

**DVK1815Z-A0**

*Product Information*  
**SSD1815Z Development Kit**

DVK1815Z-A0 is a demonstration of SSD1815Z working on a (96 X 64 + 1 icon line) panel. It is intended to help users expedite their design-in of SOLOMON LCD driver.

**PACKAGE CONTENTS**

DVK1815Z-A0 consists of the following:

- 1) LCD Module (96 X 64 + 1 icon line) (DVM1815Z-A0)
- 2) LCD Drawing (optional)
- 3) Programmed 8051 MCU Board (EVM89C52-A0)
- 4) 8051 MCU Board Schematics (EVM89C52-A0)
- 5) Demo Program in C Language (PRG1815Z-A0)

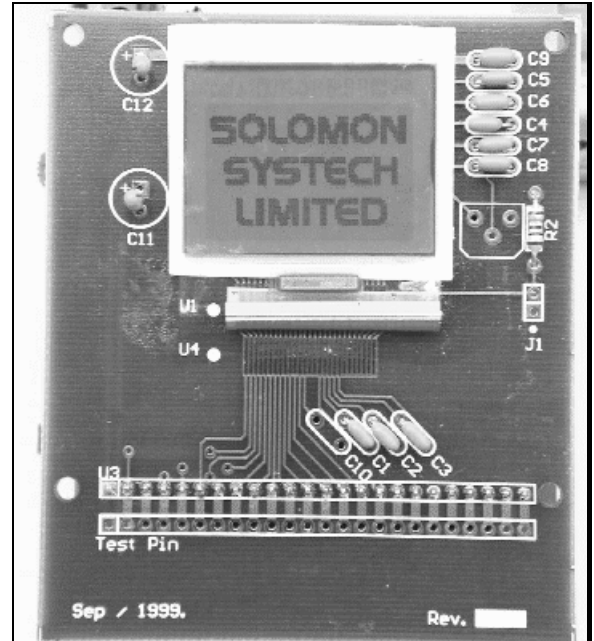
**SYSTEM REQUIREMENT**

DVK1815Z-A0 is good enough to serve as a standalone demo. 2 X AA 1.5V battery is required for normal operation. For engineering evaluation on the LCD driver and/or LCD panel, a 8051 Incircuit emulator (ICE) is required. The one Solomon Systech is using is EMMIT 8051 ICE from Syber Electronics Co Ltd. Usercangetitsinformationfromtheemail:syber@public1.pt

**LCD MODULE**

The LCD module has been configured as follows

- 1) size: 96 X 64 + 1 icon line
- 2) COG Package
- 3) 4X DC-DC converter to generate VEE
- 4) Internal feedback resistor which is software programmable is used to generate VL6.
- 5) 6800 8-bit parallel interface.
- 6) 24 pins single inline (SIL) header is used to connect 8051 MCU Board to LCD Module. Please refer to **Table 1** for pin assignment.



**Figure 1** DVM1815Z-A0 outlook

**ORDERING INFORMATION**

Item	Ordering Part Number
SSD1815Z Development Kit	DVK1815Z-A0
SSD1815Z Development Module	DVM1815Z-A0

Pin Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
LCD Module	VDD	RES	NC	D/C	R/W	CS1	VSS	D0	D1	D2	D3	D4	D5	D6	D7	NC	NC	VSS	NC	NC	NC	NC	NC	NC
8051 MCU Board	VOUT	PA-04	PA-05	PA-01	PA-07	PA-06	GND	PB-00	PB-01	PB-02	PB-03	PB-04	PB-05	PB-06	PB-07	PA-00	VOUT	GND	+9V	P34	P35	P36	P37	NC

**Table 1** DVM1815Z-A0 pin assignment

# 8051 MCU BOARD

The 8051 MCU Board is powered up by 2 X AA 1.5V battery. A 2X DC converter (ICL7660) is used to generate 4.8V to supply 8051 MCU. This voltage is also regulated by LM317 to generate VOUT (adjustable from 1.8V to 3.5V) which in turn powers up solomon LCD driver. All logic output from 8051 are down-converted from 4.8V to VOUT through 74HC4050 non-inverting buffer. User can change VOUT by tuning T1 (1K trimmer). The MCU board is configured to run at a speed of 4MHz. 4 dummy keys are reserved for developing user-interactive application such as tuning contrast.

