

## Gömülü Sistemlerde PS/2 Klavye Kullanımı

### Neden PS/2 Klavye ?



Klavyeler, bilgisayarlara veri girişinde vazgeçemediğimiz ünitelerdir. AT klavyesi, ilk IBM PC AT'nin çıktığı 1984'den yılından itibaren elektronik ve protokol olarak değişmemiştir. 1987'de çıkan IBM PS/2 bilgisayar serisi ile konektör ve tuş sayısı olarak bu günkü şeklini alarak **PS/2 Klavye** olarak anılmaya başlamıştır.

Bu de-facto standardın 20 yıllık bir geçmişi olması iki önemli sonuç yaratmıştır:

- **Düşük Fiyatlar** (3 ABD\$ !),
- Elektronik ve protokol olarak uyumlu **Alternatif Veri Giriş Üniteleri** (Bu tür üniteler arasında lazer / CCD barkod tabancaları ve barkodlu / manyetik kart okuyucular sayılabilir).



Bir PS/2 klavyeye; elektronik açıdan bakıldığında 8 x 16 boyutlarına varan bir matris tuş takımı ve bu tuş takımında basılan tuş bilgisi kodlayan bir denetleyiciden (Intel 8048 veya benzeri) oluşur. PC'mizde ise bütün işi bu denetleyiciyle iletişimi sürdürmek olan bir klavye denetleyicisi (Intel 8042 veya benzeri) yer alır.

Bu iletişimin protokolüyle ilgili detaylı bilgiyi Adam Chapweske'nin yazılarında ( (1) ve (2) ) bulabilirsiniz.

Gömülü sistemler açısından protokolün iki ana özelliğine dikkat edilmelidir:

1. İletişim senkron ve seri şekilde **sadece iki kanal** üzerinden (**CLOCK** ve **DATA**) yapılmaktadır:

Otomasyon çözümlerinde kullanılan gömülü sistemlerde matris tuş takımlarının direkt olarak denetleyicinin giriş/çıkış kanallarına bağlanması sık kullanılan bir çözümdür. Örneğin bu çalışmada kullanılan **Flashlite186** için 4x4 (yani 16 tuşlu) bir tuş takımı ve yazılım kütüphanesi üretici **JKmicro** tarafından verilmektedir.

Ancak matris tuş takımları satır ve sütun sayılarının toplamları kadar giriş/çıkış kanalı gerektirir. Flash186'daki hazır 4x4 tuş takımı uygulaması 8 kanal kullanmaktadır. Bu hesap yöntemiyle 101-102 tuşun (standart PS/2 klavyesinin) matris olarak kullanımı en az 21 kanal gerektirecektir. Flash186'da toplam 44 tane olan giriş/çıkış kanallarının önemli bir kısmının sadece klavye için kullanılması tam anlamıyla kaynak israfı olacaktır.

Özetle, PS/2'nin sadece iki kanal kullanan iletişim protokolü önemli bir avantaj sağlamaktadır.

Matris tuş takımları için kodlayıcı özel devre geliştirmek kolay ve yaygın bir uygulama olmakla birlikte maliyetlerini 3 ABD\$'ın altına çekmek zordur.

2. İletişim **çift** yönlüdür: Klavyeden PC'ye bilgi akışı olduğu gibi, PC'den klavyeyi ayarlamaya yönelik komutlar gitmektedir:

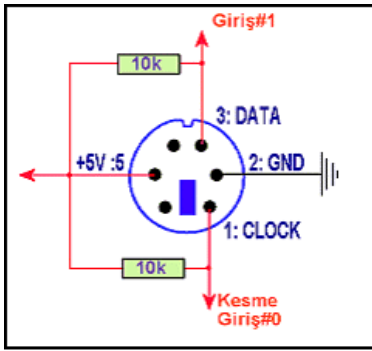
PC'den gönderilen komutlara baktığımızda; bunların temelde klavye LED'lerinin ve tuş tekrarlama özelliğinin ayarında kullanıldığını görürüz. Otomasyon çözümlerde kullandığımız gömülü sistemlerin çoğunda bu ayarlara gerek duyulmayacağı göz önüne alınarak tek yönlü bir protokol kullanılabilir. Sadece klavyeden veri almak üzere tasarlanan protokoller daha az programlama yükü ve kaynak gerektirecektir.

