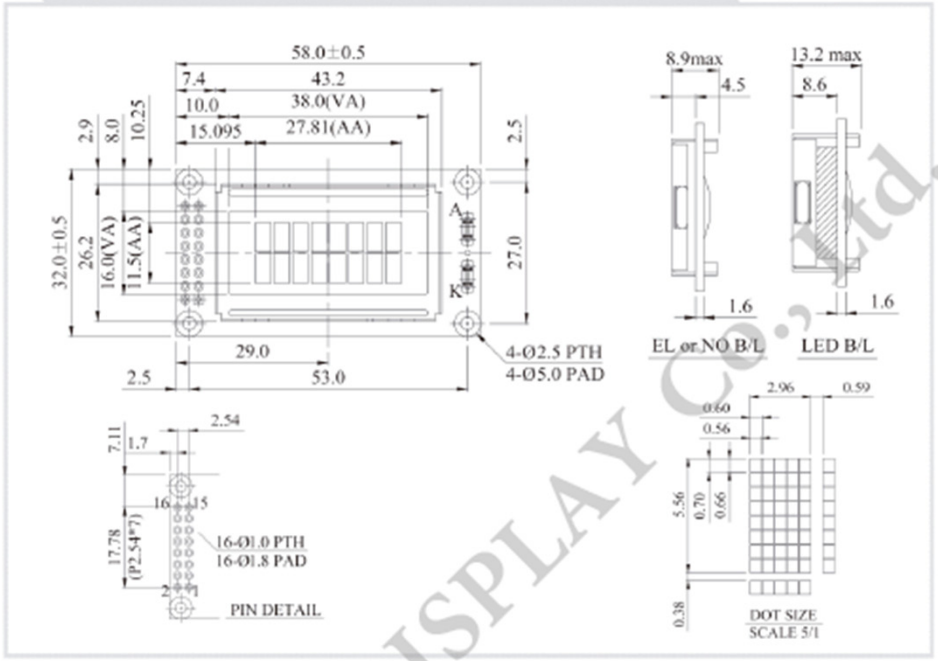
Display LCD
(caracteres alfanúmericos)

Display LCD

WH0802A1 Character 8x2



Feature

1.5x8 dots includes cursor
2.Bulit-in controller (ST7066 or Equivalent)
3.5V power supply (Also available for 3V)
4.N.V. optional for 3V power supply
5.1/16 duty cycle
6.LED can be driven by PIN1, PIN2, PIN15, PIN16 or A and K
7.Interface : 6800, option SPI/I2C (RW1063 IC)

Pin No.	Symbol	Description
1	V_{SS}	Ground
2	V_{DD}	Power supply for logic
3	V_O	Contrast Adjustment
4	RS	Data/ Instruction select signal
5	R/W	Read/Write select signal
6	E	Enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	A	Power supply for B/L +
16	K	Power supply for B/L -

Mechanical Data

Item	Standard Value	Unit
Module Dimension	58.0 x 32.0	mm
Viewing Area	38.0 x 16.0	mm
Mounting Hole	53.0 x 27.0	mm
Character Size	2.96 x 5.56	mm

