Digital Security Lock System

Using VGA to display coordinated pixels

By Chris Barr & Jett Guerrero

Introduction

This project is a security lock that requires a password code to unlock a system. The locking system has a digital lock display to indicate the current condition, locked or unlocked. When the system locks or unlocks, a physical actuator opens or closes. The password code is resettable if one decides to change it.
This project will involve a 7-Segement Display, 5 onboard Switches and a single pushbutton, and VGA output. This will be a comprehensive guide that will help lead any beginner level programmer to completing this project.

Note: This guide will be using Windows 10 to guide the user.

Software/Hardware Used

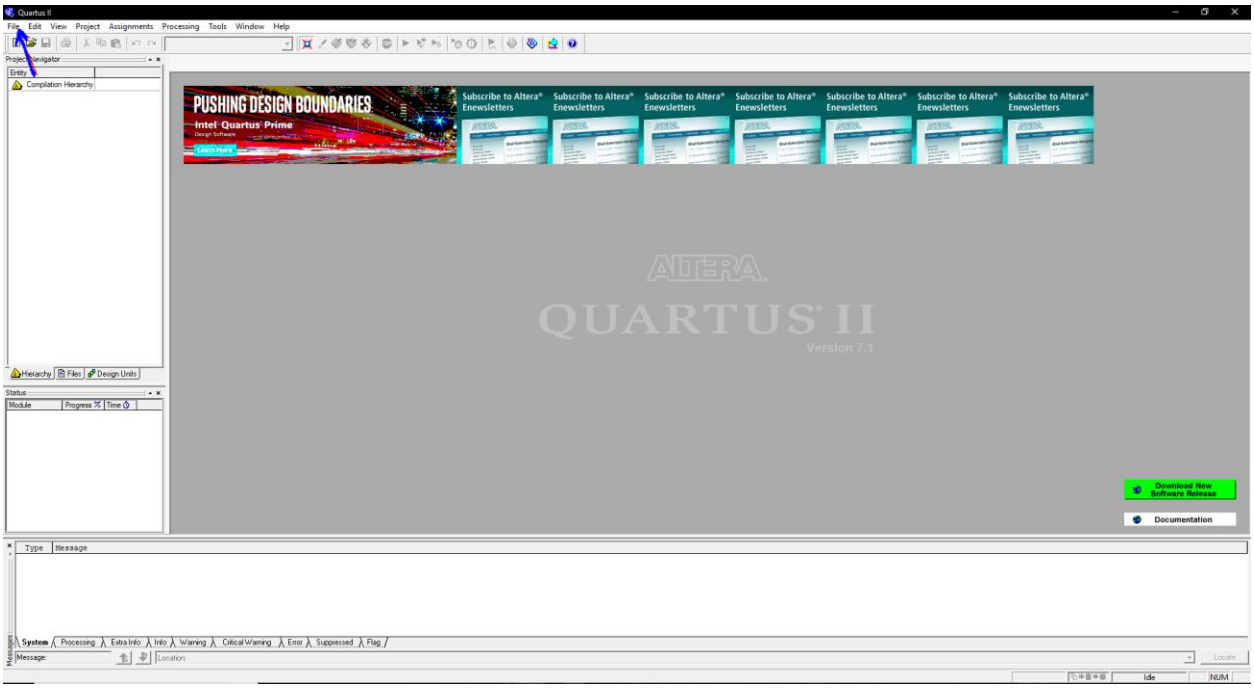
* Quartus II 7.1
* Cyclone II (EP2C35F672C6) DE2 Board
* USB Blaster (x64)

Display Info

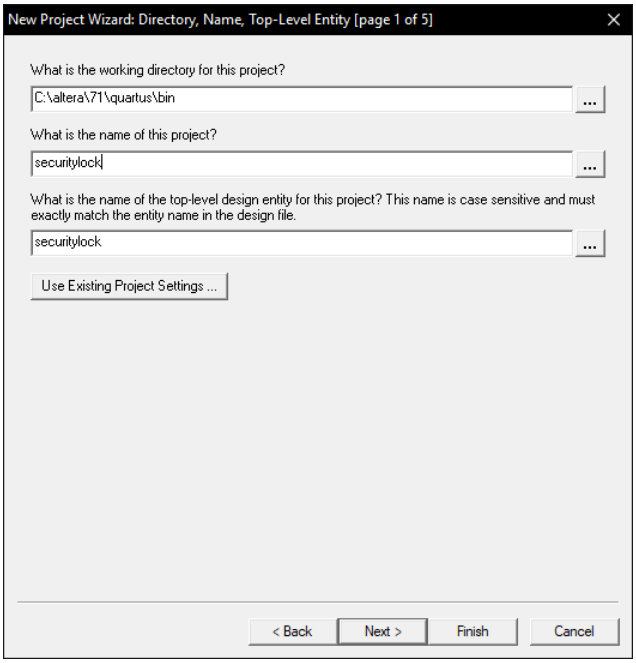
The password is to be entered using four switches on the DE2 board. The password values (0 or 1) is displayed on the seven-segment display. After the password code is entered, the right most push button (KEY0) must be pressed to unlock the system. If password is valid, the digital lock opens and turns green on the monitor. If invalid, the lock remains red and locked. If one wishes to change the password, one must first enter the valid password then switch on the left most switch (SW17) to enter a new password. In this mode, digital lock turns yellow.

