



Initialisation Codes for the 4D Systems **4DLCD-144** Display

**NOTE:** There are 3 versions of the 4DLCD-144, as the Driver IC has been changed over the years and new init codes are required. ST7735R and ILI9163C

**For displays with the ST7735 Driver IC (Sold approximately early 2010) –Initial batch sold only**

**Command Defines:**

```
#define NVCTR1          0xD9    // unique to ST7735
#define PWCTR6         0xFC    // unique to ST7735
#define VCOM4L        0xFF    // unique to ST7735
#define SWRESET       0x01
#define RDDID         0x04
#define SLPIN         0x10
#define SLPOUT        0x11
#define PTLON         0x12
#define DISPOFF       0x28
#define DISPON        0x29
#define CASET         0x2A
#define RASET         0x2B
#define RAMWR         0x2C
#define RAMRD         0x2E
#define PTLAR         0x30
#define MADCTL        0x36
#define COLMOD        0x3a
#define FRMCTR1       0xB1
#define FRMCTR2       0xB2
#define FRMCTR3       0xB3
#define INVCTR        0xB4
#define DISSET5       0xB6
#define PWCTR1        0xC0
#define PWCTR2        0xC1
#define PWCTR3        0xC2
#define PWCTR4        0xC3
#define PWCTR5        0xC4
#define VMCTR1        0xC5
#define GAMCTRP1     0xE0
#define GAMCTRNP1    0xE1
#define F0new         0xF0
#define F6new         0xF6
```

**Init Code (Command, Data1, ... DataN)**

```
SWRESET,
Delay 10ms,
SLPOUT,
Delay 120ms,
VCOM4L, 0x40, 0x03, 0x1a,
NVCTR1, 0x60, 0xc7, 0x90,
Delay 200ms,
FRMCTR1, 0x04, 0x25, 0x18,
FRMCTR2, 0x04, 0x25, 0x18,
FRMCTR3, 0x04, 0x25, 0x18, 0x04, 0x25, 0x18,
INVCTR, 0x03,
DISSET5, 0x15, 0x02,
PWCTR1, 0x02, 0x70,
PWCTR2, 0x07,
PWCTR3, 0x01, 0x01,
PWCTR4, 0x02, 0x07,
```

Porting of this Initialisation code to your chosen hosts language will be required. Please refer to the datasheet for the Driver IC used on the display, for more information.



```
PWCTR5, 0x02, 0x04,  
PWCTR6, 0x11, 0x17,  
VMCTR1, 0x3c, 0x4f,  
MADCTL, 0xc8,  
COLMOD, 0x05,  
GAMCTRPl, 0x08, 0x19, 0x16, 0x36, 0x38, 0x2d, 0x25, 0x2a, 0x28, 0x26, 0x33, 0x3d, 0x04, 0x06,  
0x03, 0x0e,  
GAMCTRn1, 0x09, 0x1f, 0x17, 0x36, 0x37, 0x33, 0x2c, 0x32, 0x2f, 0x2c, 0x33, 0x3c, 0x06, 0x06,  
0x03, 0x0f,  
DISPON,  
Delay 10ms,  
RAMWR
```

### **For displays with the ST7735R Driver IC (Sold approximately 2011 to 2013)**

#### **Command Defines:**

```
#define SWRESET          0x01  
#define RDDID           0x04  
#define SLPIN           0x10  
#define SLPOUT          0x11  
#define PTLON           0x12  
#define DISPOFF         0x28  
#define DISPON          0x29  
#define CASET           0x2A  
#define RASET           0x2B  
#define RAMWR           0x2C  
#define RAMRD           0x2E  
#define PTLAR           0x30  
#define MADCTL          0x36  
#define COLMOD          0x3a  
#define FRMCTR1         0xB1  
#define FRMCTR2         0xB2  
#define FRMCTR3         0xB3  
#define INVCTR          0xB4  
#define DISSET5         0xB6  
#define PWCTR1          0xC0  
#define PWCTR2          0xC1  
#define PWCTR3          0xC2  
#define PWCTR4          0xC3  
#define PWCTR5          0xC4  
#define VMCTR1          0xC5  
#define GAMCTRPl       0xE0  
#define GAMCTRn1       0xE1  
#define F0new           0xF0  
#define F6new           0xF6
```

#### **Init Code (Command, Data1, ... DataN)**

```
SWRESET,  
delay 10ms,  
SLPOUT,                               //Sleep Out  
delay 120ms,  
FRMCTR1, 0x02, 0x35, 0x36,           //Setup Frame Rate Control (In Normal Mode/ Full Colours)  
FRMCTR2, 0x02, 0x35, 0x36,           //Setup Frame Rate Control (In Idle Mode/ 8-Colors)  
FRMCTR3, 0x02, 0x35, 0x36, 0x02, 0x35, 0x36, //Frame Rate Control (Partial Mode/ Full Colours)  
INVCTR, 0x07,  
DISSET5, 0xB4, 0xF0,  
PWCTR1, 0xA2, 0x02, 0x84,  
PWCTR2, 0xC5,  
PWCTR3, 0x0A, 0x00,
```

