

Introduction to Project Hardware

Introduction

Each project team will start to acquire some project hardware this week. Some of it will be tailored to the anticipated needs of the team, and some initial code will be provided. Ideally, as many team members as possible will meet with Dr. Collins and/or Kevin Johnson when this is handed off, and hopefully this can occur on Monday, October 5.

Each team will get a DE2-115 board (Figure 1), a DE2-115 daughterboard (Figure 2), and some teams will get an I/O breakout board (Figure 3). Some teams will get an I/O breakout board later, as needed. These figures are all about 2/3 actual size.

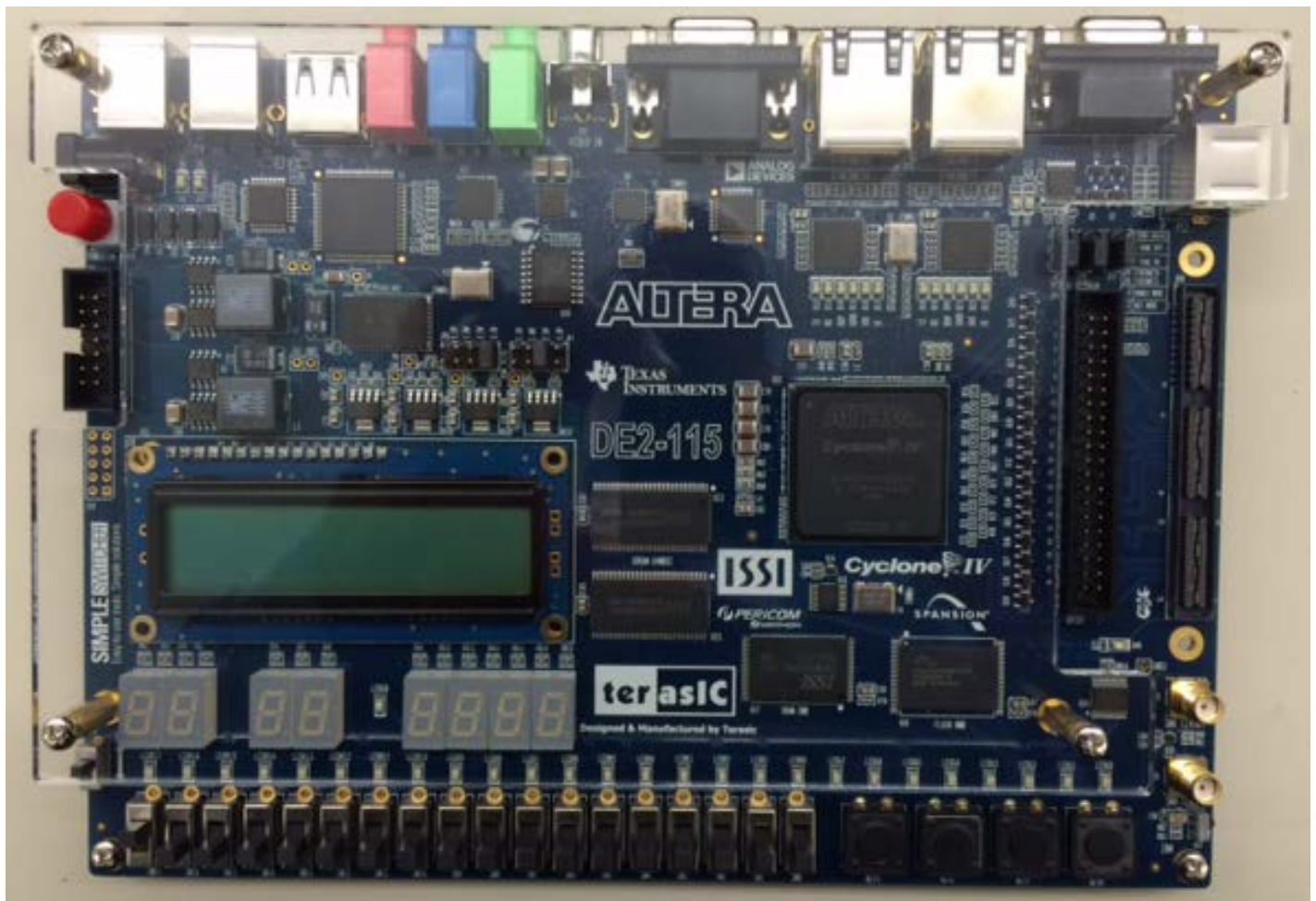


Figure 1. . DE2-115 development board with Cyclone IV FPGA.