Section 1: Quartus 7.1

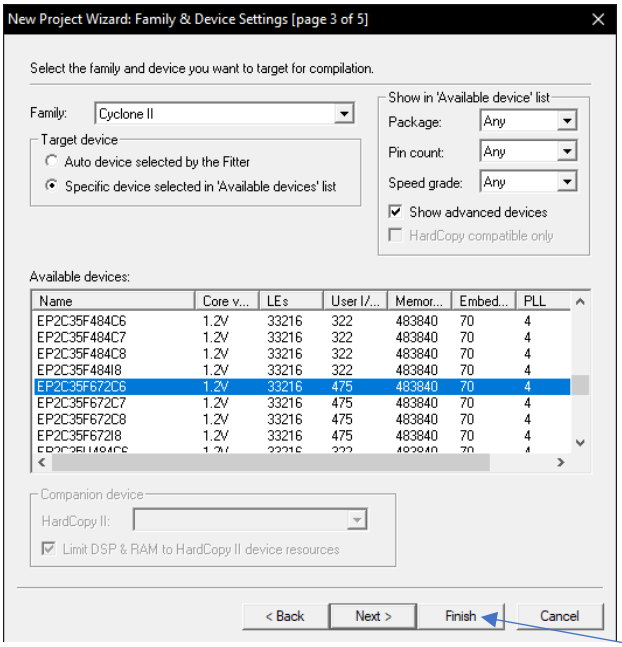
1. When Quartus 7.1 is opened, you will get this pop-up. Click on the **File** tab on the top-left of the screen, then hit **New Project Wizard…**.



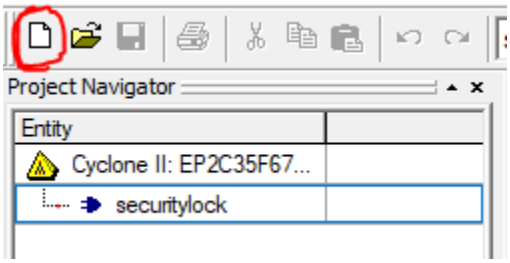
2. A smaller window will pop-up and tell you about the Project Wizard. Go ahead and click **Next**. Afterwards, you'll be asked to find a directory for your project as well as a name. Click **Next** when completed as shown below.



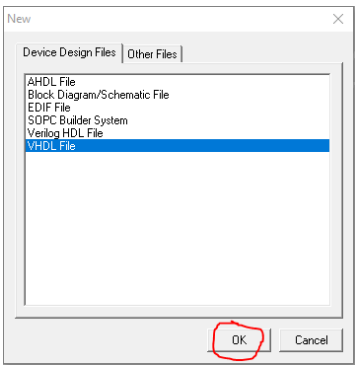
3. Page 2 of 5 will ask you to select any additional files you want to include into your project, we're going to skip that and click **Next**. Here's the important part, make sure you set your Family to **Cyclone II** and your Available Devices to **EP2C35F672C6**. Otherwise, check your boards manual to see which version you have of the board and select that under Available Devices.



4. Nothing more needs to be completed, so go ahead and hit that **Finish** button on the bottom right of that small window. Afterwards, you should see the name of your project listed under the Entity column as shown below. Now, **left-click the icon** shown from the picture circled in red.

