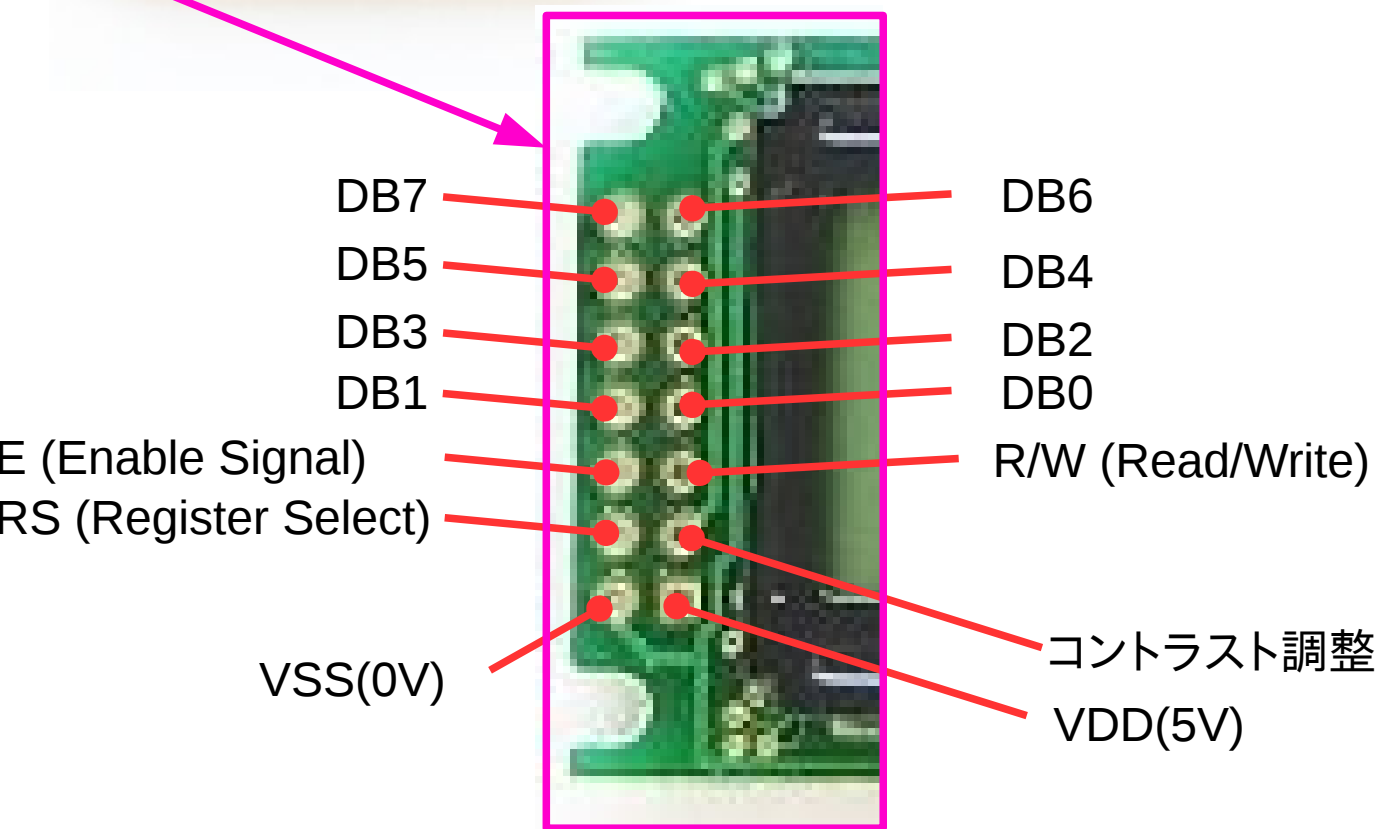


# キャラクタ型液晶の使い方

ノートPCなどの液晶と違って、コマンドを送ると文字を書いてくれる液晶  
フォントなどをマイコンに設定する必要がないので、簡単に文字を表示できる



例) SC1602BS-B(XA)