Figure 2. The DE2 daughterboard connects directly to the DE2-115, providing 32 high-current outputs and an interface to the I/O breakout board.



Figure 3. The I/O breakout board provides connectors that are more suitable for audio transducers, LED arrays, motor drivers, and some other devices.

Another useful piece of hardware for some groups is a DIN terminal block adapter, something that brings the 32 high-current output signals to industrial-style screw terminals, as shown in Figure 4.



Figure 4. Attachment of the DIN terminal block adapter.

When you get your DE2-115 board, make sure that the jumpers on JP6 and JP7 are in the rightmost position, as shown in Figure 5. The default position of JP7 is the third column, as shown in the expanded detail, but the correct position for both JP6 and JP7 is the fourth (rightmost) column, as shown in the greyed image of the entire board. If not, some of the daughterboard features will not work reliably. In particular, the default position for JP7 is the incorrect one shown in the expanded detail, and it is possible that Dr. Collins will forget to fix some of them!

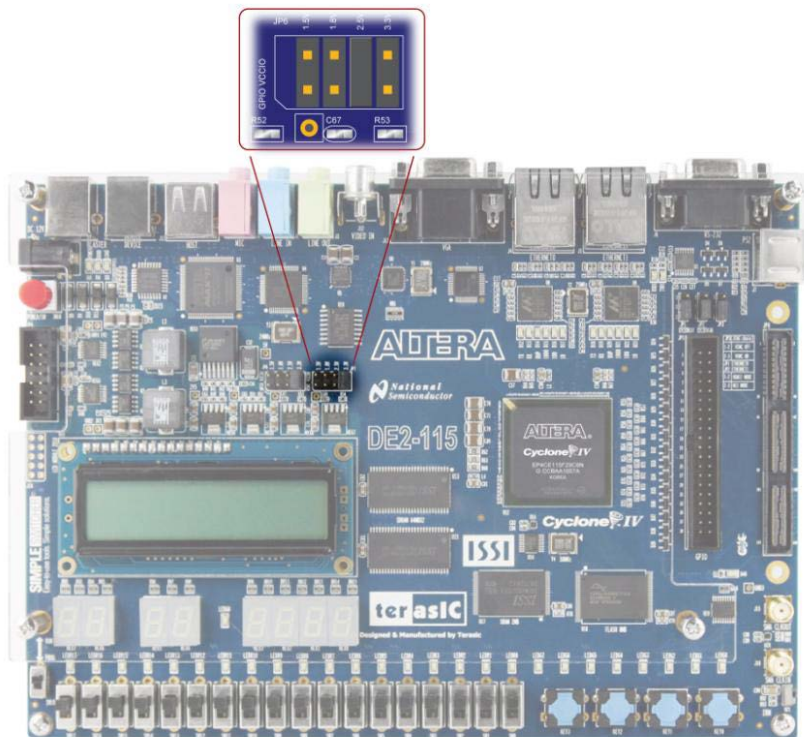


Figure 5. JP6 and JP7 are the two groups of pins just above and to the left of center on the DE2-115 board. (Image from Altera DE2-115 User Manual)

It is possible to misalign the daughterboard and the I/O breakout board when they are connected to each other and to the DE2-115 board. Make sure that you get some assistance from Dr. Collins or Kevin Johnson when first connecting them.

Using the DE2-115 board with Quartus

Generally, the DE2-115 board works the same as a target development board as any of the lab DE2 boards did. Because it is a newer board, it is recommended that you use Quartus version 13.0, Service Pack 1, which the LEs installed on their own computers as part of Lab 4.

LEs can utilize their previous experience (and a copy of Lab 4) to walk the non-LEs through the process of installing this newer version of Quartus on their own computers.

Whenever a project is compiled for the DE2-115, you must select the “Assignments => Device...” menu option, and in the dialog that follows, choose the Cyclone IV E family, and the EP4CE115F29C8 device name, as shown in Figure 6. This corresponds to a similar step in the tutorial, where a different (Cyclone II) device was used.