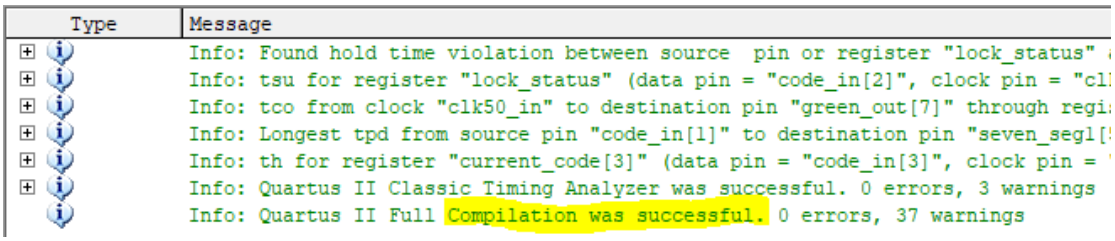


5. Highlight **VHDL File** by left-clicking it and then clicking **OK** at the bottom of that window.



6. You'll be introduced to a blank notepad that can be filled with VHDL code. Go ahead and **paste the code** at the very bottom of this document.

7. Compile the code by clicking the icon ►, or through the tabs Processing > Start Compilation, or CTRL+L. Make sure the code was compiled successfully as shown below.



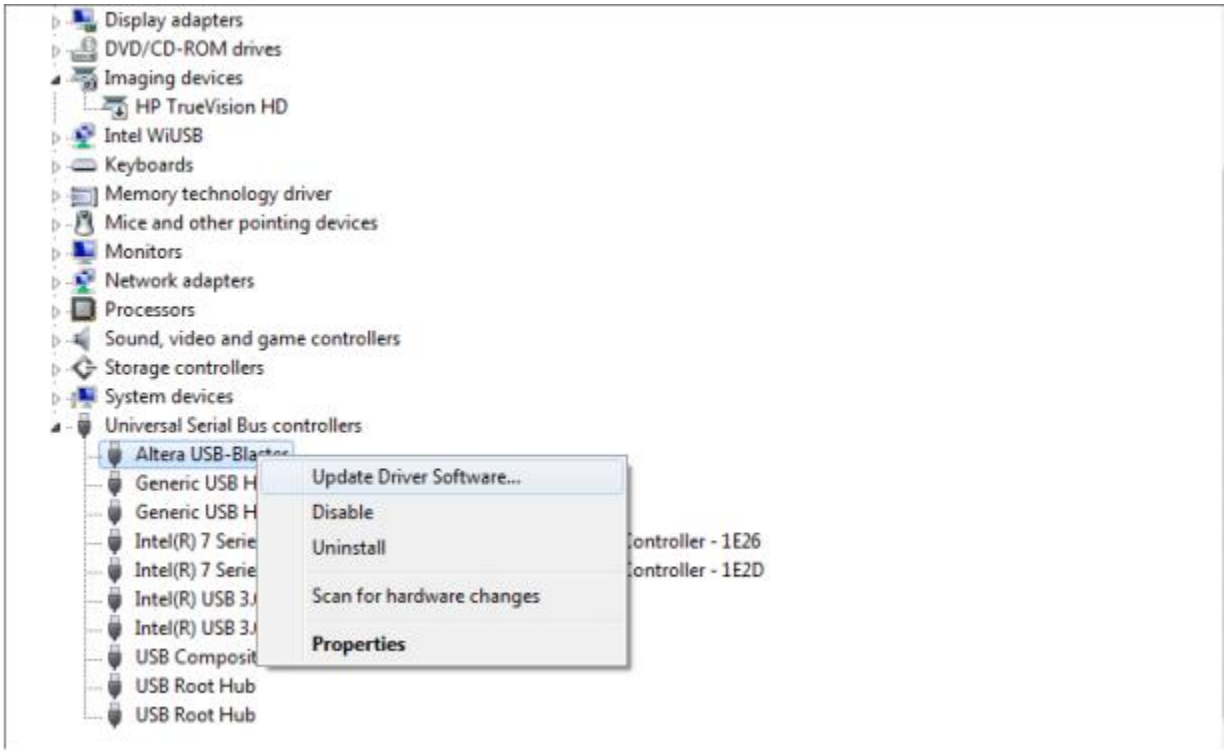
8. Now, we will need to pin-map the inputs, outputs, and signals to the components on the DE2 board. To do that, go to **Assignments > Assignment Editor**, or CTRL + SHIFT + A. In the image below, set all spots to these specifications:

	From	To	Assignment Name	Value	Enabled
1		blue_out[0]	Location	PIN_J13	Yes
2		blue_out[1]	Location	PIN_J14	Yes
3		blue_out[2]	Location	PIN_F12	Yes
4		blue_out[3]	Location	PIN_G12	Yes
5		blue_out[4]	Location	PIN_J10	Yes
6		blue_out[5]	Location	PIN_J11	Yes
7		blue_out[6]	Location	PIN_C11	Yes
8		blue_out[7]	Location	PIN_B11	Yes
9		blue_out[8]	Location	PIN_C12	Yes
10		blue_out[9]	Location	PIN_B12	Yes
11		clk25_out	Location	PIN_B8	Yes
12		clk50_in	Location	PIN_N2	Yes
13		green_out[0]	Location	PIN_B9	Yes
14		green_out[1]	Location	PIN_A9	Yes
15		green_out[2]	Location	PIN_C10	Yes
16		green_out[3]	Location	PIN_D10	Yes
17		green_out[4]	Location	PIN_B10	Yes
18		green_out[5]	Location	PIN_A10	Yes
19		green_out[6]	Location	PIN_G11	Yes
20		green_out[7]	Location	PIN_D11	Yes
21		green_out[8]	Location	PIN_E12	Yes
22		green_out[9]	Location	PIN_D12	Yes
23		hs_out	Location	PIN_A7	Yes
24		red_out[0]	Location	PIN_C8	Yes
25		red_out[1]	Location	PIN_F10	Yes
26		red_out[2]	Location	PIN_G10	Yes
27		red_out[3]	Location	PIN_D9	Yes
28		red_out[4]	Location	PIN_C9	Yes
29		red_out[5]	Location	PIN_A8	Yes
30		red_out[6]	Location	PIN_H11	Yes

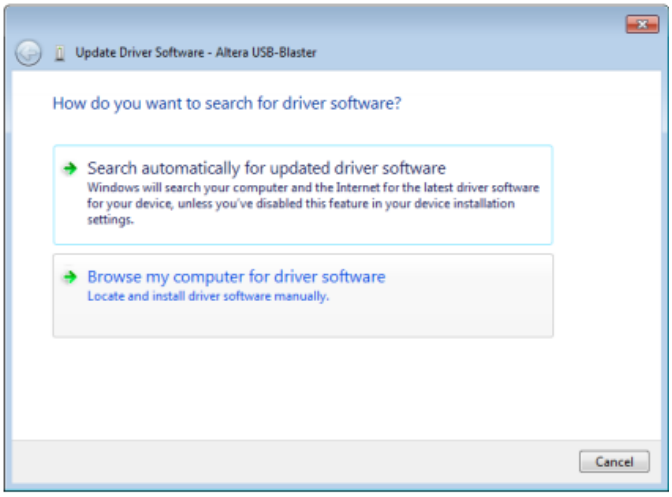
	From	To	Assignment Name	Value	Enabled
30		red_out[6]	Location	PIN_H11	Yes
31		red_out[7]	Location	PIN_H12	Yes
32		red_out[8]	Location	PIN_F11	Yes
33		red_out[9]	Location	PIN_E10	Yes
34		sync	Location	PIN_B7	Yes
35		vs_out	Location	PIN_D8	Yes
36		blank	Location	PIN_D6	Yes
37		code_in[2]	Location	PIN_P25	Yes
38		code_in[1]	Location	PIN_N26	Yes
39		code_in[0]	Location	PIN_N25	Yes
40		|	Partition Hierarchy	no_file_for_top_parti...	Yes
41		code_in[3]	Location	PIN_AE14	Yes
42		reset_code	Location	PIN_V2	Yes
43		seven_seg0[0]	Location	PIN_AF10	Yes
44		seven_seg0[1]	Location	PIN_AB12	Yes
45		seven_seg0[2]	Location	PIN_AC12	Yes
46		seven_seg0[3]	Location	PIN_AD11	Yes
47		seven_seg0[4]	Location	PIN_AE11	Yes
48		seven_seg0[5]	Location	PIN_V14	Yes
49		seven_seg0[6]	Location	PIN_V13	Yes
50		seven_seg1[0]	Location	PIN_V20	Yes
51		seven_seg1[1]	Location	PIN_V21	Yes
52		seven_seg1[2]	Location	PIN_W21	Yes
53		seven_seg1[3]	Location	PIN_Y22	Yes
54		seven_seg1[4]	Location	PIN_AA24	Yes
55		seven_seg1[5]	Location	PIN_AA23	Yes
56		seven_seg1[6]	Location	PIN_AB24	Yes
57		seven_seg2[0]	Location	PIN_AB23	Yes
58		seven_seg2[1]	Location	PIN_V22	Yes
59		seven_seg2[2]	Location	PIN_AC25	Yes
60		seven_seg2[3]	Location	PIN_AC26	Yes
61		seven_seg2[4]	Location	PIN_AB26	Yes
62		seven_seg2[5]	Location	PIN_AB25	Yes
63		seven_seg2[6]	Location	PIN_Y24	Yes
64		seven_seg3[0]	Location	PIN_Y23	Yes
65		seven_seg3[1]	Location	PIN_AA25	Yes
66		seven_seg3[2]	Location	PIN_AA26	Yes
67		seven_seg3[3]	Location	PIN_Y26	Yes
68		seven_seg3[4]	Location	PIN_Y25	Yes
69		seven_seg3[5]	Location	PIN_U22	Yes
70		seven_seg3[6]	Location	PIN_W24	Yes
71		enter_key	Location	PIN_G26	Yes
72		p_out1	Location	PIN_K25	Yes
73		p_out2	Location	PIN_K26	Yes
74	<<new>>	<<new>>	<<new>>		

