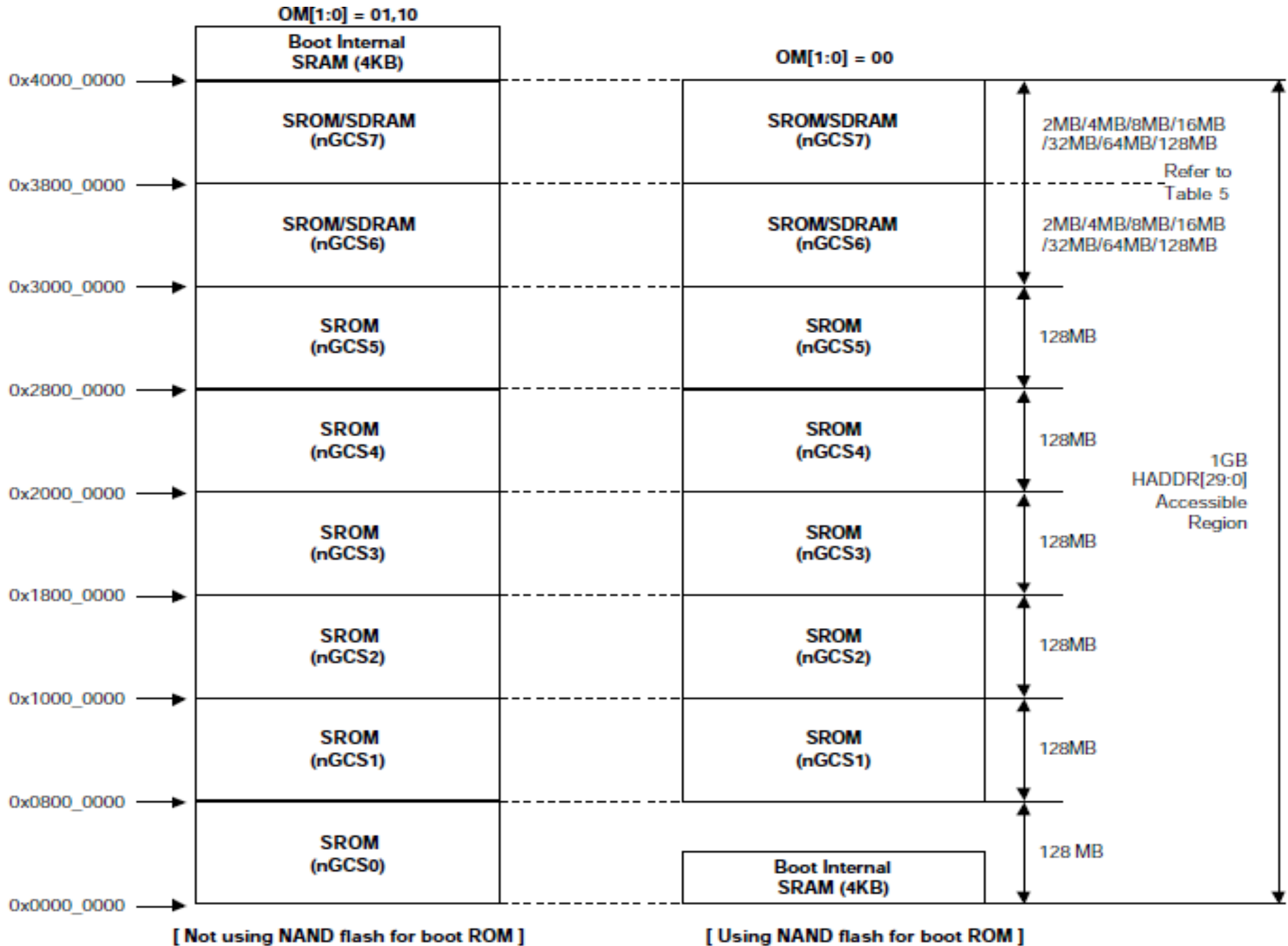
# BOOT Loader – ÖNYÜKLEYİCİ



## Linux Startup Code



# Yazılım tarafı – RESET ardından bellek haritası



## U-BOOT, VIVI ....

- x Genel amaçlı
- x Fazla büyük
- x Gereğinden fazla karmaşık
- x Bağımlılık sorunları