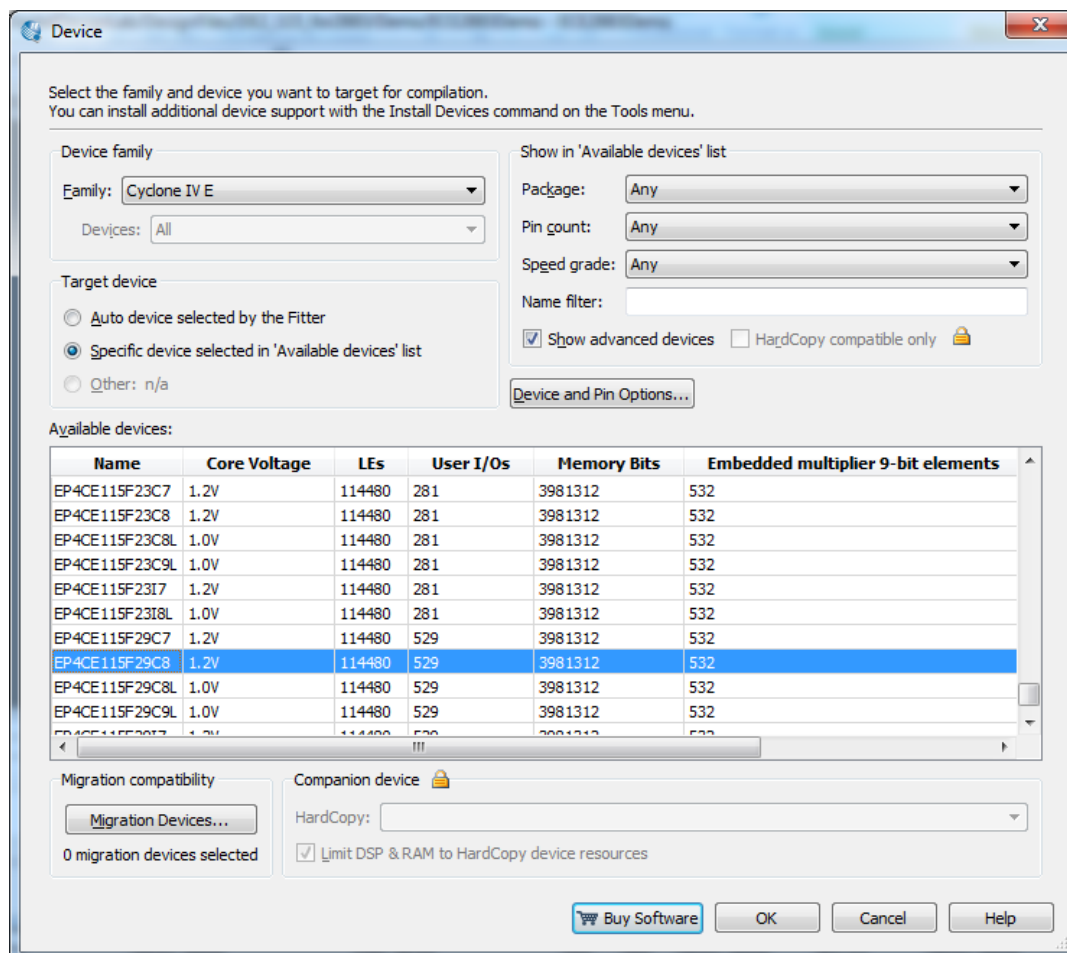


Figure 6. Device settings for the FPGA on the DE2-115 board.

There are some minor user interface differences between Quartus 13.0 and the earlier version in the lab, but you probably will not have any major difficulties unless you try to simulate something. At this time, it is probably best to avoid simulation.

For your information, the settings for these jumpers make the I/O pins operate at a “TTL-compatible” voltage level. Recall that we described logic 0 as “near 0 V” and logic 1 as “near 5 V.” The pins on the FPGA can go no higher than 3.3 V, but that is “close enough to 5V,” and these jumper settings are required for that 3.3 V level.