9. Once you've set these pins. Go ahead and recompile.

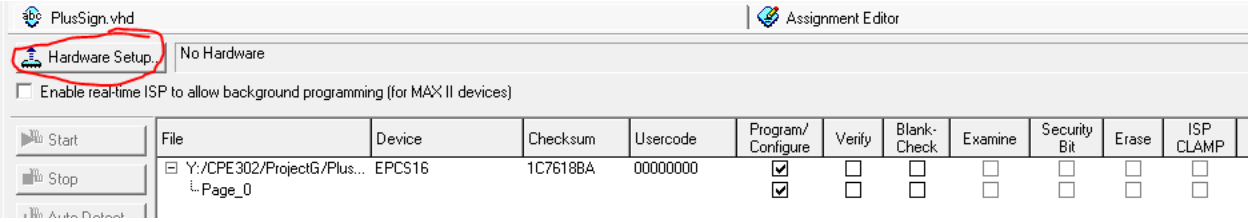
10. Great, now we have to program the board with our newly created code and pin-map locations. Before we do that, we must make sure our board is updated and in the system. <u>If the board is already updated onto the computer, skip this step and go to step 12.</u>
To update our board, we need to go to **Device Manager**. We can do that by hitting the window key and searching for Device Manager. Once open, you should get a window like this.



11. As shown, make sure you click **Update Driver Software…**. Then a window will pop up and ask you to search automatically or to browse your computer. Click **Browse my computer for driver software**. From there, you can navigate to your driver update by going from **Local Disk (C:) > altera > 7.1 > quartus > drivers > usb-blaster > x64**.

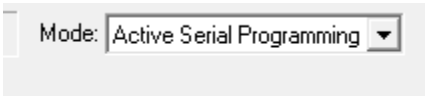


12. Going back to Quartus II 7.1, go to **Tools > Programmer**. Go ahead and click the **Hardware Setup….**



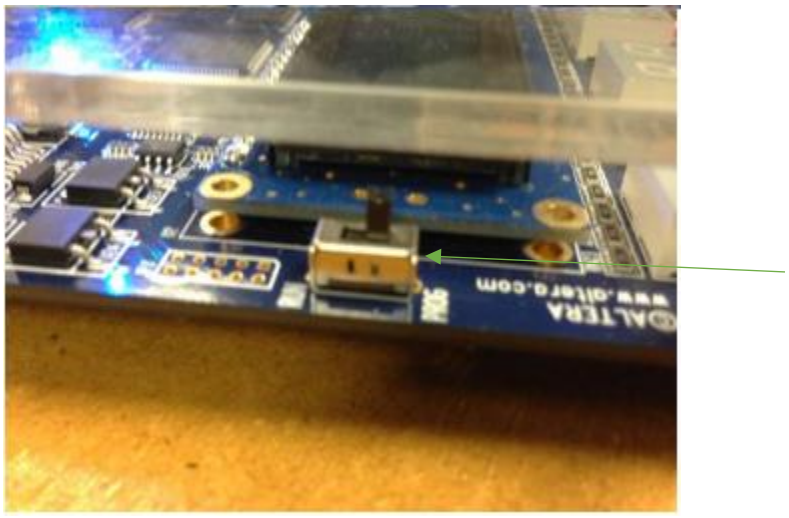
13. Hit the drop-down menu **Currently selected hardware:** and select the **USB-Blaster**.

14. Afterwards, you want to make sure that your mode is in **Active Serial Programming**.



15. Make sure your **Program/Configure** has the two boxes check-marked underneath their column:

16. Now, set your board to **Program** and hit **Start**



17. Let the board program for roughly 30 seconds till the blue LED that says “load” disappears or till you get a message back on Quartus saying the programming was a success.

18. Switch the board to **Run** mode instead of Program and restart the board by turning it off and back on. If the VGA cord is plugged into the monitor, you should see the image projected onto the monitor screen like this:

The picture of the left indicates that the lock is open, while the picture on the right indicates that the lock is closed.

19. To use this project, you must familiarize yourself with the button layout: