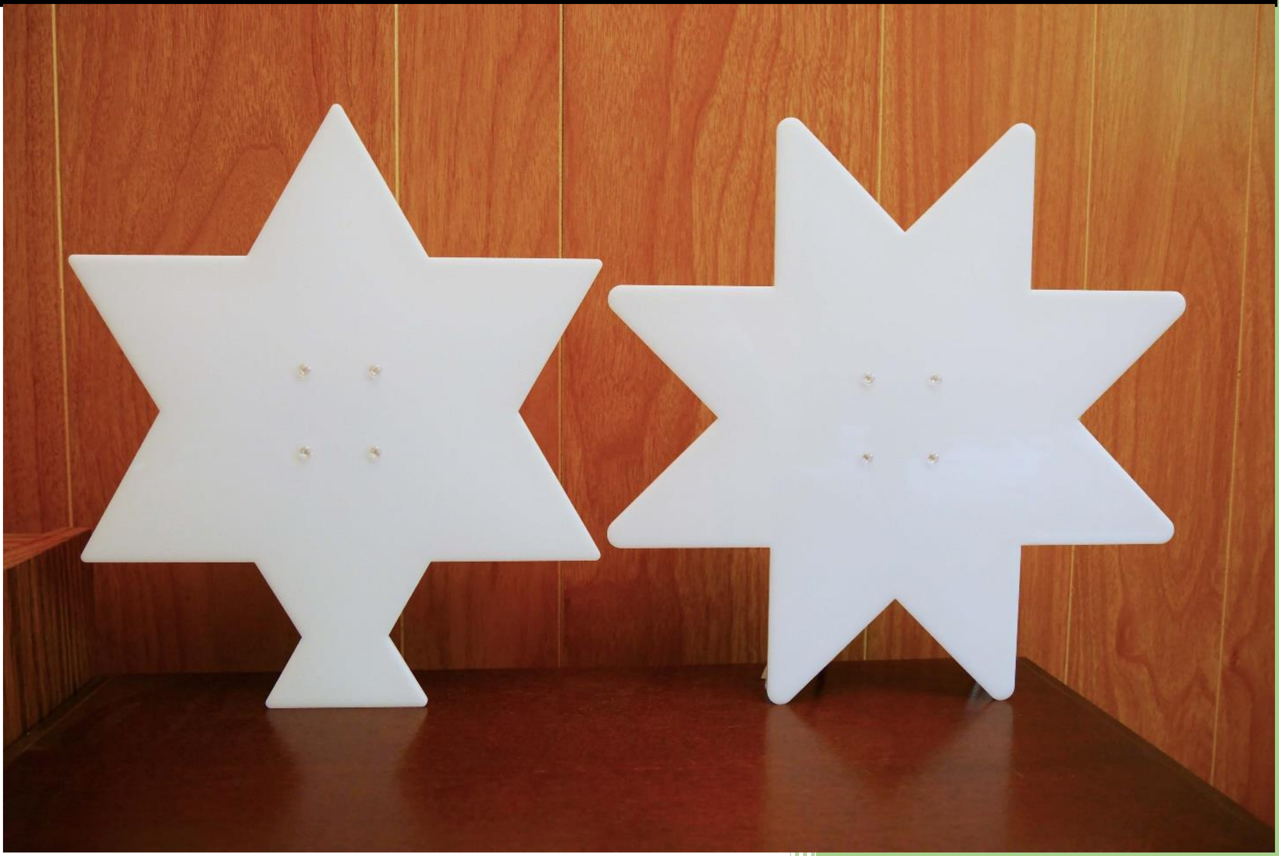
Starlite User's Manual



Revision: A.2

Grant Grummer

9/29/2020

Starlite User’s Manual

Table of Contents

Introduction

Welcome to the Starlite User’s Manual. Starlite can be constructed as a 6 or 8-point star. They’re approximately 10 inches tall. They consist of outer and inner translucent acrylic star shapes, tri-color LED strips, a circuit board, and the hardware to connect them together. An optional 3D printed back cover can also be made, for a more finished look.

Starlite uses the tri-color LED strips to produce different light displays. Starlite is programmable, so the displays are only limited by your imagination. A set of default light displays along with an introduction to Starlite can be viewed at either of the following links:

https://youtu.be/trDGcNV_JYw

https://drive.google.com/file/d/1uRoxA3LyNXThGwm_Sx0gEvdsMLY5Yu8b/view?usp=sharing

A small set of display specific instructions are used to create different display patterns. There are up to 16 possible instructions. Not all of these instructions are defined at the time of this writing and are available for those who wish to dive deeper into this project.