Electrical Characteristics

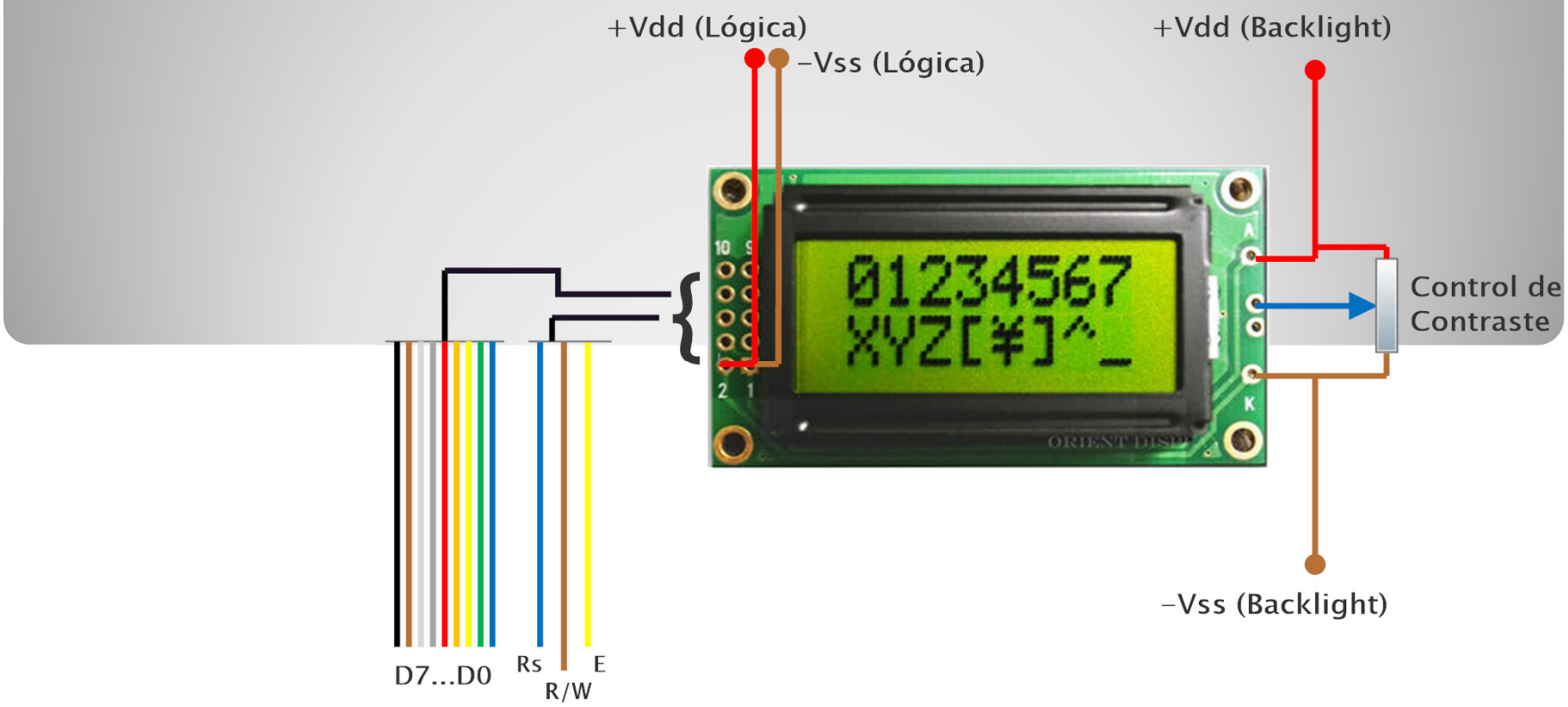
Item	Symbol	Standard Value typ.	Unit
Input Voltage	VDD	3/5	V
Recommended LCD Driving Voltage for Normal Temp. Version module @25°C	VDD-VO	4.35	V