Porting of this Initialisation code to your chosen hosts language will be required. Please refer to the datasheet for the Driver IC used on the display, for more information.



```
PWCTR4, 0x8A, 0x2A,  
PWCTR5, 0x8A, 0xEE,  
VMCTR1, 0x06,  
MADCTL, 0xC8,  
GAMCTRP1, 0x12, 0x1C, 0x10, 0x18, 0x33, 0x2C, 0x25, 0x28, 0x28, 0x27, 0x2F, 0x3C, 0x00, 0x03,  
0x03, 0x10,  
GAMCTRN1, 0x12, 0x1C, 0x10, 0x18, 0x2D, 0x28, 0x23, 0x28, 0x28, 0x26, 0x2F, 0x3B, 0x00, 0x03,  
0x03, 0x10,  
F0new, 0X01,  
F6new, 0X00,  
COLMOD, 0X05,  
CASET, 0x00, 0x02, 0x00, 0x81,  
RASET, 0x00, 0x03, 0x00, 0x82,  
DISPON,  
delay 10ms,  
RAMWR
```

**For displays with the ILI9163C Driver IC (Sold approximately 2013 to present)**

**Command Defines:**

```
#define SWRESET           0x01  
#define SLPOUT           0x11  
#define GAMMASET        0x26  
#define DISPOFF         0x28  
#define DISPON          0x29  
#define CASET           0x2A  
#define PASET           0x2B  
#define RAMWR           0x2C  
#define RAMRD           0x2E  
#define MADCTL          0x36  
#define IPF             0x3A  
#define FRMCTR1          0xB1  
#define SDDC            0xB7  
#define PWCTR1          0xC0  
#define PWCTR2          0xC1  
#define VMCTR1          0xC5  
#define VOC             0xC7  
#define GAMCTRP1        0xE0  
#define GAMCTRN1        0xE1  
#define UNDOC           0xEC  
#define GRSEL           0xF2
```

**Init Code (Command, Data1, ... DataN)**

```
SWRESET,  
delay 10ms,  
DISPON,  
delay 100ms,  
SLPOUT,  
delay 20ms,  
GAMMASET, 0x04,  
FRMCTR1, 0x0B, 0x14,  
PWCTR1, 0x10, 0x00,  
PWCTR2, 0x03,  
VMCTR1, 0x46, 0x40,  
VOC, 0xBD,  
UNDOC, 0x0C,  
IPF, 0x05,  
CASET, 0x00, 0x00, 0x00, 0x7F,
```



**4D SYSTEMS**  
*TURNING TECHNOLOGY INTO ART*

[www.4dsystems.com.au](http://www.4dsystems.com.au)

```
PASET, 0x00, 0x00, 0x00, 0x7F,  
MADCTL, 0xC8,  
SDDC, 0x00,  
GRSEL, 0x01,  
GAMCTRP1, 0x3F, 0x29, 0x27, 0x2C, 0x27, 0x0C, 0x54, 0xC7, 0x40, 0x19, 0x17, 0x1E, 0x02, 0x01,  
0x00,  
GAMCTRN1, 0x00, 0x16, 0x18, 0x13, 0x18, 0x13, 0x2B, 0x38, 0x3F, 0x06, 0x18, 0x21, 0x3D, 0x3E,  
0x3F,  
DISPON
```

Porting of this Initialisation code to your chosen hosts language will be required. Please refer to the datasheet for the Driver IC used on the display, for more information.



**For reading what Driver IC is used on your display**

The following application (written in 4DGL for 4D Systems Processors) can be used to identify which driver IC is used. This program can be ported to another language if not using a 4D Processor. Essentially it reads 3 bytes from the display at a specified address, and those 3 bytes dictate which driver IC is being used.

```
#platform "GOLDELOX"
#inherit "4DGL_16bitColours.fnc"

#constant RDDID          0x04

func main()
  var ID1, ID2, ID3, msg ;

  disp_WriteControl(RDDID);
  disp_ReadByte();           // dummy read
  ID1:= disp_ReadByte();
  ID2:= disp_ReadByte();
  ID3:= disp_ReadByte();

  gfx_MoveTo(0,0);

  // decide which device
  if(ID1 == 0x5C && ID2 == 0x88 && ID3 == 0x35) // "ST7735"
    msg := "ST7735" ;
  else if (ID1 == 0x5C && ID2 == 0x89 && ID3 == 0xF0) // "ST7735R"
    msg := "ST7735R" ;
  else if (ID1 == 0x54 && ID2 == 0x80 && ID3 == 0x66) // "ILI9163C"
    msg := "ILI9163C" ;
  else
    msg := 0 ;
  endif

  if (msg == 0)
    print("Unknown Driver IC,\nID bytes:-\n", [HEX2] ID1, " ", [HEX2] ID2, " ", [HEX2]
ID3) ;
    to(COM0) ;
    print("Unknown Driver IC, ID bytes:- ", [HEX2] ID1, " ", [HEX2] ID2, " ", [HEX2]
ID3, "\n") ;
  else
    print("Driver IC is\n", [STR] msg) ;
    to(COM0) ;
    print("Driver IC is ", [STR] msg, "\n") ;
  endif
  repeat forever
endfunc
```