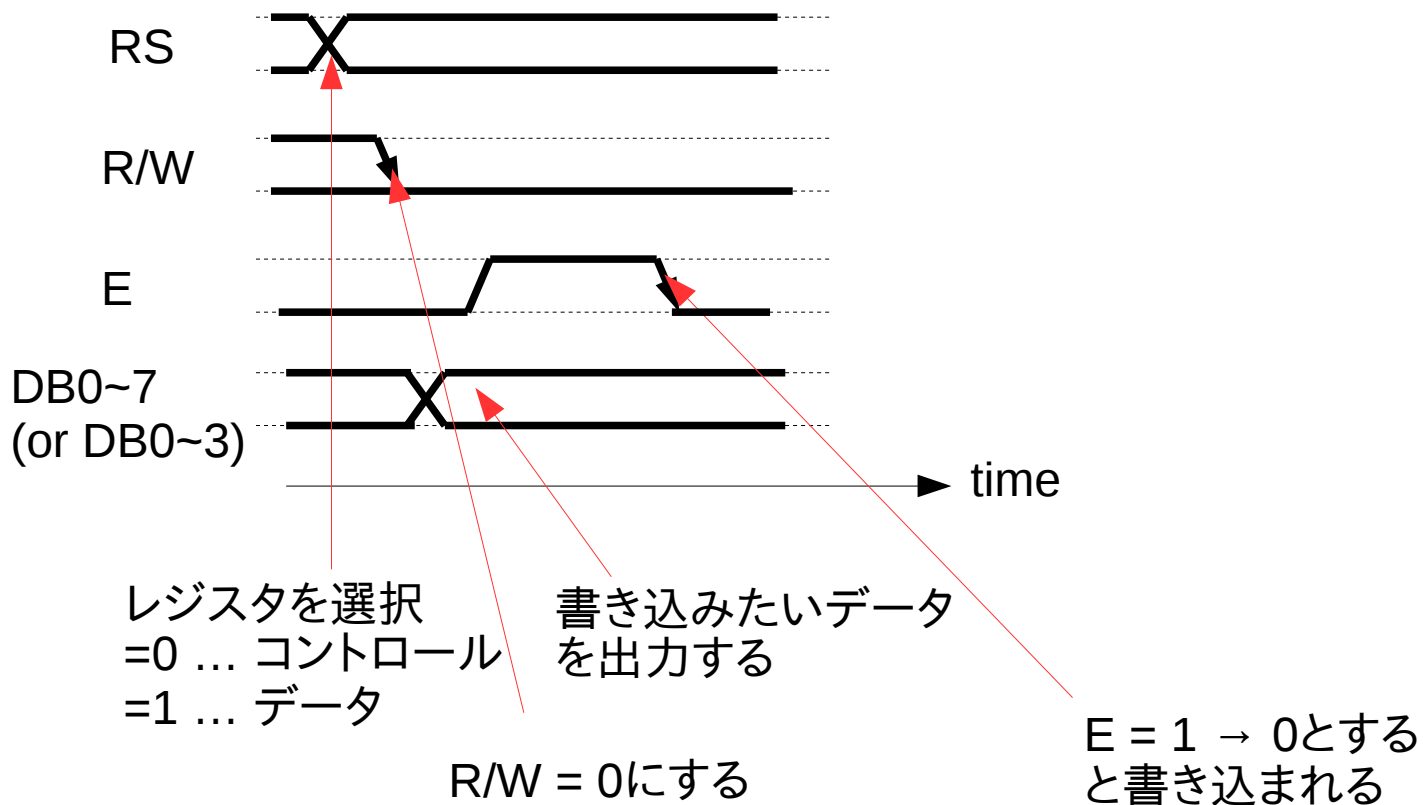


データ転送は8bitモードと  
4bitモードがある。

4bitモードの時は、DB0~3は  
使わない

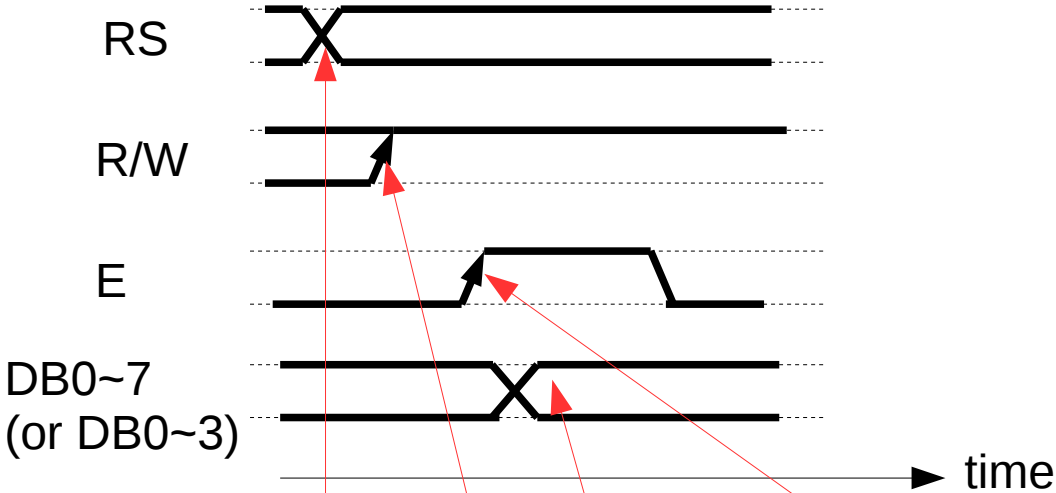
# キャラクタ型液晶の使い方

## データの書き込み



# キャラクタ型液晶の使い方

## データの読み込み



レジスタを選択  
=0 ... コントロール  