Display Character Address Code

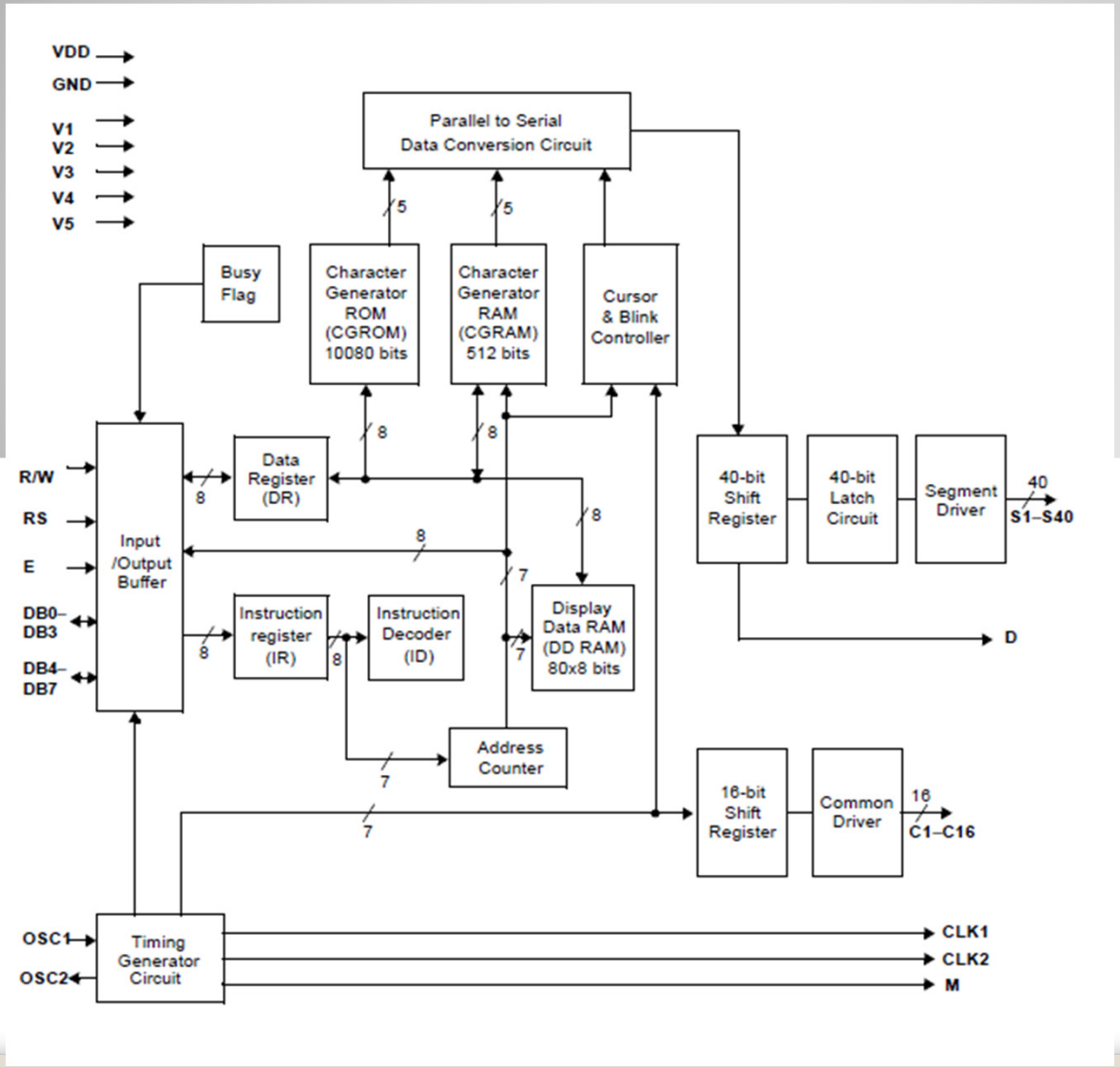
Display position

	1	2	3	4	5	6	7	8
DD RAM Address	00	01						07
DD RAM Address	40	41						47

Display LCD


+Vdd (Lógica)
–Vss (Lógica)
+Vdd (Backlight)
01234567
XYZ[¥]^_
Control de
Contraste
–Vss (Backlight)
D7...D0
Rs
R/W
E