Bu nedenle çalışmamızda tek yönlü iletişim baz alınmıştır.

## Bağlantılar ve Protokol

PS/2 klavyelerin sağladığı avantajlar nedeniyle, İnternet'te çeşitli platformlar için geliştirilmiş bir çok uygulama yer almaktadır. Örnek olarak, 68HC705 için Craig Peacock'un geliştirdiği kodun yer aldığı sayfada (3) PS/2 klavyelerle ilgili detaylı bilgiler verilmektedir. Atmel'in AVR işlemcisi için C'de yapılmış bir örnek çözüm ise (4) teki belgede verilmektedir.

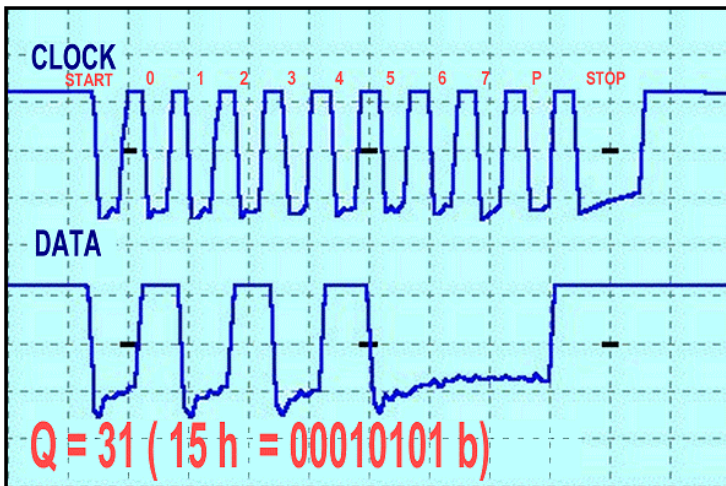
Klavyeyi herhangi bir denetleyiciye bağlamadan önce yapılması gereken iki önemli hazırlık vardır:



- 1) Klavyenin bazen 100mA'e kadar çıkabilen; regüleli, +5VDC besleme gereksiniminin karşılanması. Bu bağlantıda klavyeye bağlanan besleme toprağının denetleyici besleme toprağıyla kısa devre olmasına dikkat edilmelidir.
- 2) Açık kolektör (open collector) yapıdaki DATA ve CLOCK sinyallerinin 10kOhm'luk "pull-up" dirençleriyle koşullandırılması. Açık kolektör bağlantıda; sinyal yoksa yada mantıksal "0" seviyesinde ise çıkışta 5V, sinyal "1" seviyesinde ise 0V gözlemlenir.

Klavyede bir tuşa basılması bir grup veri paketi akışına neden olur. Her veri paketi 11 adet CLOCK çevrimi süresinde tamamlanır: Her biri **11 bit** olarak algılanabilecek bu paketler; 1 adet başlangıç (= 0), 8 adet veri, 1 adet parite (= 0 veya =1) ve 1 adet stop (=1) bitinden oluşur.

Adam Chapweske'nin yazısında (1); PS/2 klavyede 'Q' tuşuna "basıldığında" ASCII değeri olarak 31 (16 tabanında "15") şeklinde algılanması gereken değer için CLOCK ve DATA' da gözlemlenen dalgalar aşağıdaki şekilde görüntülenmiştir:



Dalga görüntüleri yardımıyla iletişim protokolü için vurgulanması gereken sonuçlar aşağıdadır:

**CLOCK sadece veri akışı süresince sinyal üretmektedir:**

Yani bu sinyal bir kesme (interrupt) uygulaması için uygundur.