Pin assignments on the DE2-115

In the earlier tutorial, it was important to assign the correct pins when using them for direct connection to features such as slide switches or LEDs. The same is true for the DE2-115, but the pin assignments for switches and LEDs are different, and there are many other pins to consider, including over 100 pins that are connected to the daughterboard and the I/O breakout board. You will be provided with a spreadsheet listing all of these functions, as well as a “QSF” file, which can be associated with any project you create, automatically providing assignments for pins, as long as you use the exact names in the spreadsheet. Some DE2-115-related pins are included in the QSF file that are not in the spreadsheet, but they are ones that you are less likely to ever need. An excerpt from the spreadsheet is shown in Table 1. Since every pin in the spreadsheet has either a primary function on the DE2-115 board **or** on one of the attached boards, two columns are provided to make it clearer where the function resides.

Table 1: Excerpt from spreadsheet of pins. If the pin name in the third column is used for daughterboard pins, or the name in the second column is used for DE2-115-only pins, automatic pin assignment is possible with the use of a QSF file.

FPGA pin	Usage on DE2-115	Usage on attached boards	Description
PIN_AB22	Passed through	AUDIO1_A_L	Audio transducer
(pins omitted)
PIN_D28	Passed through	HCIO[6]	High current output
PIN_F24	Passed through	HCIO[7]	High current output
PIN_K21	Passed through	HCIO[8]	High current output
PIN_L24	Passed through	HCIO[9]	High current output
PIN_P27	Passed through	KEYPAD1_INT	Keypad interface
PIN_J24	Passed through	KEYPAD1_SCL	Keypad interface
PIN_J23	Passed through	KEYPAD1_SDA	Keypad interface
PIN_P28	Passed through	KEYPAD1_SPARE	Keypad interface
PIN_Y2	CLOCK_50	Does not go to daughterboards	50 MHz clock input
PIN_AG14	CLOCK2_50	Does not go to daughterboards	50 MHz clock input
PIN_AG15	CLOCK3_50	Does not go to daughterboards	50 MHz clock input
(many more pins)

Familiarization exercise with DE2-115

The familiar schematic project from the Quartus tutorial should work fine on the DE2-115, as long as pins are assigned correctly. Both LEs and non-LEs have gone through that tutorial.

Copy your project directory from the tutorial to a location on your computer that has Quartus 13.0 installed. You do not have to copy any subdirectories, like “db” or “incremental_db,” as these will be recreated when recompiled. Open the project in Quartus 13.0, and you will get a warning like that of Figure 7. This is normal so select “Yes.” From experiments made thus far, it appears to be possible to move projects BACK to Quartus 9.1 with no ill effects, but you will continue to get warnings when moving them forward again.

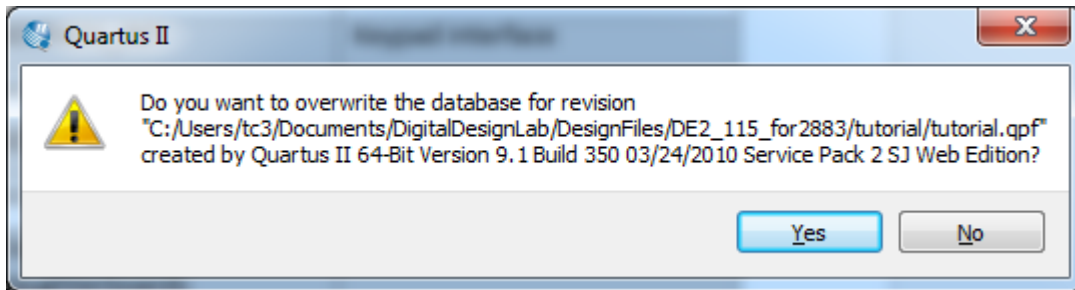


Figure 7. Warning when opening previously created project.