Display LCD diagrama de bloques

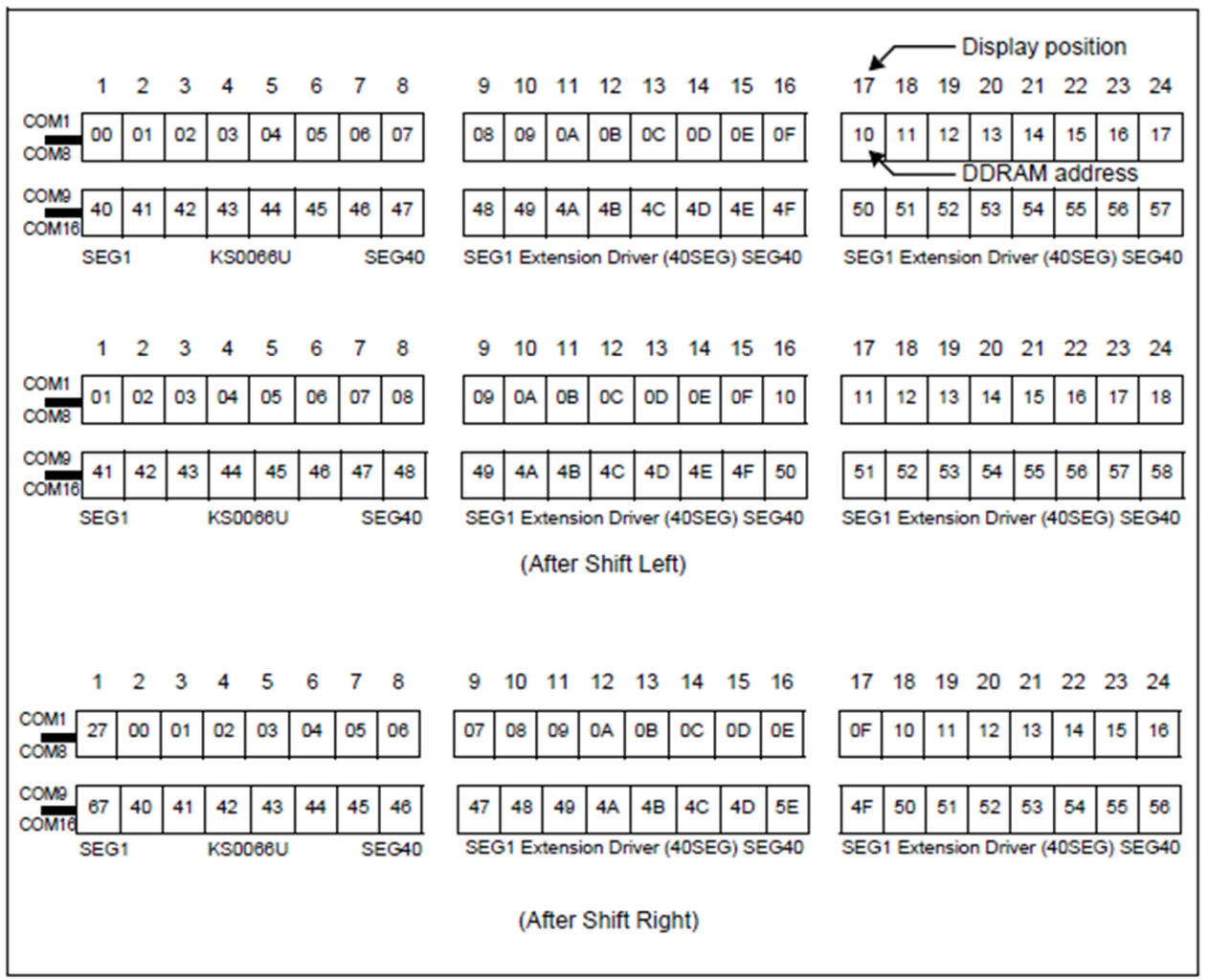


Display LCD: instrucciones

Instruction	Instruction Code										Description	Execution time (fosc= 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM and set DDRAM address to "00H" from AC	1.53 ms
Return Home	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39 µs
Display ON/ OFF Control	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39 µs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	39 µs
Function Set	0	0	0	0	1	DL	N	F	-	-	Set interface data length (DL: 8-bit/4-bit), numbers of display line (N: 2-line/1-line) and, display font type (F:5×11dots/5×8 dots)	39 µs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 µs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 µs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 µs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 µs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 µs

