User Manual for DC6688SLP (Former name: SL Programmer Board ver1.1)

Document Revision 3.0 Aug, 2008

Revision History

The following table shows the revision history for this document.

Date	Document Revision	Remark	Edited By	Reviewed By
Nov, 2004	1.0	Preliminary for SL Programmer Board ver1.0	-	-
Nov, 2004	1.1	Preliminary for SL Programmer Board ver1.1	-	-
Dec, 2004	1.2	Modify section 3.1, 4, 5.1, 5.2	-	-
Feb, 2005	1.3	added description on DC6688F05S	-	-
Feb, 2005	1.4	fix LED flashing for the board, use configurator file “fpga_v1_1.mcs”	-	-
July, 2005	1.5	Change to use configurator file “fpga_v1_5.mcs”	-	-
Aug, 2005	1.6	Change to use configurator file “fpga_v1_10.mcs”	-	-
Jan, 2005	1.8	Change to use configurator file “fpga_v1_11.mcs”	-	-
July, 2006	1.9	Change to use configurator file “fpga_v2_0.mcs”	-	-
July, 2006	2.0	a) Revise section 2, 3.2, 5.1, 5.2 b) Change to use configurator file “fpga_v2_2.mcs”	-	-
Sept, 2006	2.1	Change to use configurator file “fpga_v2_4.mcs”	-	-
Jan, 2007	2.2	1) Change to use configurator file “fpga_v2_5.mcs” 2) Revise section 5.2	-	-
May, 2007	2.3	1) Change to use configurator file “fpga_v2_6.mcs”	-	-
July, 2007	2.4	1) Change to use configuratot file “fpga_v2_7.mcs” 2) Hardware modified	-	-
Oct, 2007	2.5	1) Change to use configurator file “fpga_v2_8.mcs”	-	-
May, 2008	2.6	1) Change to use configurator file “fpga_v3_0.mcs”	-	-
Jul, 2008	2.7	1) Change to use configurator file “fpga_v3_0.mcs” or “fpga_v3_1.mcs”	-	-
Jul, 2008	2.8	1) Remove Appendix A 2) Remove Glossary section 3) Added edited by & reviewed by at revision history table 4) Some typing error	Ken Yeung	Danny Ho
July, 2008	2.9	Change the name of the board to “DC6688SLP”	Danny Ho	Ken Yeung
Aug, 2008	3.0	Change the name of the board to “DC6688SLP” in all section	Danny Ho	Ken Yeung

Contents

1 Introduction

The Objective of this document is to provide the user a quick start to evaluate DC6688SLP. This board is applicable to the following:

1) DC6688F05S
2) DC6688FSA
3) DC6688FSB
4) DC6688FLB
5) DC6688FL32A
6) DC6688FLX
7) DC6688FSX

To program the DC6688 family, it involves 2 steps:

a) Download code from PC to DC6688SLP refer to section 3 for detail
b) Program DC6688 family in each device refer to section 4 for detail

Section 5 describes the hardware in detail.
Section 6 summarizes the difference between versions of configurator file in detail.

It should be noted that this document applies only to DC6688SLP with the configurator 'XC18V01PC' programmed with configurator file "fpga_v3_0.mcs" or "fpga_v3_1.mcs" or with the configurator 'XC18V01PC' marked with "v3.0", or "v3.1".

2 Download code from PC to DC6688SLP

To download the code to DC6688SLP, it involves the hardware and software setup. Setting up the hardware first, and then use software to control the download.

2.1 Hardware setup

The procedure is listed below:

a) Connect the board via a parallel port cable to PC as shown in diagram 3.1
b) Attach a fixed power supply to the power-connector at 'J13'. An unregulated +7.5V up to +12V/800mA power source can be used to supply the power of the board. The correct polarity of the power plug is shown in diagram 3.2.
c) Turn on the switch 'S1'. The LED 'D14' should turn on to indicate the board is ready. The on/off position is shown in diagram 3.3

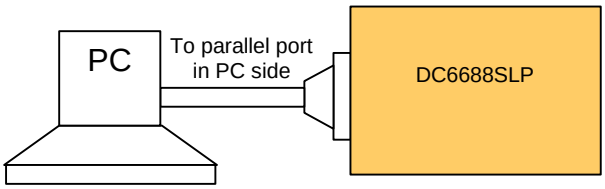


Diagram 3.1 Connection between SL Programmer board and PC

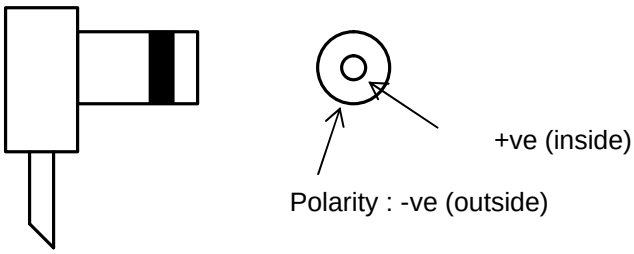


Diagram 3.2 Polarity of DC Jack

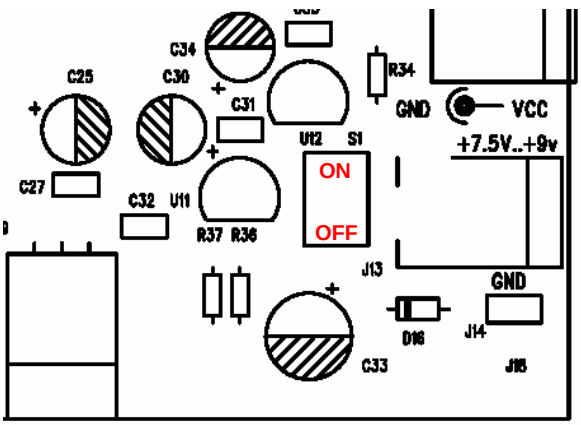


Diagram 3.3 On/Off position of power switch

2.2 Software setup

The software setup needs ‘Dragonchip ISP Programmer’ with version 4.1.0 or higher. Details on setup and how to use the software to download code refer to document “Dragonchip In-system programmer Manual”. This software downloads the code to DC6688SLP.

After downloading the code, the checksum displayed on the 7-segment LEDs is not updated until turning off/on the power.

3 Programming devices

The procedure is listed below:

1. Connect the board to each device as shown in diagram 4.1. Use connector ‘J1’ to ‘J10’ to connect to each device. The pin assignment on each connector is shown in diagram 4.2.

   The SL Programmer board can support up to 10 devices. For example, if only 6 devices are needed to program, then the connectors ‘J1’ to ‘J6’ are used.

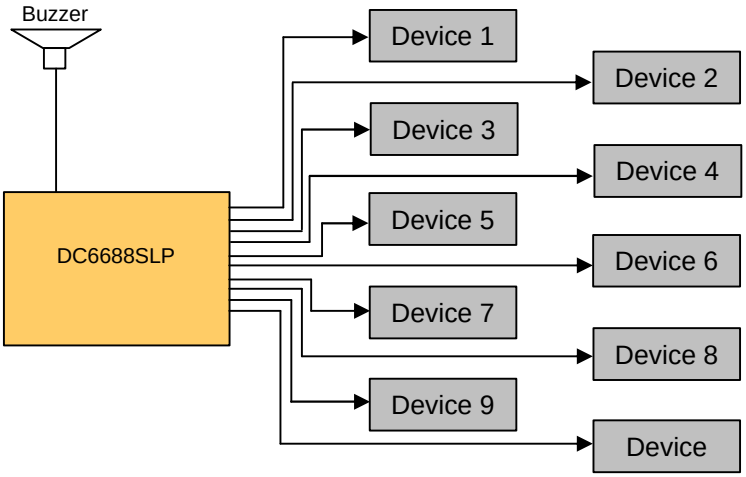


Diagram 4.1 Connection between SL

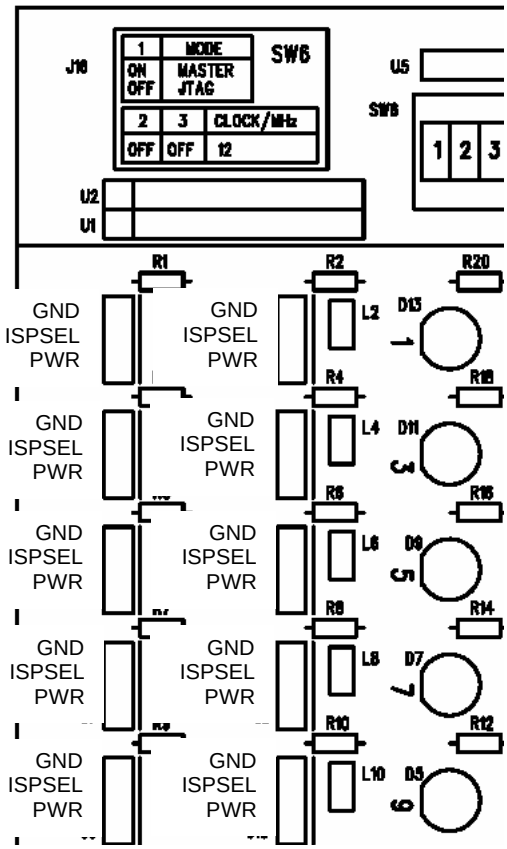


Diagram 4.2 Pin assignments for each connector ‘J1’ – ‘J10’

2. Connect a 3VDC buzzer to ‘J15’ on the board. This buzzer produces beep sounds. It provides an alternative to user a notice that the 10 devices finish programming no matter success or failure. Below is a table to show how to use buzzer according to different number of devices to be program for each turn:

For example: there are 2 devices to be programmed for each turn. It can be found in the table that connector ‘J1’ – ‘J2’ will be used. A connection between the point ‘BUZZAR_TEST_PAD’ and LED3 should be made. Then each time finished programming, the buzzer beeps sound immediately.

No. of devices to program	connector to device	Buzzer connection[1]
1	‘J1’	a) open the connection on ‘R58’, and b) make a connection between the point ‘BUZZAR_TEST_PAD’ and LED2
2	‘J1’ - ‘J2’	a) open the connection on ‘R58’, and b) make a connection between the point ‘BUZZAR_TEST_PAD’ and LED3
3	‘J1’ - ‘J3’	a) open the connection on ‘R58’, and b) make a connection between the point ‘BUZZAR_TEST_PAD’ and LED4
4	‘J1’ - ‘J4’	a) open the connection on ‘R58’, and b) make a connection between the point ‘BUZZAR_TEST_PAD’ and LED5
5	‘J1’ - ‘J5’	a) open the connection on ‘R58’, and b) make a connection between the point ‘BUZZAR_TEST_PAD’ and LED6
6	‘J1’ - ‘J6’	a) open the connection on ‘R58’, and b) make a connection between the point ‘BUZZAR_TEST_PAD’ and LED7
7	‘J1’ - ‘J7’	a) open the connection on ‘R58’, and b) make a connection between the point ‘BUZZAR_TEST_PAD’ and LED8
8	‘J1’ - ‘J8’	a) open the connection on ‘R58’, and b) make a connection between the point ‘BUZZAR_TEST_PAD’ and LED9
9	‘J1’ - ‘J9’	a) open the connection on ‘R58’, and b) make a connection between the point ‘BUZZAR_TEST_PAD’ and LED10
10	‘J1’ - ‘J10’	short the connection on ‘R58’

Remarks:
[1] R58, ‘BUZZAR_TEST_PAD’, LED1 - LED10 are shown on schematics ‘CASE’

3. Check the settings on the board. Details refer to section 5.2
4. Attach a fixed power supply to the power-connector at 'J13'. An unregulated +7.5V up to +12V/800mA power source can be used to supply the power of the board. The correct polarity of the power plug is shown in diagram 3.2.
5. Turn on the switch 'S1'. The LED 'D14' should turn on to indicate the board is ready. The 7-segment LED shows up the checksum.
6. After displayed the checksum, press 'Auto' button to do programming.
7. After hearing the buzzer beeping sounds, it indicates programming finished. All the devices can unplug without switching off power. If the devices need to re-program('Auto' button), verify('Verify' button) or read checksum('Checksum' button), the devices should unplug and plug again.
8. Repeat step 1 to 7 for the second turn programming. Step 2 – step 5 can be skipped. When reaching step 6, press the 'Auto' button no matter the programmer still undergo last turn programming. For example, assuming that there is only 1 device to program for each turn. The programmer still goes to program the rest of 9 devices. When pressing 'Auto' button, the device being programming will continue until finish. The programmer then goes to restart from device 1.

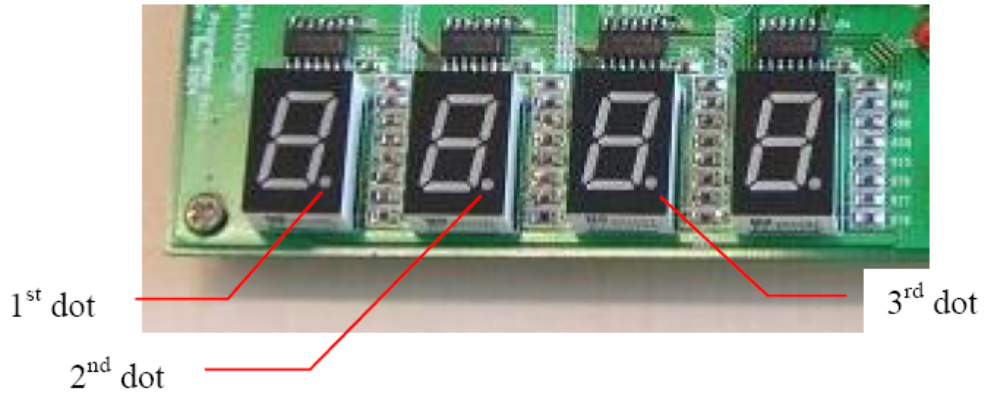
4 Hardware description

4.1 Function keys

4.1.1 DC6688FSA

Keys	Description
Auto	This key performs a) write code to device, b) read back the code from device, c) verify the code[2], d) if 'Lock' switch is on, lock the device preventing from reading back From step a to d, the LED corresponding to a device will flash. If the result success, the LED turns on. If the result fails, the LED turns off.
Verify	This key performs a) read back the code from device, b) verify the code[2] From step a to b, the LED corresponding to a device will flash. If the result success, the LED turns on. If the result fails, the LED turns off.
CheckSum[1]	This key performs a) read back the Model, Version, Checksum stored at 0x3A ~ 0x3F in data flash memory on device 1(i.e. connector "J1") b) display the Model, Version, Checksum on the four 7-segment LEDs On step b, the four 7-segment LEDs will display the Model, Version and Checksum in the order and then recycle again[3].
Reset	This key, when pressed, will reset the board to initial state without turn off/on the power. When ready, the LED 'D14' turns on. It is recommended to use power switch 'S2' to do reset. This is also equivalent to the function of 'Reset' button.

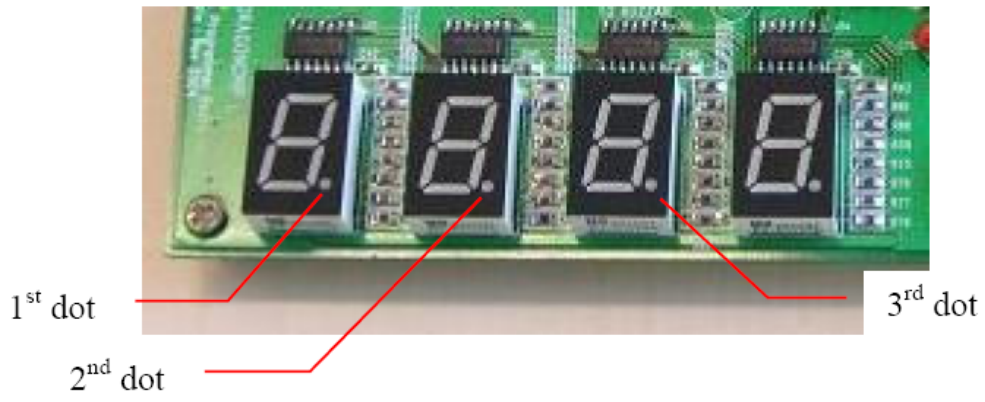
Remarks:
[1] Only device 1 will be read for checksum.
[2] Verify the code by comparing byte by byte. If at least one byte fails, the LED will be off.
[3] The Model/Version/Checksum display is shown below:
When the 1st dot is on, 'Model' is displaying. When the 2nd dot is on, 'Version' is displaying. When the 3rd dot is on, 'Checksum' is displaying.



4.1.2 DC6688FL32A

Keys	Description
Auto	This key performs a) write code to device, b) read back the code from device, c) verify the code[2], d) if 'Lock' switch is on, lock the device preventing from reading back From step a to d, the LED corresponding to a device will flash. If the result success, the LED turns on. If the result fails, the LED turns off.
Verify	This key performs a) read back the code from device, b) verify the code[2] From step a to b, the LED corresponding to a device will flash. If the result success, the LED turns on. If the result fails, the LED turns off.
CheckSum[1]	This key performs a) read back the Model, Version, Checksum in flash memory on device 1 (i.e. connector "J1") b) display the Model, Version, Checksum on the four 7-segment LEDs On step b, the four 7-segment LEDs will display the Model, Version and Checksum in the order and then recycle again[3].
Reset	This key, when pressed, will reset the board to initial state without turn off/on the power. When ready, the LED 'D14' turns on. It is recommended to use power switch 'S2' to do reset. This is also equivalent to the function of 'Reset' button.

Remarks:
[1] Only device 1 will be read for checksum.
[2] Verify the code by comparing byte by byte. If at least one byte fails, the LED will be off.
[3] The Model/Version/Checksum display is shown below:
When the 1st dot is on, 'Model' is displaying. When the 2nd dot is on, 'Version' is displaying. When the 3rd dot is on, 'Checksum' is displaying.



4.1.3 DC6688F05S

Keys	Description
Auto	Refers to section 4.1.2 for details
Verify	Refers to section 4.1.2 for details
CheckSum[1]	Refers to section 4.1.2 for details
Reset	Refers to section 4.1.2 for details

Remarks:
[1] Only device 1 will be read for checksum.

4.1.4 DC6688FLX

Keys	Description
Auto	Refers to section 4.1.2 for details
Verify	Refers to section 4.1.2 for details
CheckSum[1]	Refers to section 4.1.2 for details
Reset	Refers to section 4.1.2 for details

Remarks:
[1] Only device 1 will be read for checksum.

4.1.5 DC6688FSX

Keys	Description
Auto	Refers to section 4.1.2 for details
Verify	Refers to section 4.1.2 for details
CheckSum[1]	Refers to section 4.1.2 for details
Reset	Refers to section 4.1.2 for details

Remarks:
[1] Only device 1 will be read for checksum.

4.1.6 DC6688FSB

Keys	Description
Auto	Refers to section 4.1.1 for details
Verify	Refers to section 4.1.1 for details
CheckSum[1]	Refers to section 4.1.1 for details
Reset	Refers to section 4.1.1 for details

Remarks:
[1] Only device 1 will be read for checksum.

4.1.7DC6688FLB

Keys	Description
Auto	Refers to section 4.1.2 for details
Verify	Refers to section 4.1.2 for details
CheckSum[1]	Refers to section 4.1.2. for details
Reset	Refers to section 4.1.2 for details

Remarks:
[1] Only device 1 will be read for checksum.

4.2 Hardware settings

DIP switch 'SW6' have the following setting:
The following settings apply to only configurator XC18V01 with "fpga_v3_0.mcs" or "fpga_v3_1.mcs", or marked with v3.0, or v3.1.