- Compile the unmodified project with your own Quartus installation just to verify that everything is alright.
- Download two files to your computer. You can put them in your project directory or a more general location for ECE2883, since you will use them for other projects:
 - http://powersof2.com/resources/DE2_115_for2883.qsf (a file of pin assignments that Quartus can interpret)
 - http://powersof2.com/resources/DE2_115_Pins.xlsx (a spreadsheet file of the same information that you can view easily)
- Then make the device assignment shown earlier in Figure 6, noting that Quartus will recommend removing all pin assignments. Agree to this, since those pin assignments for a different FPGA are useless.
- Do something new. Select “Assignments => Import Pin Assignments...” from the menu. When you are requested to provide a file name, browse to the location where you saved DE2_115_for2883.qsf, select it, and all of those assignments will now be part of your project.
- Rename the pins to the names used in the spreadsheet DE2_115_Pins.xlsx. For example, even with the information just provided, Quartus will not recognize “d3” as the third red LED unless you rename it with the spreadsheet name of LEDR[3]. Or pick a green LED if you want! Or use different switches (SW[0] through SW[17] or KEY[0] through KEY[3]).
- If the pin assignments do not show up next to the pin symbols on the schematic, make sure that “View => Pin Assignments” is checked. Also, if it WAS checked, you may need to uncheck it, then check it again, to make it refresh the view. If some assignments appear, while others do not, you probably mistyped some of the names, so check the exact spelling on the spreadsheet.
- Recompile the project.

- Place your new DE2-115 board on a flat surface, always being careful not to allow anything metal to electrically short pins on the back of the board. Plug it into its power supply, connect the USB cable between your computer and the Blaster port of the DE2-115 (not the other USB port), and you should be able to program the FPGA with steps similar to the earlier tutorial, and the project should operate correctly.

Team-specific exercises

Each team will have some specific activities designed to expose them to some relevant hardware and a prototype design file that may be useful as a starting point for their project. These instructions will be a little sketchy, but hopefully you will be working with Dr. Collins or Kevin nearby. Soon, you will be more self-sufficient and able to continue development outside the lab. In the interests of open development and fairness, every team can see the starting point of the other teams. You are also welcome to share these and future design files between teams, as long as there is mutual contribution and a general understanding of everything that finds its way into your completed project.

Laser Spirograph/Laser Show

Connecting lasers and controlling on/off or variable modulation is something to consider soon. You also need to look at simple servo motor control and H-bridge control of a DC motor to pin down what is useful there.

Right away, you can look at putting mirrors on an audio transducer to see how interesting that is.

Needs: DE2-115, Daughterboard, I/O breakout board, one audio transducer, small mirror, audio-amplifier ribbon cable, dual-channel audio amplifier

Steps in lab:

- With assistance from Dr. Collins or Kevin, connect the daughterboard to the DE2 board
- There is a small 3-pin header called JP4 near the top left corner of the daughterboard. Make sure that it has a jumper connecting the lower two pins. (This provides 5 V power to the daughterboard, and eventually to the audio amplifier, from the DE2-115 board. Later, if more audio transducers are used, you may need to provide external 5 V power and change that jumper.)
- Still with assistance, connect the I/O breakout board to the daughterboard.
- Using a provided 20 pin ribbon cable, connect one end to the AUDIO1 plug on the breakout board, and the other end to the audio amplifier.
- Making sure that nothing is shorted, turn on the DE2-115 board
- Download the AudioTransducer.zip file from the Project page of the web site.
- Unzip it to a folder on your computer
- Compile it and download it to your DE2-115 board
- What this does:
 - A 100 kHz square wave clock is generated by the altp110 device in the schematic.
 - That clock drives a 12 bit counter, and when it reaches 4095 after about 40 ms, it rolls over, essentially starting again.
 - Any bit of the count is a square wave, with the lowest bit being about 50 kHz and the highest being about 24 Hz
 - Switches SW3 to SW0, treated as a binary number, select a count bit between 0 (highest frequency) and 11 (lowest frequency).

- The AND gate acts as an output enable. You must press KEY3 to send output to the audio amplifier
 - FYI, there are 4 red LEDs showing the binary switch settings.
- Try to fully understand why this works the way it does

Remote-Controlled Multisensory Painting

The wireless controller is a key component, and if possible, we should work together on the communication interface (serial, I2C, etc.) from the DE2 board to the controller.

More immediately, it would be useful to begin experimenting with the audio transducer.

Needs: DE2-115, Daughterboard, I/O breakout board, one audio transducer, small mirror, audio-amplifier ribbon cable, dual-channel audio amplifier

Steps in lab:

Same as Laser Spirograph team above.