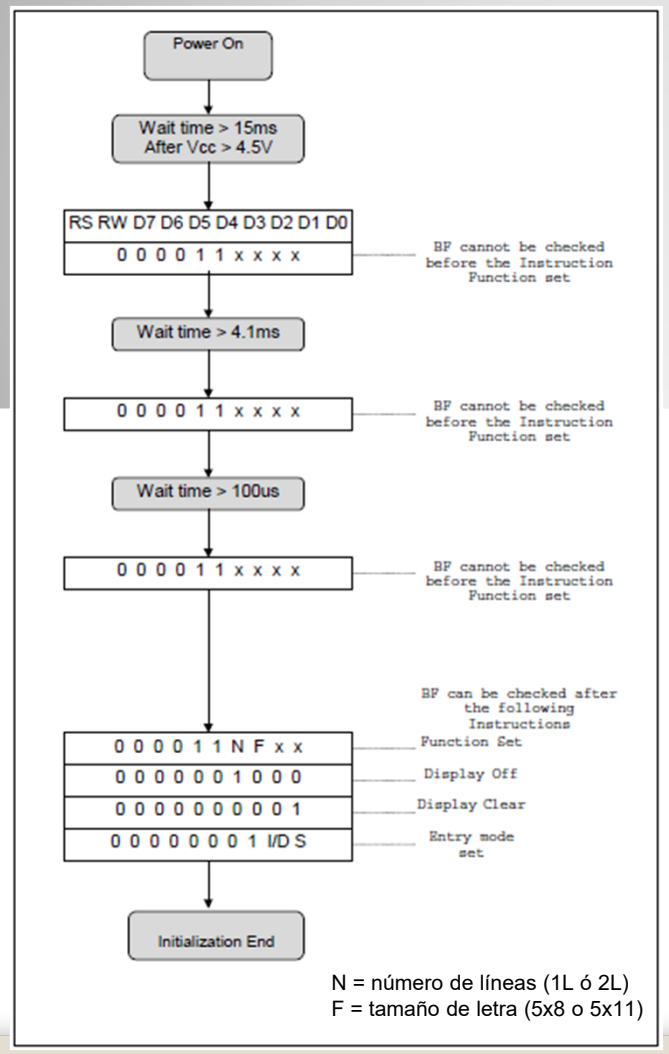
* "-" dont care

Display LCD: memoria

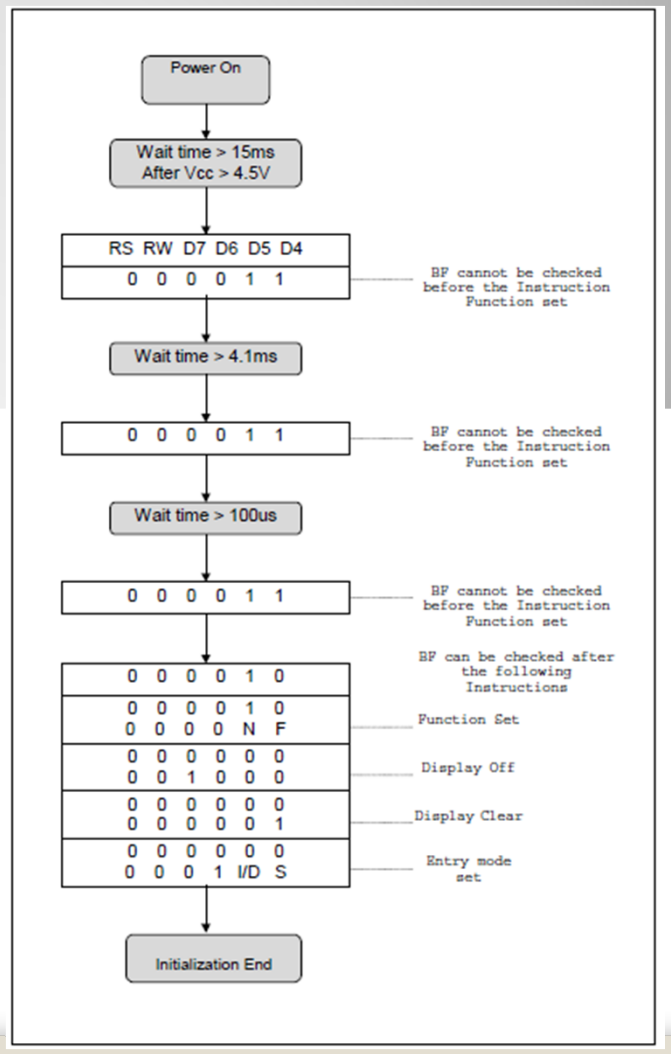


Display LCD: inicialización

Modo 8-bits

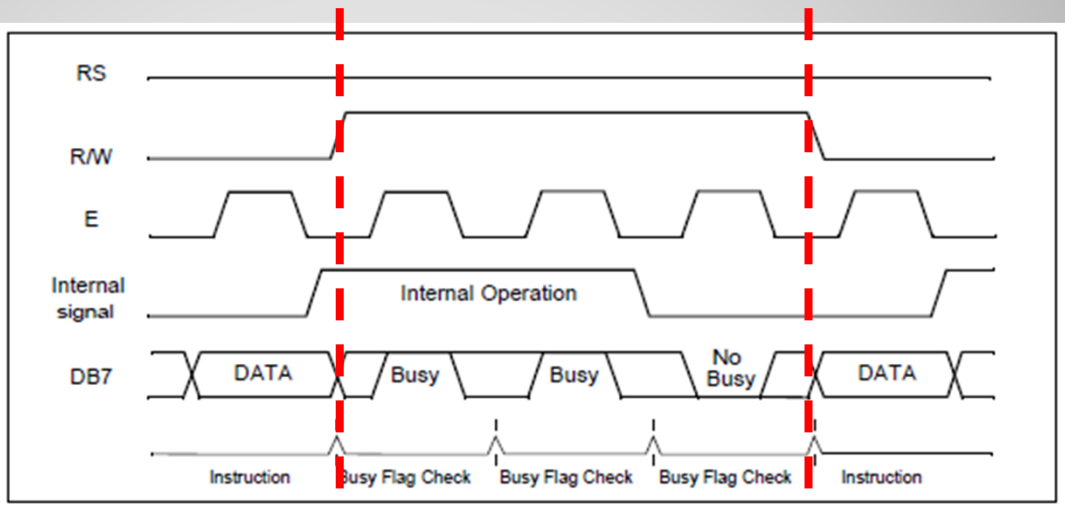


Modo 4-bits



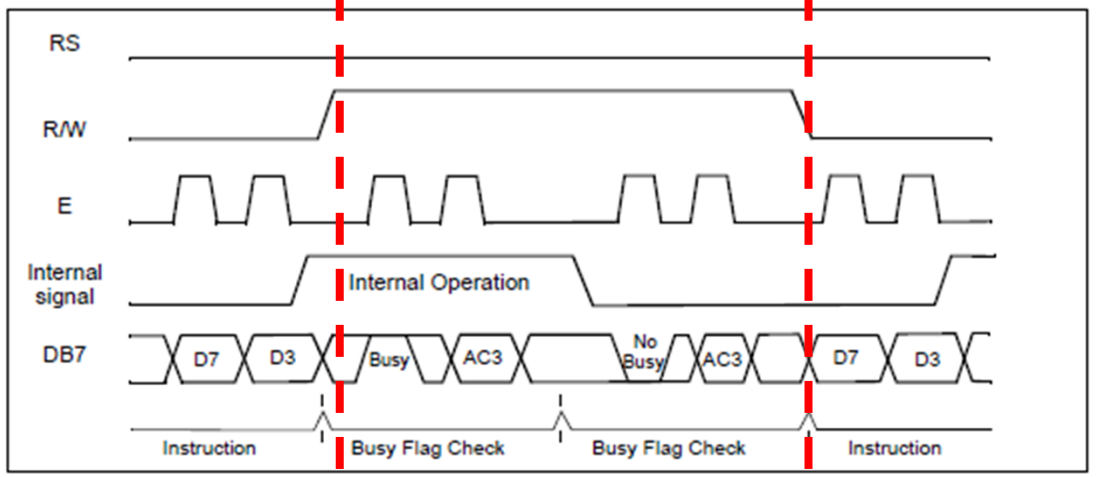
Display LCD temporización

Modo 8-bits

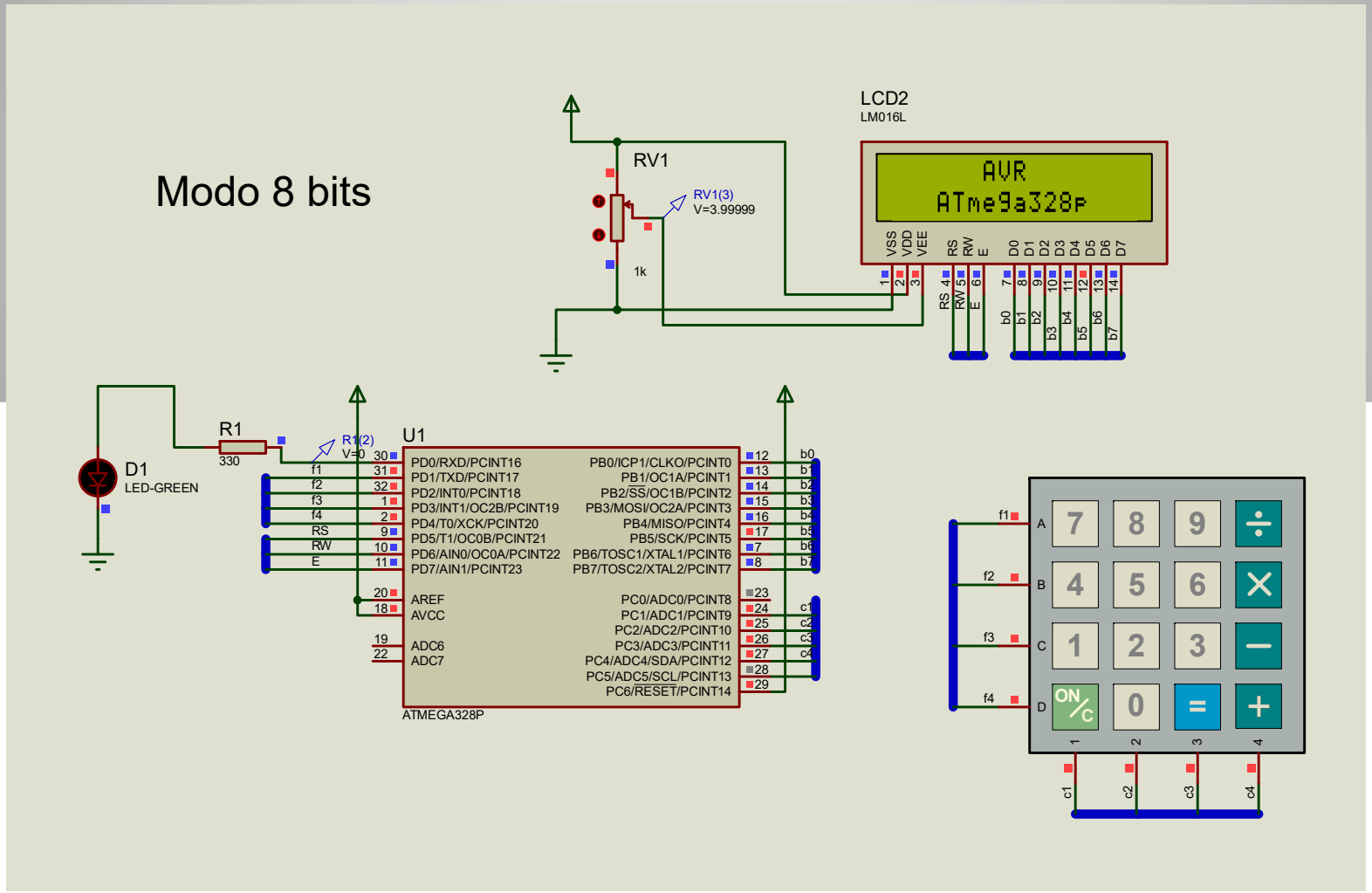


Modo 4-bits

delay

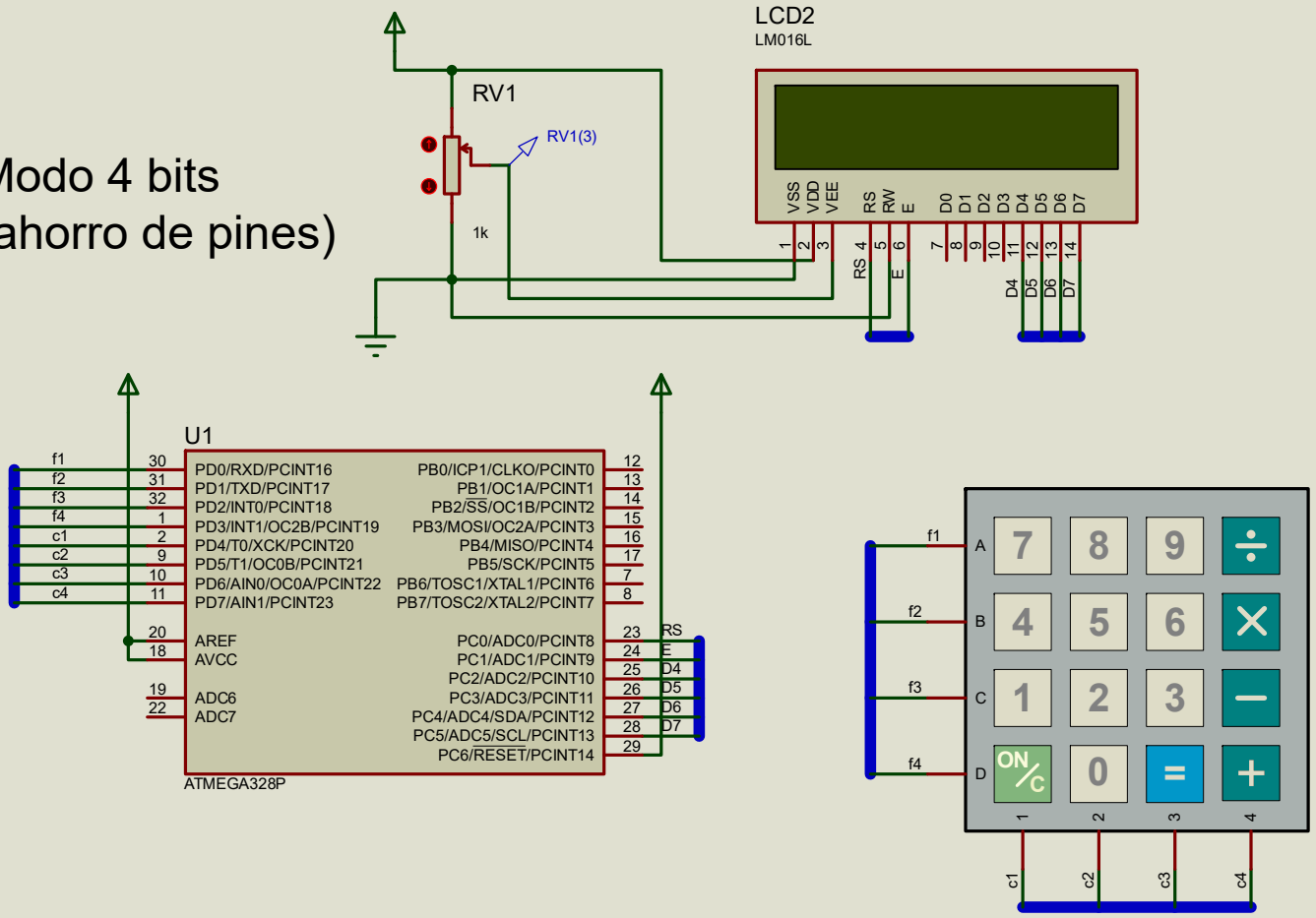


Conexión


Modo 8 bits
LCD2
LM016L
AVR
ATme9a328p
RV1
RV1(3)
V=3.99999
1k
R1
330
R1(2)
V=0
D1
LED-GREEN
U1
ATMEGA328P
PD0/RXD/PCINT16
PD1/TXD/PCINT17
PD2/INT0/PCINT18
PD3/INT1/OC2B/PCINT19
PD4/T0/XCK/PCINT20
PD5/T1/OC0B/PCINT21
PD6/AIN0/OC0A/PCINT22
PD7/AIN1/PCINT23
AREF
AVCC
ADC6
ADC7
PB0/ICP1/CLKO/PCINT0
PB1/OC1A/PCINT1
PB2/SS/OC1B/PCINT2