=1 ... データ

R/W = 1にする

読み出す

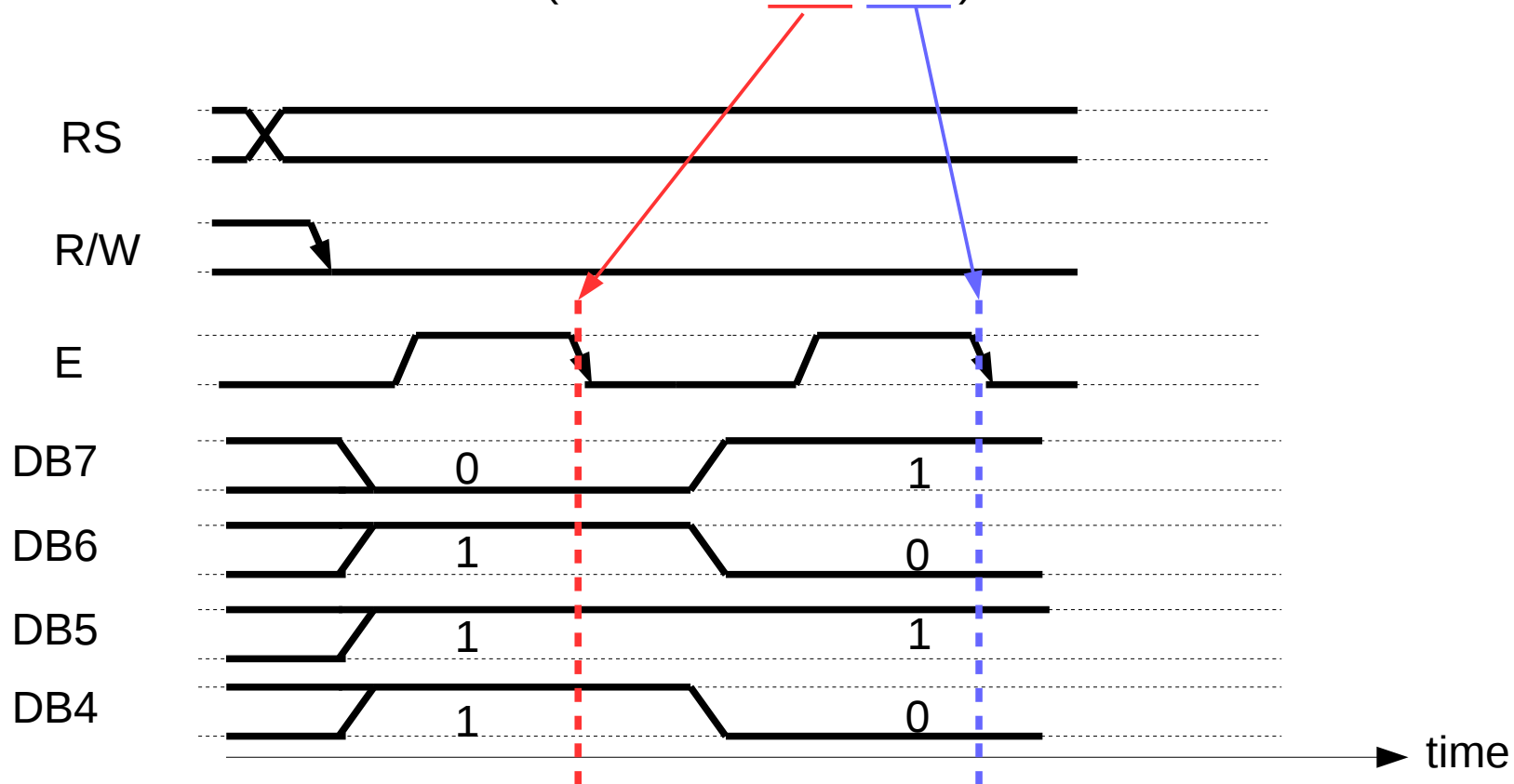
E = 0 → 1とする  
とデータが出力さ  
れる

# キャラクタ型液晶の使い方

## 4ビットモードでの使用法

1バイト(8bit)のデータを、2回に分けて出力する。  
最初に上位4ビット、次に下位4ビットを転送する。(使用するデータ線はDB4~7)

例) 0x7Aを転送したいとき(2進数では01111010)