Collaborative Music Device

You will probably need some more guidance as you consider the best way to synchronize audio files, but the near-term exercise would be to play with some solenoid connected to the high-current outputs and tell us more about the LED hardware you envision.

Needs: DE2-115, Daughterboard, two different solenoids, DIN terminal block adapter, external power supply.

Steps in lab:

- With assistance from Dr. Collins or Kevin, connect the daughterboard to the DE2 board
- There is a 2-pin connector called TB1 near the lower right corner of the daughterboard. Connect the external power supply to this connector. (It provides the current needed for the solenoid.)
- Connect the DIN terminal block adapter to the edge connector on the right side of the daughterboard, labelled JP3.
- Use the screw terminals to connect a solenoid. One wire should be connected to screw terminal 1, and the other should be connected to screw terminal 5.
- Making sure that nothing is shorted, turn on the DE2-115 board
- Download the Solenoid.zip file from the Project page of the web site.
- Unzip it to a folder on your computer
- Compile it and download it to your DE2-115 board
- What this does:
 - A 10 kHz square wave clock is generated by the altpll0 device in the schematic.
 - That clock drives a 16 bit counter, and when it reaches 65532 after about 6.4 s, it rolls over, essentially starting again. Only the upper 8 bits (slowest counting) are used.
 - Any bit of these upper 8 bits of the count is a square wave, with the lowest bit being about 19.5 Hz and the highest being about 0.15 Hz
 - Switches SW2 to SW0, treated as a binary number, select a count bit between 0 (highest frequency) and 7 (lowest frequency).
 - The selected frequency is sent to the high-current output driving the solenoid.
 - FYI, there are 3 red LEDs showing the binary switch settings.
- Try to fully understand why this works the way it does

Mimicry of Dance through Solenoid Application in Tap Shoes

Kevin has a prototype design for the sequence capture that you can play with, and you can experiment with some solenoids connected to the high-current outputs.

Needs: DE2-115, Daughterboard, two different solenoids, DIN terminal block adapter.

Steps in lab:

- Download the SimpleSequencer.zip file from the Project page of the web site.
- Unzip it to a folder on your computer
- Compile it and download it to your DE2-115 board
- What this does:
 - About 250 times a second, the address of the 1024-word memory increments
 - About every four seconds, it rolls over, essentially starting again
 - As long as you have SW0 up, it writes the settings of the four blue keys to memory, essentially saving a 4-second snapshot
 - When you lower SW0, it plays it back on four green LEDs
 - FYI, there are four red LEDs showing the upper bits of the count sequence.
- Try to fully understand why this works the way it does

Moving Mirrors

The method of sensing people is something that we need to work on.

Meanwhile, getting familiar with a servo motor controller device will be useful, so we will start with a variation on the one used in the non-LE lab a few weeks ago.

Needs: DE2-115, Daughterboard, servo motors, ribbon cable adapter.

Steps in lab:

- Download the RCServo.zip file from the Project page of the web site.
- Unzip it to a folder on your computer
- Compile it and download it to your DE2-115 board (you do not need to have the daughterboard connected)
- With help from Dr. Collins or Kevin,
 - Connect the white or yellow wire from an RC Servo to Pin 10 of header JP1
 - Connect the red wire from the servo to 5 V
 - Connect the black wire to ground
- What this does:
 - The settings of SW9 through SW0 are considered to be a binary number
 - The constant 50 is added to it ensuring that a number of at least 50 is produced
 - A 12.5 MHz clock is generated from a 50 MHz clock by the altpII0 device
 - An 18-bit counter is clocked by the 12.5 MHz clock, rolling over about every 21 msec
 - The upper ten bits of that count, SERVO_TIME[17..8], is a ten-bit count, increasing from 0 to 1023 in that 21 msec repeating cycle
 - About every 21 msec, a new pulse starts on JP1-10 (goes high), because the count resets to zero, and zero is less than whatever number is coming from the adder
 - It remains high until the count exceeds the number coming from the adder

- Disconnect the servo from JP1-10 and look at the signal on the oscilloscope as the value of the switches is changed.
- Look at the pulse signal expected by an RC Servo (see references provided for the non-LE version of Lab 2). Why is the constant of 50 used? What bug happens when the switches are set for a large number?