PB3/MOSI/OC2A/PCINT3
PB4/MISO/PCINT4
PB5/SCK/PCINT5
PB6/TOSC1/XTAL1/PCINT6
PB7/TOSC2/XTAL2/PCINT7
PC0/ADC0/PCINT8
PC1/ADC1/PCINT9
PC2/ADC2/PCINT10
PC3/ADC3/PCINT11
PC4/ADC4/SDA/PCINT12
PC5/ADC5/SCL/PCINT13
PC6/RESET/PCINT14
ON/C

Conexión

Modo 4 bits
(ahorro de pines)



Bibliotecas

LCD_AVR.h
8 bits

Definiciones de Comandos

```
#ifndef F_CPU
#define F_CPU 1000000UL // clock speed is 1MHz
#endif

#include<avr/io.h>        // AVR header
#include<util/delay.h>    // delay header

//=========================================================
 // libería LCD inicio Hardware
 //=========================================================
```

```
#define LCD_DATA PORTC   // puerto C es puerto de datos LCD

#define ctrl PORTC              //  puerto C es puerto de comandos LCD

#define en 1                    // enable en pin 1 puerto C

//#define rw PIN_RW             // read/write en pin

#define rs 0                    // RS en pin 0 puerto C

#define SHIFT_Datos 2           // desplazamiento desde el MBS dato al MSB pin
```

Conexiones

```
//=========================================================
// LCD constantes de configuración
//=========================================================
#define CURSOR_ON 2
#define CURSOR_OFF 0
#define CURSOR_BLINK 1
#define CURSOR_NOBLINK 0
#define DISPLAY_ON 4
#define DISPLAY_OFF 0
#define DISPLAY_8X5 0
#define DISPLAY_10X5 4

#define _2_LINES 8
#define _1_LINE 0

#define BITS8 0x30
#define BITS4 0x20

#define MEM_ini_Lin1 0x80
#define MEM_ini_Lin2 0xC0
```