***Kendi bootloaderinizi türetin veya yazın.***

***Bootloader yerine doğrudan kerneli yükleyin***

## LCD için Framebuffer sürücüsü

- ✓ LCD seçerken RGB veya LVDS tercihi  
*Framebuffer işini CPU'ya yüklemek*
- ✓ MCU seçerken Linux desteğini gözetmek  
*Hazır FB sürücülerini kullanabilme imkanı*
- ✓ LCD + MCU için tam Linux desteği
- ✓ Kontrolcüsü dahili LCD'lerden kaçının.

## ✓ linux/include/linux/fb.h

fb\_var\_screeninfo

fb\_fix\_screeninfo

fb\_cmap

fb\_info

fb\_ops

## ✓ fbmem.c

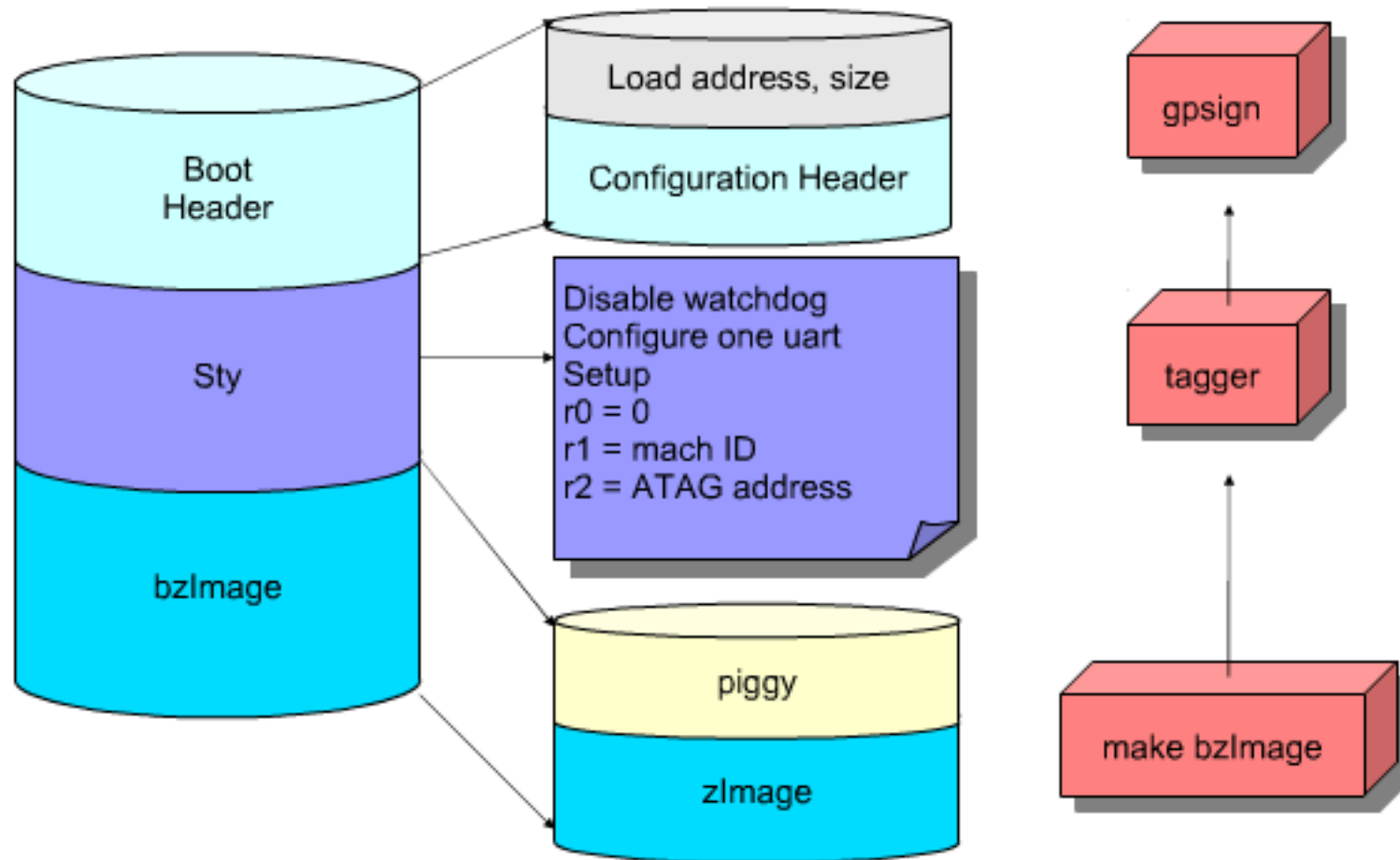
register\_framebuffer(struct fb\_info \*fb\_info)

unregister\_framebuffer(struct fb\_info \*fb\_info)

## ✓ linux/drivers/fb/skeleton.c

- ✓ Frame buffer'in fiziksel adresini belirleyin  
`fb_get_fix()`
- ✓ `ioctl()` için gereken operasyonları gerçekleştirin.
  - `FBIOGET_VSCREENINFO` `fb_get_var`
  - `FBIOPUT_VSCREENINFO` `fb_set_var`
  - `FBIOGET_FSCREENINFO` `fb_get_fix`
  - `FBIOPUTCMAP` `fb_set_cmap`
  - `FBIOGETCMAP` `fb_get_cmap`
  - `FBIOPAN_DISPLAY` `fb_pan_display`
- ✓ `fb_set_var()`
  - Mod (320x240x8 gibi) değişimi gerekiyor mu?
  - Yeni modu hazırlayın, FB bit paternlerini belirleyin.
  - Gerekliyse “colormap” ları düzenleyin.
  - LCD kontrol registerleri yeni duruma göre düzenleyin.

# KERNEL – BOOTLOADER



✓ Ext2, ext3, FAT32 ...

HDD için optimize.

NAND okuma/yazma limitlerine uygun değil.

FS yükü fazla.

✓ Yaffs, yaffs2, jfs, jffs2 ...

NAND/NOR vs. için optimize.

Embedded sistemler için kullanışlı ekler.

*On-the-fly compression, bad sector move..*

Genellikle daha yavaş

CONFIG\_MTD=y

CONFIG\_MTD\_PARTITIONS=y

CONFIG\_MTD\_CHAR=y

CONFIG\_MTD\_BLOCK=y

CONFIG\_MTD\_NAND=y

CONFIG\_MTD\_NAND\_YOURBOARD=y

CONFIG\_JFFS2\_FS=y

CONFIG\_JFFS2\_FS\_DEBUG=0

CONFIG\_JFFS2\_FS\_NAND=y

- ✓ **Sistem için kökdizini**  
mkdir /nav\_device\_path/rootfs  
cd /nav\_device\_path/rootfs  
for I in sys dev etc bin sbin usr lib; do  
    mkdir \$I  
done
- ✓ **Gerekli /dev girdilerini oluşturma**  
mknod *T MAJOR MINOR* /dev/hede
- ✓ **Init dosyaları**  
vi /etc/inittab  
mkdir /etc/rc.d || vi /etc/rc.d/init-script

## LIBC

- ✓ Statik / Dinamik kullanım
- ✓ Çeşitli varyasyonlar
  - GLIBC*
  - EGLIBC*
  - uClibc*
  - dietlibc*
  - newlib*
  - android libc*

[http://www.etalabs.net/compare\\_libcs.html](http://www.etalabs.net/compare_libcs.html)

Grafik arabirim

DirectFB

GTKFB

X over FB

Micro X

SDL

Qtopia & Qt4/E

## Grafik arabirimler için ipuçları

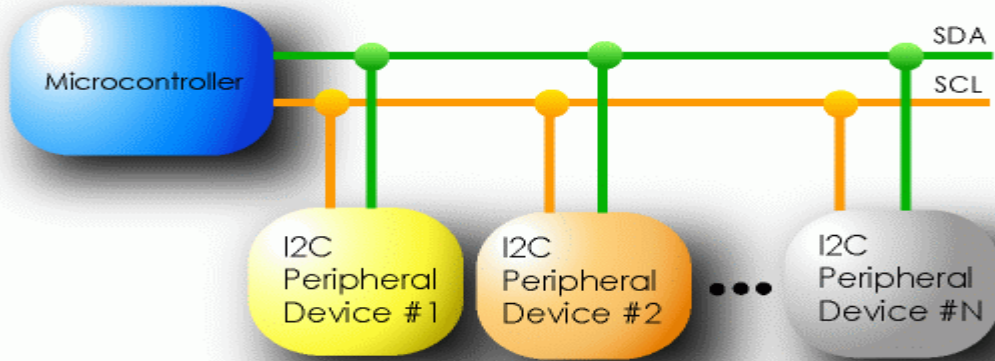
- ✓ RGBA yerine Palet tabanlı GUI
- ✓ Gerekmeyen her şeyi atın.
  - Bitmap'ler, ikonlar.*
  - Kullanılmayan widgetler*
  - Dil/karakter destekleri.*
  - Giriş yöntemleri*
  - Hata ayıklama kodları*

- ✓ C en taşınabilir dildir, onu C++ izler.
- ✓ Uygunsa emülatör kullanımı  
QEMU, P-SIM ...
- ✓ Geliştirme kitleri / Embedded kartlar
  1. Kodlama
  2. Derleme
  3. Emülatörle test / Geliştirme kitinde test
  4. Prototip üzerinde test.
  5. Nihai uygulama imajı

## USB erişimi

- ✓ USB Host, OTG desteği için kernel...
- ✓ USB Device olarak...
  - Doğrudan kernel kodlama
  - GADGETFS (libgadgetfs)
  - USB Storage
  - USB Serial
  - USB Ethernet
  - RNDIS

## I<sup>2</sup>C eriřimi



- ✓ Kernel sürücüsü gerekebilir !
- ✓ Kernel sürücüsü, sadece bus için řarttır.
- ✓ Bazı I2C cihazları kernelden desteklenir.
- ✓ I2C cihazları için, I2C Framework..  
/sys/class/i2c-adapter  
/dev/i2c-\*

## I<sup>2</sup>C erişimi

```
#include <linux/i2c-dev.h> // from i2c-tools
```

```
int file = open("/dev/i2c-0", O_RDWR);
```

```
if (file < 0) ERROR HANDLING;
```

```
int addr = 0x40;
```

```
if (ioctl(file, I2C_SLAVE, addr) < 0)
    ERROR HANDLING; 3)
```

```
buf[0] = DEV_register;
```

```
buf[1] = 0x55;
```

```
buf[2] = 0xAA;
```

```
if (write(file, buf, 3) != 3)
```

```
    ERROR HANDLING;
```

```
    if (read(file, buf, 1) != 1)
```

```
        ERROR HANDLING;
```

```
    /* buf[0] = readed byte */
```

## SPI eriřimi

- ✓ CS – Chip Select problemi !
- ✓ Kernel sűrűcűsű kullanılmalıdır.
- ✓ Bazı MSSP'lar GPIO desteđiyle hazır olabilir.
- ✓ SPI kontrolcűsű yazmak en kolay yoldur.

## SPI erişimi

- ✓ Kernel sürücüsü hazır olunca  
`/dev/spidevB.C`
  - B = Bus (MSSP) numarası.*
  - C = CS numarası*
- ✓ Kullanıcı erişimi IOCTL ile sağlanır.  
Documentation / spi / spidev  
*spidev\_test.c*