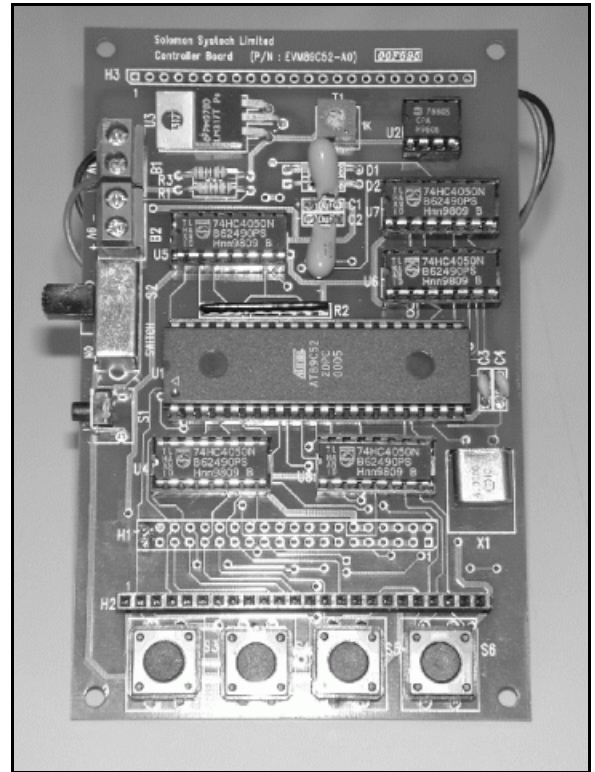


Figure 2 EVM89C52-A0 outlook

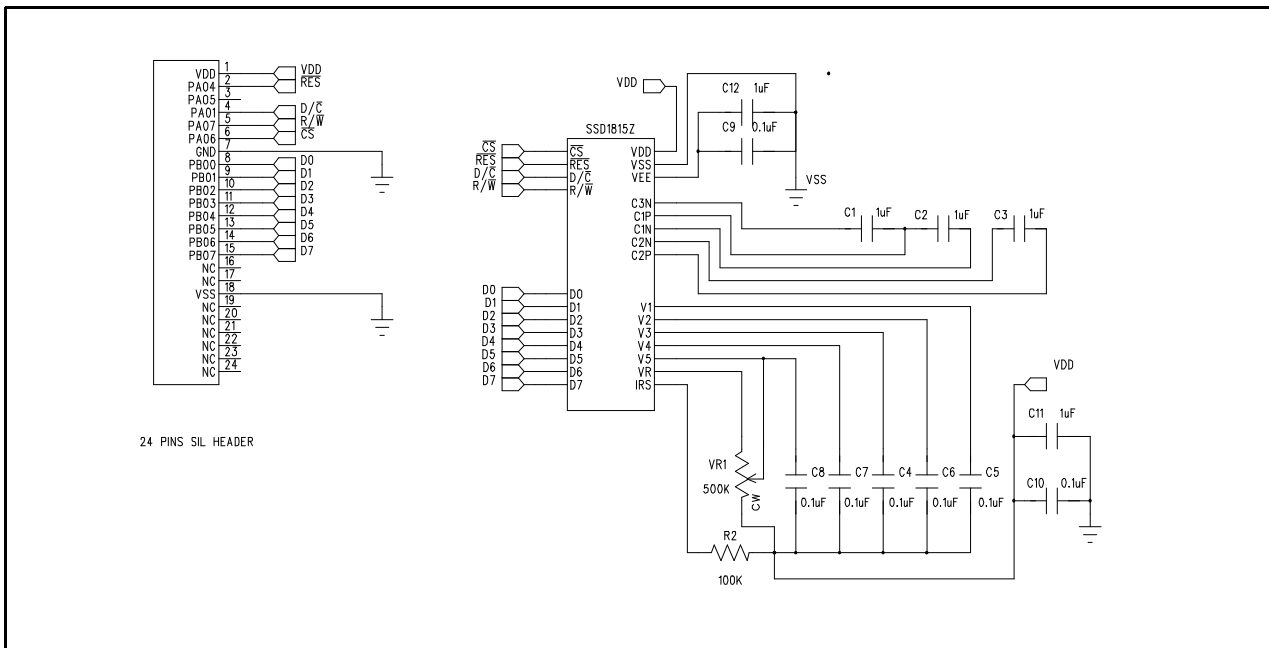


Figure 3 DVM1815Z-A0 schematics

## PROGRAMMING NOTE

```

/*****
*   PRG1815Z-A0
*   1815 Evaluation Program
*****/
#include "reg8751.h"
#define controlport P0
#define dataport    P1

/*****
* A) HardWare Interface
*
*   PORT A  7  6  5  4  3  2  1  0
*           RW CS -- RES -- -- DC --
*   PORT B  7  6  5  4  3  2  1  0
*           D7 D6 D5  D4  D3 D2 D1 D0
*
*****/

/*****
* B) Command Table per device
*****/
#define DisplayOff    0xAE
#define DisplayOn    0xAF
#define DisplayStart  0x40
#define PageAddr     0xB0
#define ColAddrHi    0x10
#define ColAddrLo    0x00
#define SegRemapOff  0xA0
#define SegRemapOn   0xA1
#define NormalDisp   0xA6
#define ReverseDisp  0xA7
#define SWRest       0xE2
#define ComRemapOff  0xC0
#define ComRemapOn   0xC8
#define PwrCtrlReg   0x28
#define OPampBuffer  0x01
#define IntReg       0x02
#define IntVolBstr   0x04
#define IntRegRatio  0x20
#define ContCtrlReg  0x81

#define Device SSD1815Z /* device under demo */
#define ColNo 96        /* number of Column/Seg on LCD glass*/
#define RowNo 64        /* number of Row/Com/Mux */
#define PS 1           /* fixed to Parallel mode */
#define PageNo 8        /* Total no of RAM pages */
#define IconPage 8      /* Icon Page number */
#define SSLNameNo 3

/*****
* C) Global Variable Definition
*****/
```

```

unsigned char WC_CSH;
unsigned char WC_CSL;
unsigned char WD_CSH;
unsigned char WD_CSL;