Instruction	Instruction Code										Description	Execution time (fosc= 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM and set DDRAM address to "00H" from AC	1.53 ms
Return Home	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39 µs
Display ON/ OFF Control	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39 µs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	39 µs
Function Set	0	0	0	0	1	DL	N	F	-	-	Set interface data length (DL: 8-bit/4-bit), numbers of display line (N: 2-line/1-line) and, display font type (F:5×11dots/5×8 dots)	39 µs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 µs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 µs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 µs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 µs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 µs

* "-": dont care

```
//=========================================
// Prototipos
//=========================================

void LCD_cmd(unsigned char cmd);

void LCD_init(unsigned char mode1, unsigned char mode2);

void LCD_write(unsigned char data);

void LCD_gotoxy (unsigned char x, unsigned char y);

void LCD_write_string(unsigned char *str);
```

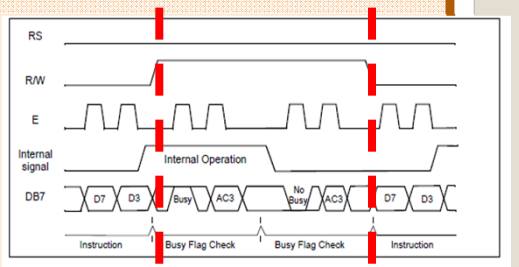
Funciones

Bibliotecas

```
#include <inttypes.h>
#include <avr/io.h>
#include"LCD_4b.h"

//=====================================
// Funciones
//=====================================
void init_LCD(void)
{
  DDRB=0xF0;            // datos LCD como salida
  DDRD=0xE0;            //  RS, RW, E como sallidas

 LCD_cmd(0x30);         //    inicialización    //
 _delay_ms(1);
 LCD_cmd(0x30);         //                      //
 _delay_ms(10);
 LCD_cmd(0x32);         //    inicialización    //
 _delay_ms(1);

 LCD_cmd(0x28);         //   Modo 4 bits      //
 _delay_ms(1);

 LCD_cmd(0x0C);         //  2 lineas      //
 _delay_ms(1);
 LCD_cmd(0x01);         //  clear LCD
 _delay_ms(1);
 LCD_cmd(0x02);         // return home
 _delay_ms(1);
 LCD_cmd(0x06);         //  incrementa  cursor
 _delay_ms(1);
 LCD_cmd(0x80);         //  1 y 0 al primer caracter
 _delay_ms(1);
 return;
}
```

```
//=========================================
//Ubicacion de cursor
//=========================================

void LCD_gotoxy (unsigned char x,unsigned char y)          // y= 1 o 2, x de 1  a 16
{
   unsigned char inicio[]={0x80 | 0x00, 0x80 | 0xC0};    // Inicio de: L1=0x00,  L2 = 0x40
   LCD_cmd(inicio[y-1]+x-1);
   _delay_ms(100);
}
```

```
//=======================================
//  Envio de Comando
//=======================================

void LCD_cmd(unsigned char cmd)

{

LCD_DATA = cmd;                // comando en linea , parte alta

PORTD  &= ~(1<<rs);            // RS sets 0

PORTD  &= ~(1<<rw);            // RW sets 0

PORTD  |= (1<<en);             // habilita enable

_delay_ms(20);

PORTD  &= ~(1<<en);            // deshabilita enable

 LCD_DATA = (cmd<<4);          // parte baja

 PORTD  &= ~(1<<rs);           // RS sets 0

PORTD  &= ~(1<<rw);            // RW sets 0

PORTD  |= (1<<en);   // habilita enable

_delay_ms(20);

PORTD  &= ~(1<<en);  // deshabilita enable

return;
}
```



Bibliotecas

```
//================================
// Escribir Dato
//================================

void LCD_write(unsigned char data)
{
LCD_DATA= (data);        // dato en linea, parte alta

PORTD  |= (1<<rs);       // RS sets 1

PORTD  &= ~(1<<rw);   // RW sets 0

PORTD  |= (1<<en);       // habilita enable

_delay_ms(20);

PORTD &= ~(1<<en);      // deshabilita enable

LCD_DATA = (data<<4);  // parte baja

PORTD  |= (1<<rs);        // RS sets 1

PORTD  &= ~(1<<rw);     // RW sets 0

PORTD  |= (1<<en);       // habilita enable

_delay_ms(20);

PORTD &= ~(1<<en);     // deshabilita enable

return ;

}
```

```
//===================================
//  Escribir String
//===================================
void LCD_write_string(unsigned char *str)
{
       unsigned char i=0;

      while (str[i] != 0x00)
       {
                    LCD_write ( str[i++]);
       }
 return;
  }
```