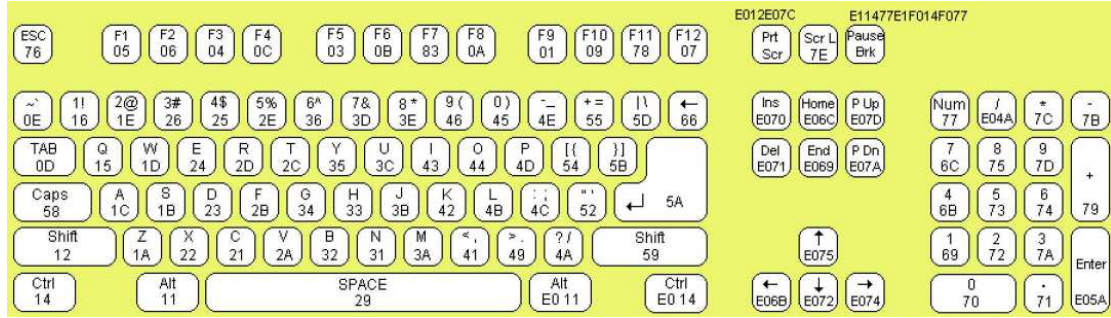
**DATA sinyalinin geçerli verileri 5V'tan 0'a inerken oluşmaktadır:**

Kesme yordamında veri okuma bu aşamada gerçekleştirilmelidir.

## Tuş Kodları

Klavye kullanımımıza dikkat edersek aslında birkaç aşamalı bir işlem yürüttüğümüz fark ederiz. Tek tuşa kısa bir süre bastığımızda bile; tuşa basılması, ardından tuştan parmağın çekilmesi göz önüne alınmalıdır. Tuşa uzun süre basılması sonrasında tekrarlama, [SHIFT], [ALT], [ALTGR], [CTRL] tuşlarıyla birlikte basmamız ile çıkan farklı sonuçlar ise kodlayıcının becerdiği diğer işlerdir.

Craig Peacock'ın yazısında (3), bir tuşa basıldığı zaman ürettiği kodları aşağıdaki çizimle verilmektedir:



Bu çözüm yardımıyla kod çözümü yaparken aşağıdaki noktalara dikkat edilmelidir:

- Bazı tuşlar iki, [PRT SCR] ve [PAUSE] tuşları ise daha çok karakter üretir.
- Tuşlar, parmak çekildiği anda (F0)(XX) şeklinde tekrar kod gönderir.
- Çizim ABD'de kullanılan İngilizce klavye sıralamasını vermektedir. Farklı klavyelerin ürettiği tuş kodları değişmeyeceği için yorumlayan modülünün bu durumu göz önüne alması gerekecektir. Örneğin Türkçe F klavye kullanımı halinde 'Q' harfinin yerine sadece etiket olarak 'F' gelmekte ancak kodlayıcı yine (15h) kodunu göndermektedir.

## FlashLite186 için Örnek Uygulama

Bu çalışmada kullanılan FlashLite186; 33Mhz 80186 uyumlu RDC8822 işlemcili, üzerinde DOS yüklü olarak gelen, kompakt, endüstriyel ( çalışma sıcaklık aralığı -20 ila +85 °C olan) bir ana karttır. İstenirse 7-34 VDC, regülesiz bir adaptörle beslenebilmektedir. Detaylı teknik özelliklerine <http://www.lima.com.tr/JK/FlashLite186.asp> bağlantısından erişilebilir.

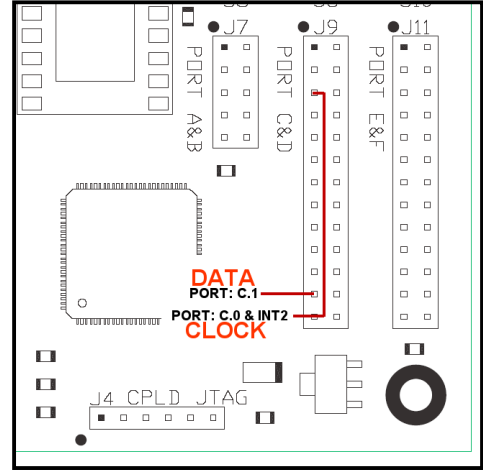


İşletim sistemi DOS olunca bir çok alışlagelmiş geliştirme platformundan yararlanılabilmektedir. Bu çalışmada Borland C/C++ V.4.52 kullanılmıştır. FlashLite186'ya uygulama yükleme işlemi; kart üzerinde "konsol" portu olarak adlandırılan seri port yardımıyla, PC üzerinde Hyperterminal benzeri bir programda XModem protokolü kullanılarak yapılmaktadır. Seri portun konsol olarak adlandırılması bilinçli bir seçimdir: bir PC'de klavyeden beklediğimiz girişler ve ekrana metin yazma şeklindeki çıkışlar bu porta yönlendirilmektedir.