# キャラクタ型液晶の使い方

## 初期化

1. 電源投入後15ms以上待つ
2. RS=0, RW=0として、DB[7:4]に0011を出力する
3. 4.1ms以上待つ
4. RS=0, RW=0として、DB[7:4]に0011を出力する
5. 100us以上待つ
6. RS=0, RW=0として、DB[7:4]に0011を出力する
7. 1ms待つ
8. RS=0, RW=0として、DB[7:4]に0011を出力する(ここから先は4ビットモードになる)
9. データシートのレジスタ表をみて、欲しい機能のセッティングを行う。  
(これ以降は、BF(ビジーフラグ)が使えるようになるので、BFを見て、書き込みできるかどうか判断できる。BFを使わないなら、ひとつの書き込みが終わった後、固定時間待つようにする。  
動作がおかしい場合は、この、待ち時間が短いことが多いので、適宜調整してみる。)

# キャラクタ型液晶の使い方

データの書き込み

```
void LCD_putc(char dat){  
    int i;
```

```
LCD_WAIT();
```

```
LCD_D = (LCD_D & ~LCD_D_MASK) | (LCD_D_MASK & (dat>>4)); //Upper 4 bits
```

```
LCD_RS = 1;
```

```
LCD_RW = 0;
```

```
LCD_E = 1;
```

```
asm("nop");
```

```
LCD_E = 0;
```

```
LCD_D = (LCD_D & ~LCD_D_MASK) | (LCD_D_MASK & dat); //Lower 4 bits
```

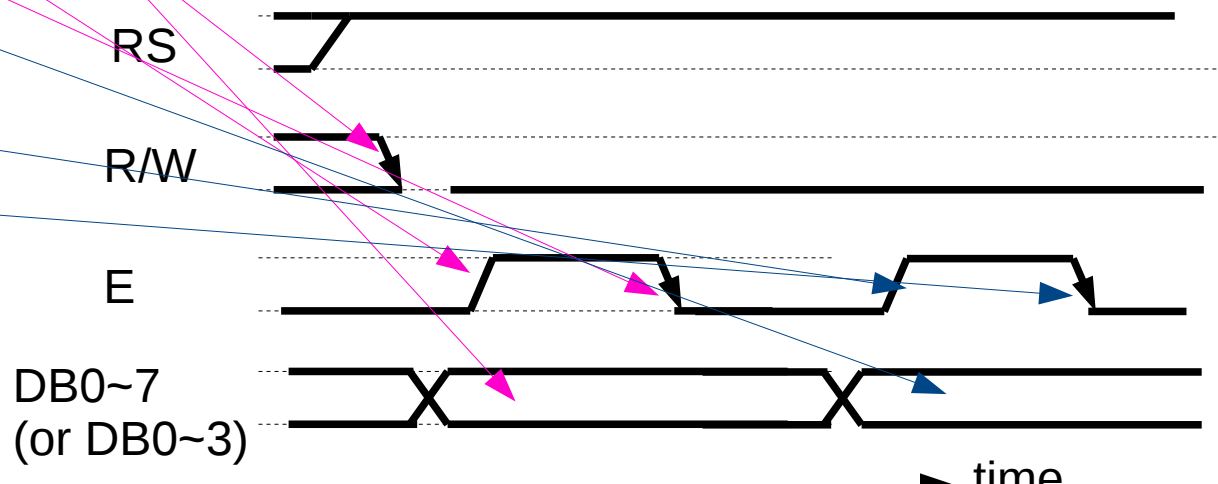
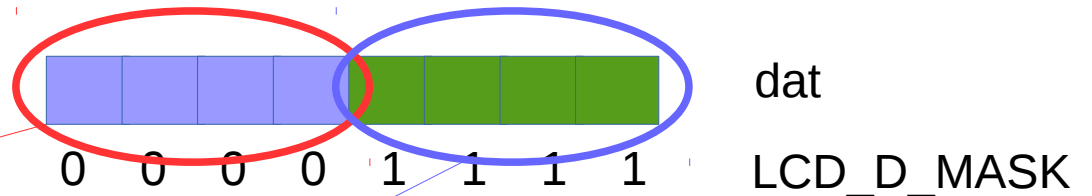
```
asm("nop");
```

```
LCD_E = 1;
```

```
asm("nop");
```

```
LCD_E = 0;
```

```
}
```



# キャラクタ型液晶の使い方

文字列を出力したい場合は、↓のような関数を使うとべんり

```
int LCD_puts(char* str, int num){  
    while(*str){  
        LCD_putc(*str);   ここで, 1文字出力  
        str++;  
        if( ! (num--)) break;  
    }  
    return 0;  
}
```