unsigned char RES_CSH;
unsigned char RES_CSL;

```

```

/*****
* D) Hardcoded Graphics *
*****/

```

```

unsigned char code PAGEQ0[48]={ 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
                                0x80,0x80,0x80,0x80,0x80,0x80,0x80,0xC0,
                                0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
                                0x00,0x00,0x00,0x00,0x80,0x80,0x80,0x80,
                                0x80,0x80,0xC0,0x80,0x00,0x80,0x80,0x80,
                                0x80,0x80,0x80,0xC0,0x80,0x00,0x00,0x00};
unsigned char code PAGEQ1[48]={ 0x00,0x00,0x00,0x00,0x80,0x80,0x80,0x80,
                                0xBF,0x92,0xD2,0x92,0x12,0x92,0x92,0xBF,
                                0x80,0x80,0xC0,0x80,0x00,0x00,0x00,0x00,
                                0x00,0x00,0x00,0x00,0xFF,0x12,0x12,0x12,
                                0x12,0x12,0x3F,0x00,0x00,0x3F,0x12,0x12,
                                0x12,0x12,0x12,0xFF,0x00,0x00,0x00,0x00};
unsigned char code PAGEQ2[48]={ 0x00,0x00,0x00,0x00,0x7F,0x24,0x24,0x24,
                                0x24,0x24,0x7F,0x00,0x00,0x7F,0x24,0x24,
                                0x24,0x24,0x7F,0x00,0x00,0x00,0x00,0x00,
                                0x00,0x00,0x00,0x00,0x7F,0x00,0x00,0x00,
                                0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,
                                0x20,0x60,0x60,0x3F,0x00,0x00,0x00,0X00};
unsigned char * SSLName[3]={PAGEQ0, PAGEQ1, PAGEQ2};