<table>
<tr><th>Position</th><th>Device</th></tr>
<tr><td>1</td><td>Mode of XC2S100
On: Master (default)
Off: JTAG
User should not modify this setting</td></tr>
<tr><td>2, 3</td><td>Frequency selection:
<table>
<tr><td>SW2</td><td>SW3</td><td></td></tr>
<tr><td>ON</td><td>ON</td><td>Fosc/12 = 4MHz</td></tr>
<tr><td>ON</td><td>OFF</td><td>Fosc/8 = 6MHz</td></tr>
<tr><td>OFF</td><td>ON</td><td>Fosc/1</td></tr>
<tr><td>OFF</td><td>OFF</td><td>Fosc/4 = 12MHz (default)</td></tr>
</table></td></tr>
<tr><td>4, 5, 6, 7</td><td>Program device selection
<table>
<tr><td>SW4</td><td>SW5</td><td>SW6</td><td>SW7</td><td></td></tr>
<tr><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>DC6688F05S/SP/SN</td></tr>
<tr><td>ON</td><td>ON</td><td>ON</td><td>OFF</td><td>DC6688F2SB/SBT
DC6688F4SB/SBT/SA</td></tr>
<tr><td>ON</td><td>ON</td><td>OFF</td><td>ON</td><td>DC6688F8SB/SBT/SA</td></tr>
<tr><td>ON</td><td>ON</td><td>OFF</td><td>OFF</td><td>DC6688F14SB/SBT
DC6688F16SB/SBT/SA</td></tr>
<tr><td>ON</td><td>OFF</td><td>ON</td><td>ON</td><td>DC6688F24SB/SBT/SA
(default)</td></tr>
<tr><td>ON</td><td>OFF</td><td>ON</td><td>OFF</td><td>DC6688F30SB/SBT/SBE/SA</td></tr>
<tr><td>ON</td><td>OFF</td><td>OFF</td><td>ON</td><td>DC6688FE4</td></tr>
<tr><td>ON</td><td>OFF</td><td>OFF</td><td>OFF</td><td>DC6688FE8</td></tr>
<tr><td>OFF</td><td>ON</td><td>ON</td><td>ON</td><td>DC6688FE16</td></tr>
<tr><td>OFF</td><td>ON</td><td>ON</td><td>OFF</td><td>DC6688FE24</td></tr>
<tr><td>OFF</td><td>ON</td><td>OFF</td><td>ON</td><td>DC6688FL32A
DC6688FL16B/BT</td></tr>
<tr><td>OFF</td><td>ON</td><td>OFF</td><td>OFF</td><td>reserved</td></tr>
<tr><td>OFF</td><td>OFF</td><td>ON</td><td>ON</td><td>DC6688F62SX/SXE/SXR</td></tr>
<tr><td>OFF</td><td>OFF</td><td>ON</td><td>OFF</td><td>DC6688FL64X/XE</td></tr>
<tr><td>OFF</td><td>OFF</td><td>OFF</td><td>ON</td><td>reserved</td></tr>
<tr><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>reserved</td></tr>
</table></td></tr>
<tr><td>8</td><td>Lock
This key decides to lock DC6688FSA, DC6688FSB, DC6688FL32A, DC6688FLB and DC6688F05S when pressing 'Auto' button
On: lock(program flash memory) enable (default)
Off: no lock

This key decides to lock DC6688F62SX/SXE/SXR when pressing 'Auto' button
On: lock(program flash memory only) enable (default)
Off: no lock

This key decides to lock DC6688FL64X/XE when pressing 'Auto' button
On: lock(program flash memory and data flash memory) enable (default)
Off: no lock</td></tr>
</table>

4.3 Where does the code store?

When PC downloads the code to the DC6688SLP, the code is stored to the flash memory ‘U3’ (AT29LV512).

4.4 Maximum Power to device

The SL Programmer Board ver1.0 can support up to 10 devices. The maximum current to supply by regulator ‘U10’ to the 10 devices simultaneously is 800mA.

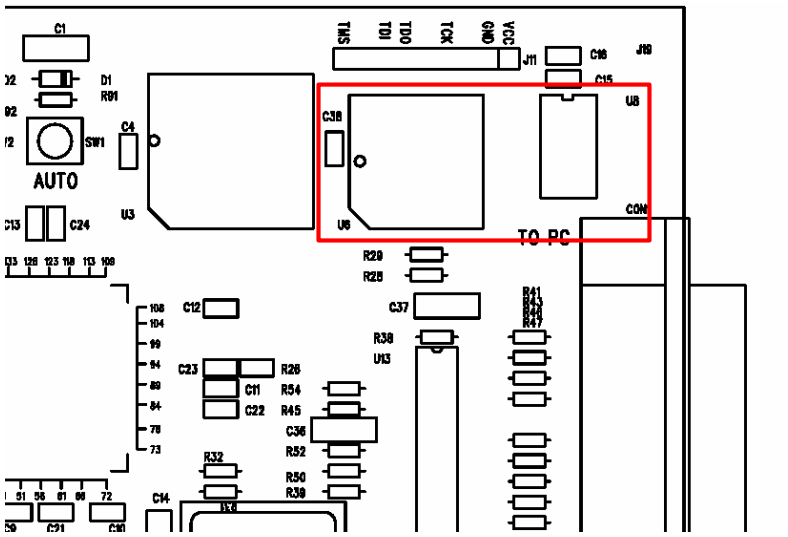
4.5 7-segment LED Display

The SL Programmer Board can provide display of checksum to identify the source code.