FlashLite186 üzerinde 44 adet dijital giriş/çıkış kanalı yer almaktadır.

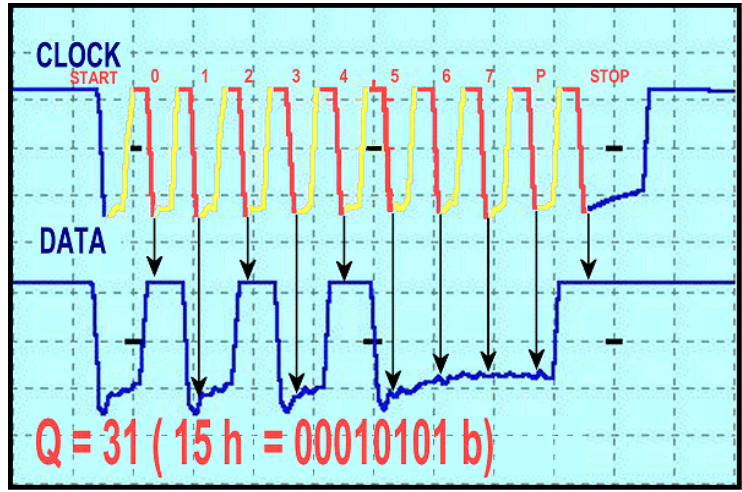
Uygulamada J9 konektörü üzerinde yer alan ve PORTC diye anılan kanallardan ikisi kullanılmıştır. Bu konektör üzerinde ayrıca INT2 kesmesi için de bir pin yer almaktadır.

CLOCK sinyali, PortC'nin 0. pinine ek olarak bu kesme pinine bağlanarak programlama kolaylığı sağlanmıştır.



Uygulamada kesme yordamı, CLOCK sinyali 0V'tan 5V'a geçerken tetiklenmektedir. Bu aşama, çizimde gösterilen dalgaların sarı renkli aşamalarında gerçekleşir. Programda, kırmızı aşamalarda CLOCK tekrar 0V'a düşene kadar beklenmekte ve DATA ancak bu durumda okunmaktadır.

Kesmeleri yandaki dalgalara göre toplayan ardından karakter ve 'shift' kodlarını çözümleyen C fonksiyonları aşağıda verilmektedir:



```
/* **** */
/* ps2_int: PS/2 klavye kodlayıcısının gönderdiği 10 bitli */
/* toplayan kesme hizmet yordamı */
/* Parite ve Stop bitlerinin doğruluğunu kontrol etmez. */
/* **** */
void interrupt ps2_int(void)
{
    unsigned char retry = 0, binp;
    unsigned static char bkey = 0, bcount = 0;

    if (bcount < 10)
    {
        // CLOCK sinyali aşağı (0V) inene kadar bekle
        while ( ((binp = inportb(0x602)) & 1) && (retry++ < PS2_WAIT) );

        if ( retry < PS2_WAIT )
        {
            if ( (bcount < 8) && (binp & 2) ) bkey = bkey + bits[bcount];
            bcount++;
        } else {
            // CLOCK sinyali aşağı inmedi - HATA
            bcount = 0;
            bkey = 0;
        }
    } else {
        // 10. bit geldi - Tuşu çözümler
        ps2_decode(bkey);
        bcount = 0;
        bkey = 0;
    }
    // INT2 için kesme sonu
    outport(EOI, 0x000E );
}

```

```

/*****
/* ps2_decode: Kesme yordamının topladığı değerleri */
/* 'key' (klavye) ve 'shift' kodlarına dönüştürür. Son */
/* iki dönüşüm arasında geçen zaman farkına bağlı olarak */
/* karakter sayısını (charcount) artırır */
/* shiftcode bitlerinin kullanım şekli aşağıdadır: */
/*      0 : Sol / Sağ Shift */
/*      1 : Sol / Sağ Control */
/*      2 : Sol / Sağ Alt */
/*      3 : 'Extended' Tuş (kodu 'E0' ile başlıyor) */
*****/
void ps2_decode(unsigned char keyin)
{
#define PS2_RATE 3
    //Decode key or shift code
    switch (keyin)
    {
        case 0x0://NULL
            break;

        case 0xF0://Key UP identifier
            break;
        case 0xE0://Extended key
            shiftcode[charcount] = shiftcode[charcount] | 8;
            break;
        case 0x11://Left or Right Alt
            shiftcode[charcount] = shiftcode[charcount] | 4;
            break;
        case 0x12://Left SHIFT
            shiftcode[charcount] = shiftcode[charcount] | 1;
            break;
        case 0x14://Left or Right CONTROL
            shiftcode[charcount] = shiftcode[charcount] | 2;
            break;
        case 0x59://Right SHIFT
            shiftcode[charcount] = shiftcode[charcount] | 1;
            break;
        default :
            keycode[charcount] = keyin;
            break;
    }

    if ( (keycode[charcount] != 0) && \
        (labs(biostime(0, 0L) - ps2_clock ) >= PS2_RATE ))
    {
        if (charcount < PS2_MAX)
        {
            charcount++;
            shiftcode[charcount] = 0;
            keycode[charcount] = 0;
        }
    }
    // Son çözümleme zamanını güncelle
    ps2_clock = biostime(0, 0L);
}

```

Uygulamanın tamamını aşağıdaki web sayfasından indirebilirsiniz.  
<http://www.lima.com.tr/Support/JKSupport.asp>

```
PS2TEST
PS/2 Keyboard Test Application for FL186 Ver.1.01
by Lima Endüstriyel Bilgisayar - www.lima.com.tr

30 bits received.
00 : 1 : 0 : 1 :
01 : 0 : 0 : 0 :
02 : 1 : 0 : 1 :
03 : 0 : 0 : 0 :
04 : 1 : 1 : 1 :
05 : 0 : 1 : 0 :
06 : 0 : 1 : 0 :
07 : 0 : 1 : 0 :
PAR: 0 : 1 : 0 :
STP: 1 : 1 : 1 :
KEY: 15: SHIFT : 00
```

**PS2Test** programı araştırma amaçlı '#define DEBUG' tanımı açıkken derlenip çalıştırdıktan sonra 'Q' tuşuna basılırsa yandaki görüntüyü verecektir.

İlk bayt (15h) 'Q' tuşuna basıldığını, F0h tuşun bırakıldığını, sonuncu 15h ise bırakılan tuşun 'Q' olduğunu göstermektedir.

Uygulama, lazer tabanca ile yapılan barkod okutmalarını da başarılı olarak çözümlenmektedir.

## Sonuç

PS/2 klavyenin elektronik protokolü, gömülü sistemlerde kolaylıkla gerçekleştirilebilmektedir. Bu çözüm; giriş/çıkış kanallarından tasarruf veya alternatif veri giriş ünitelerinden yararlanılmasını gerektiren ortamlarda unutulmamalıdır.

## Kaynaklar

- 1) Adam Chapweske - PS/2 Mouse/Keyboard Protocol, 1999
- 2) Adam Chapweske - The AT-PS/2 Keyboard Interface, 2001
- 3) Craig Peacock - Interfacing the AT keyboard, 19 Agu. 2001 ([www.beyondlogic.org/keyboard/keybrd.htm](http://www.beyondlogic.org/keyboard/keybrd.htm))
- 4) Atmel - AVR313: Interfacing the PC AT Keyboard , <http://www.atmel.com/atmel/acrobat/doc1235.pdf>
- 5) JKmicro – FlashLite186 User's Manual [http://www.jkmicro.com/documentation/pdf/FL186\\_man14.pdf](http://www.jkmicro.com/documentation/pdf/FL186_man14.pdf)

Cem Çelik  
Lima Endüstriyel Bilgisayar  
[www.lima.com.tr](http://www.lima.com.tr)

İstanbul - Haziran 2004