```

```

/*****
* E) Function Prototypes *
*   SetMode(Mode) *
*   delay(n) *
*   resetchip() *
*   SingleCmd(i) *
*   SingleData(i) *
*   SetRAMAddr(Page,Col) *
*   SetContrast(Gain,Step) *
*   InitDisplay() *
*   clearRAM(start,stop) *
*****/

```

```

void SetMode(unsigned char Mode)
{
    WC_CSH = 0x7d; //for demoboard MCU
    WC_CSL = 0x3d; /* Parallel mode */
    WD_CSH = 0x7f; /* Master mode */
    WD_CSL = 0x3f; /* 6800 mode */
    RES_CSH = 0xff;
    RES_CSL = 0xef;
}

```

```

void delay(unsigned int n)                /* wait n seconds*/
{
    unsigned int i;
    unsigned int j;
    for (i=0;i<500;i++)
        for (j=0;j<n*2;j++) { ;}
}

void resetchip()
{
    unsigned int i;
    controlport=RES_CSL;
    for (i=1;i<500;i++);
    dataport = 0xff;
    dataport = 0xff;
    dataport = 0xff;
    for (i=1;i<500;i++);
    controlport=RES_CSH;
}

void SingleCmd(unsigned char i)
/* send the value in the accumulator to LCD driver as a command*/
{
    dataport=i;
    controlport=WC_CSH;
    controlport=WC_CSL;
    controlport=WC_CSH;
}

void SingleData(unsigned char i)
/* send the value in the accumulator to LCD driver as a command*/
{
    dataport=i;
    controlport=WD_CSH;
    controlport=WD_CSL;
    controlport=WD_CSH;
}

void SetRAMAddr (unsigned char Page, unsigned char Col)
{
    unsigned char temp;
    temp = 0x0f & Page;
    SingleCmd(PageAddr | temp);
    temp = 0x0f & (Col >> 4);
    SingleCmd(ColAddrHi | temp);
    temp = 0x0f & Col;
    SingleCmd(ColAddrLo | temp);
}

void SetContrast(unsigned char Gain, Step) {
    SingleCmd(IntRegRatio | (0x0f & Gain));
    SingleCmd(ContCtrlReg);
}

```

```

    SingleCmd((0x3f & Step));
}

void InitDisplay() {
    SingleCmd(DisplayOff);
    SingleCmd(SegRemapOn);
    SingleCmd(ComRemapOn);
    SetContrast(6, 0x36);
    SingleCmd(PwrCtrlReg | IntVolBstr | IntReg | OPampBuffer);
    SingleCmd(DisplayOn);
}

```

```

void clearRAM(unsigned char startpage, stoppage)
{
    unsigned char i, j;
    for (j=startpage; j<stoppage; j++) {
        SetRAMAddr(j, 0x00);
        for (i=0; i<132; i++) {
            SingleData(0x00);
        }
    }
}

```

```

/*****
* F) Main Function *
*****/

```

```

main()
{
    unsigned char i, j;
start: SetMode(PS);
    resetchip();
    clearRAM(0,9);
    /*****/
    /* Show device demo */
    /*****/
    InitDisplay();
    SingleCmd(DisplayOn);

    /*****/
    /* Display SSL Chinese */
    /* Message */
    /*****/
    SingleCmd(DisplayStart);
    for (j=0; j<SSLNameNo; j++) {
        SetRAMAddr(j+2, 0x18);
        for (i=0; i<48; i++) {
            SingleData(SSLName[j][i]);
        }
    }
    delay(100);
    goto start;
}

```

Solomon reserves the right to make changes without further notice to any products herein. Solomon makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Solomon assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Solomon does not convey any license under its patent rights nor the rights of others. Solomon products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of unintended or unauthorized application, Buyer shall indemnify and hold Solomon and its offices, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Solomon was negligent regarding the design or manufacture of the part.