The circuit board contains all the electronic hardware needed to drive the LED strips. The brain of the hardware is a Lattice iCE40 Ultra Plus 5K FPGA (Field Programmable Gate Array). It, along with programming and other support logic, is on the UPduino v3.0 low cost daughter board. It's mounted onto the StarLiteMain board. The StarLiteMain board contains switches, power connector and interface hardware.

Starlite is controlled via three switches. The power switch enables/disables power from the power input micro USB connector. The upper pushbutton switch selects one of four display instruction sets. Each set can have up to 256 instructions. The lower pushbutton switch selects different brightness levels.

Starlite Design Environment

In order to program Starlite, one must first setup a design environment. Refer to the "Starlite Installation and Setup" document for instruction on how to set it up. The rest of this document assumes this step has been completed.

Quick Custom Displays

A quick way to get started with making your own custom displays is to use the default instruction set to build new display instruction sets. One doesn't need to know anything about FPGAs to do this. Use the supplied "LED_StarLiteUP_Instruction" spreadsheet to create each instruction. Use a text editor to combine the new instructions into a set of instructions. The text editor needs to be able to select columns of text like "Notepad++". Use the supplied example display instruction sets, in the "patterns" folder, to see how this works. Remember to end each instruction set with the Finish instruction!

Once an instruction set has been completed, it needs to be split in half vertically. Refer to the supplied example display *_hi.mem and *_lo.mem instruction sets to see how the split looks. Splitting the instructions is necessary because the FPGA memories are 16-bits wide, while instructions are 32-bits wide.

Replace the current instruction set(s) with your new instruction set(s) in the "instr_ram_full6.list" or "instr_ram_full8.list" (depending on whether you're building the 6 or 8-point star). Then run "memloader6.bat" or "memloader8.bat" batch file. Once that completes successfully, run "bitmapper.bat".

Programming Starlite

There are many programmers that can be used to program Starlite. One is the Radiant Programmer Standalone and the other is the Diamond Programmer Standalone. They're both very similar. The following steps can be used with the Radiant Programmer to program Starlite. The Diamond Programmer steps may be slightly different.

1. Connect the micro USB port on the UPDuino v3.0 to your PC
2. Open the Radiant Programmer
3. Click `Detect Cable` then `OK`
4. After scanning, select `Generic JTAG Device` and `Select iCE40 UltraPlus`
5. Under `Device` click iCE40UP3K and change it to iCE40UP5K
6. Under `Operation` double click `Fast Program` and change `Target Memory: ` to `External SPI Flash Memory`
7. Select your `*.hex` programming file under `Programming file`.
8. Configure the following `SPI Flash Options`
   a. Winbond
   b. W25Q32JV
   c. 208mil 8-pin SOIC
9. Click `Load from File` under `SPI Programming` to get load size
10. Click OK
11. Click `Run` -> `Program Device`

Exploring and/or Editing the Starlite design

The code base for Starlite is contained in the download (clone) from GitHub (as explained in "LED_StarLiteUP_Instruction"). Feel free to explore the design and customize it to your liking. If changes were made and you'd like to build the design, synthesis will need to be run first.

Run Synthesis

Double click "Run Lattice …". Once it completes successfully, move on to the next section.

Run the Build

Right click on "Run P&R". Select "Run Router". This will run the steps before "Run Router" and "Run Router" too. There's no need to run "Bitmap" at this point. Once these complete successfully, it's time to insert display patterns. This is covered in the previous Quick Custom Displays section. Refer to it for directions. "Bitmap" can be run using the "bitmapper.bat" file outside of iCEcube2 (as noted in the Quick Custom Displays section) or one can run "Bitmap" in iCEcube2. This completes the build. Refer to the previous Programming Starlite section to program Starlite.

Instructions

As previously noted, there are up to 16 possible instructions. Most are specifically tailored for RGB LED strings. Instructions are 32-bits long. All instructions start with a command. Some commands are followed by additional fields. A detailed description of each instruction is presented in the following sections. Note all bits of the instructions are zeroed at powerup unless otherwise noted.

The LED_StarLiteUP_Instruction spreadsheet provides an easy way to create instructions. It allows instructions to be build using decimal number and automatically provides the hexadecimal value (of the instruction) needed by the FPGA.