When power on, the display will show up the checksum for the code stored at the flash memory ‘U3’ (AT29LV512/AT29LV010A). The checksum is consistent with the one shown in the software ISP Programmer V3.5.5 or higher.

4.6 Configurator to Xilinx XC2S100

There are two areas, ‘U6’ and ‘U8’, to put configurator on the board as shown in red below. The xilinx XC2S100 only need one configurator. It is recommended to choose ‘U6’ as this is PLCC package type.



4.7 Connection to device

AWG#26 wire is recommended for power line. The wire, which the thickness is too thin, is not appropriate.

AWG#28 wire is recommended for ISPSEL line.

4.8 PCB layout consideration

The Xilinx XC2S100 need clock to run. The clock is provided by can oscillator ‘U7’. This clock signal line from ‘U7’ to pin 88 of Xilinx XC2S100 should be surrounded by ground line as shown on the board.

To reduce noise on the board, both sides of the 2-layer PCB should be poured with ground copper wherever space is available as shown on the board.

5 Programming Time

5.1 DC6688FSA/FSB

The maximum programming time for each device operating at 12MHz is shown in the table below:

Program Size/kbytes	Programming time/s
2	0.850
4	0.850
8	1.340
14	2.330
16	2.330
24	3.320
30	4.070

5.2 DC6688FL32A/DC6688FLB

The maximum programming time for each device operating at 12MHz is 4.400s

5.3 DC6688F05S

The maximum programming time for each device operating at 4MHz is 3.840s

5.4 DC6688FSX

The maximum programming time for each device operating at 12MHz is shown in the table below:

Devices	Programming time/s
DC6688F62SX/SXE/SXR	8.000

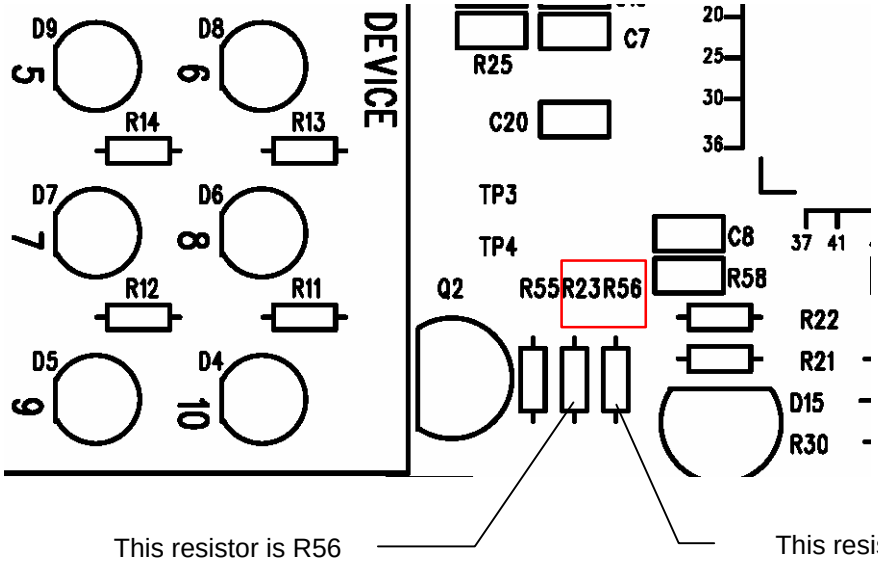
5.5 DC6688FLX

The maximum programming time for each device operating at 12MHz is shown in the table below:

Devices	Programming time/s
DC6688FL64X/XE	8.240

6 Correction on PCB

The label “R23” and “R56” on silkscreen as shown in red below should interchange.



Copyright Notice

Disclaimer

Dragonchip Ltd. assumes no responsibility for any errors contained herein.

Dragonchip Ltd.
TEL: (852) 2776-0111
FAX: (852) 2776-0996
http://www.dragonchip.com