Instruction 0: Blackout Display

Blackouts the entire display.

31 28	27				0
CMD0	Unused	Unused	Unused	Unused	Unused

Bits	Name	Function
[31:28]	CMD0	Blackouts all the LEDs in the string(s). All bits must be zero.
[27:0]	Unused	Not defined

Instruction 1: Delay

Adds a programmable time delay. Each additional count adds 5.12 mSeconds. The maximum count is 255 (0xFF). A count a zero creates a random delay.

31 28	27 20	19			0
CMD1	DLY	Unused	Unused	Unused	Unused

Bits	Name	Function
[31:28]	CMD1	Programmable delay. Value = 1
[27:20]	DLY	Delay count. Max = 255 (0xFF). All zeros create a random count.
[19:0]	Unused	Not defined

Instruction 2: Finish

All instruction sets must end with a finish command!

31 28	27				0
CMD2	Unused	Unused	Unused	Unused	Unused

Bits	Name	Function
[31:28]	CMD2	Informs the FPGA that the present instruction set is finished. Value = 2
[27:0]	Unused	Not defined

Instruction 3: Set Random LED

Used to set one LED at a random location/address and/or color. The random color is selected from the 16 colors in the color palette (see Instruction 9: Set Color Palette).

31 28	27|26| 25 20	19 12	11		1 | 0
CMD3	U | R | ADDR	BRITE	Unused	Unused	Unused|N

Bits	Name	Function
[31:28]	CMD3	Sets one LED. Value = 3
[27]	U	Unused, not defined
[26]	R	Random LED address 0 = bits [25:0] specify address of LED 1 = random LED address generated
[25:20]	ADDR	Specific LED address when [26] = 0. Values 0 to 63 (0x3F)
[19:12]	BRITE	Set the brightness of the LED. Values 0 to 255 (0xFF)
[11:1]	Unused	Not defined
[0]	N	0 = use the same color as used in the last CMD3. That color was the one last selected from the color palette. 1 = use the next color in the color palette.

Instruction 4: Set LED

Used to set one LED’s address and color.

31 28	27 26 | 25 20	19 12	11 8	7 4	3 0
CMD4	U | ADDR	BRITE	BLUE	GREEN	RED

Bits	Name	Function
[31:28]	CMD4	Sets one LED. Value = 4
[27:26]	U	Unused, not defined
[25:20]	ADDR	Specific LED address. Values 0 to 63 (0x3F)
[19:12	BRITE	Set the brightness of the LED. Values 0 to 255 (0xFF)
[11:8]	BLUE	Sets the amount of blue. Value 0 to 15 (0xF)
[7:4]	GREEN	Sets the amount of green. Value 0 to 15 (0xF)
[3:0]	RED	Sets the amount of red. Value 0 to 15 (0xF)

Instruction 5: Chase

Adds an offset to all LED addresses. Increments or decrements that address along with their brightness and color. The offset wraps around at maximum numbers of LEDs per string.

31 28	27	14|13|12	11		0
CMD5	Unused	Unused | Z | ID	Unused	Unused	Unused

Bits	Name	Function
[31:28]	CMD5	Increments or decrements every LED’s address by one. Value = 5
[27:14]	Unused	Not defined
[13]	Z	0 = do not zero LEDs address offset

		1 = zeros LEDs address offset (ID value ignored)
[12]	ID	0 = decrement offset 1 = increment offset
[11:0]	Unused	Not defined

Instruction 6: Reserved

Do not use. It's used in other versions of this project.

31 28	27				0
CMD6	Unused	Unused	Unused	Unused	Unused

Bits	Name	Function
[31:28]	CMD6	Reserved. Value = 6
[27:0]	Unused	Not defined

Instruction 7: Fade In and Out

Fades all LEDs brightness. Increments and then decrements all LEDs brightness.

<table>
<tr><td>31 28</td><td>27 26|25 24|23 22|21 20</td><td>19 13|12</td><td>11</td><td></td><td>0</td></tr>
<tr><td>CMD7</td><td>U | LLIM| ULIM| SPD</td><td>Unused | EF</td><td>Unused</td><td>Unused</td><td>Unused</td></tr>
</table>

Bits	Name	Function
[31:28]	CMD7	Increments or decrements every LED's address by one. Value = 7
[27:26]	U	Unused, not defined
[25:24]	LLIM	Limits how low the brightness value can go
[23:22]	ULIM	Limits how high the brightness value can go
[21:20]	SPD	Speeds up the incrementing and decrementing
[19:13]	Unused	Not defined
[12]	EF	0 = disable fading once decrement reaches its lowest value 1 = enables fading starting at its lowest value
[11:0]	Unused	Not defined

Instruction 8: Loop

Loop the previous x number of instructions.

31 28	27 20	19 12	11		0
CMD8	LOOPS	QTY	Unused	Unused	Unused

Bits	Name	Function
[31:28]	CMD8	Reserved. Value = 8
[27:20]	LOOPS	Sets the number of times a loop is executed. A value of 255 (0xFF) is a continuous loop.

[19:12]	QTY	Quantity of instruction to repeat. Subtract the start instruction from the loop instruction line number.
[11:0]	Unused	Not defined

Instruction 9: Set Color Palette

Storage location for 16 definable colors. Used in conjunction with Instruction 3: Set Random LED. A set of predefined colors are loaded at powerup. Use this instruction to change them.

31 28	27 24	23 20	19 12	11 8	7 4	3 0
CMD9	Unused	CNUM	Unused	BLUE	GREEN	RED

Bits	**Name**	**Function**
[31:28]	CMD9	Sets one LED. Value = 9
[27:24]	Unused	Not defined
[23:20]	CNUM	Specific palette address. Values 0 to 15 (0xF)
[19:12]	Unused	Not defined
[11:8]	BLUE	Sets the amount of blue. Value 0 to 15 (0xF)
[7:4]	GREEN	Sets the amount of green. Value 0 to 15 (0xF)
[3:0]	RED	Sets the amount of red. Value 0 to 15 (0xF)

Instruction 10 (0xA): Instruction Offset & Block

This instruction does two somewhat unrelated, Starlite specific, functions. The first is to offset the instruction given to each LED string by one. The second is to block any number of LED strings from getting an instruction. This blocking is only available for up to the first 12 points.

31 28	27	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD10	Unused	Unused	PO	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

Bits	**Name**	**Function**
[31:28]	CMD10	Increments or decrements every LED's address by one. Value = 10 (0xA)
[27:13]	Unused	Not defined
[12]	PO	0 = disable point offset 1 = enable point offset
[11]	BLK11	0 = unblock point 11 1 = block point 11
[10]	BLK10	0 = unblock point 10 1 = block point 10
⋮	⋮	⋮
[0]	BLK0	0 = unblock point 0 1 = block point 0

Instruction 11 (0xB): Rainbow

Transitions through the 12-bit color spectrum.

31 28	27	13	12	11		0
CMD11	Unused	Unused	ER	Unused	Unused	Unused

Bits	Name	Function
[31:28]	CMD11	Increments or decrements every LED's address by one. Value = 11 (0xB)
[27:13]	Unused	Not defined
[12]	ER	0 = disable the transitions 1 = enables the transitions
[11:0]	Unused	Not defined

Instruction 12 (0xC): Reserved

Do not use. It's used in other versions of this project.

31 28	27				0
CMD12	Unused	Unused	Unused	Unused	Unused

Bits	Name	Function
[31:28]	CMD12	Reserved. Value = 12 (0xC)
[27:0]	Unused	Not defined

Instruction 13 (0xD): Automatic Timeout

Automatically turns the LEDs on and off according to the following settings. The instruction set in use before the time "off" will resume once the time "on" is attained. One should set this up before issuing display instructions.

31 28	27 25	24 20	19 17	16 12	11	5	4 0
CMD13	Unused	HON	Unused	HOFF	Unused	Unused	HDLY

Bits	Name	Function
[31:28]	CMD13	Sets one LED. Value = 13 (0xD)
[27:25]	Unused	Not defined
[24:20]	HON	Hours on. "0" is always on. Values 0 to 31 (0x1F)
[19:17]	Unused	Not defined
[16:12]	HOFF	Hours off. "0" is once off stay off. Value 0 to 31 (0x1F)
[11:5]	Unused	Not defined
[4:0]	HDLY	Hours delayed (off). Value 0 to 31 (0x1F)

Instruction 14 (0xE): Unused

Available for a new instruction.

31 28	27				0
CMD14	Unused	Unused	Unused	Unused	Unused

Bits	Name	Function
[31:28]	CMD14	Unused but available. Value = 14 (0xE)
[27:0]	Unused	Not defined

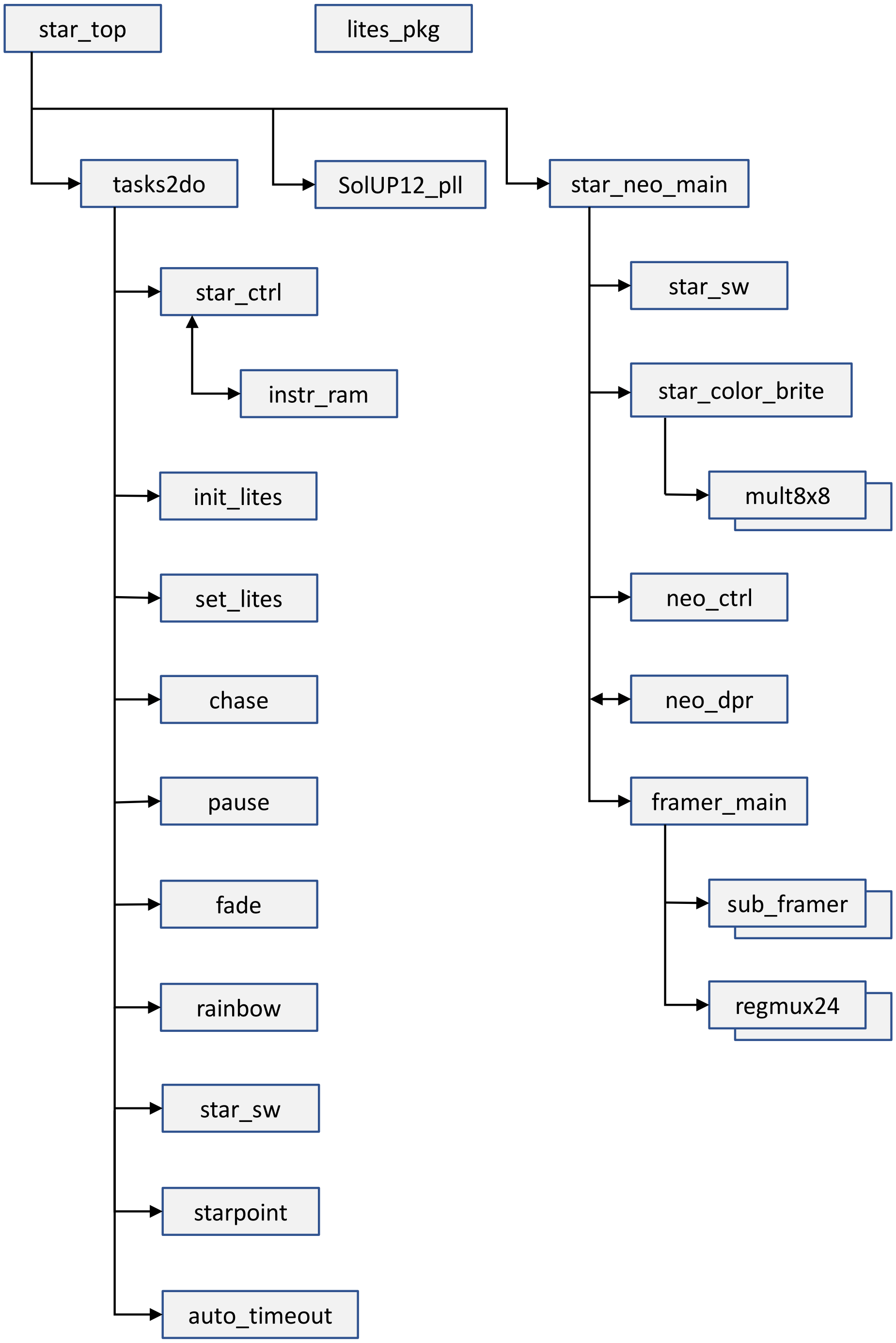
Instruction 15 (0xF): Unused

Available for a new instruction.

31 28	27				0
CMD15	Unused	Unused	Unused	Unused	Unused

Bits	Name	Function
[31:28]	CMD15	Unused but available. Value = 15 (0xF)
[27:0]	Unused	Not defined

Block Diagram of Starlite Code


star_top
lites_pkg
tasks2do
SolUP12_pll
star_neo_main
star_ctrl
instr_ram
init_lites
set_lites
chase
pause
fade
rainbow
star_sw
starpoint
auto_timeout
star_sw
star_color_brite
mult8x8
neo_ctrl
neo_dpr
framer_main
sub_framer
regmux24