

Experiment 0

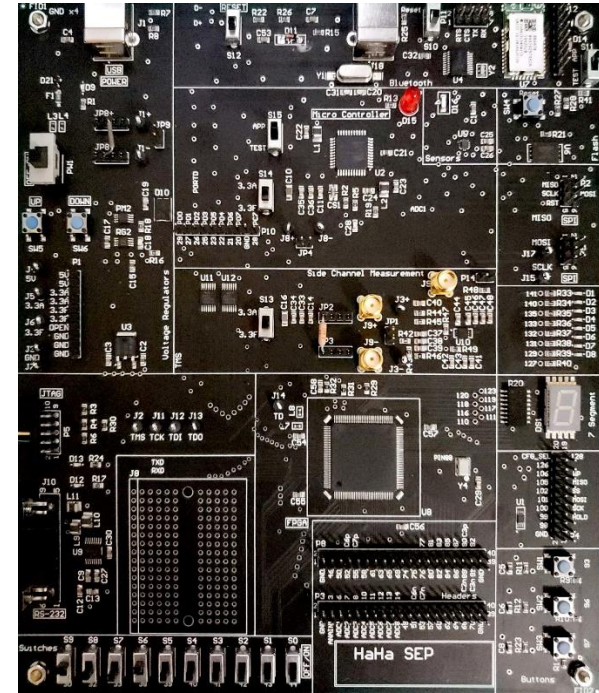
Self-Test on HaHa Board

In this experiment, you will go through a complete test of the Hardware Hacking (HaHa) Board.

Experiment Set-up: Configuration

The hardware and software needed for this experiment include:

1. The HaHa Board
2. A USB Blaster
3. A USB A-to-B cable
4. A Windows machine/computer with ≥ 2 USB ports.
 - a. Quartus (≥ 15.0) to program the FPGA w/ MAX 10 FPGA device support [1]
 - b. Atmel Studio [2] to program the microcontroller
 - c. *Optional* – Quartus 13.1 to install USB-Blaster [1]
5. An oscilloscope or Analog Discovery Board
 - a. *Optional* – Waveforms 2015 to view AD board signals [3]



Instructions and Questions

To perform the following self-test procedures, you must download the Self-Test code binaries for the HaHa board. Also, you must install the Quartus (≥ 15.0) [1] and Atmel Studio programs into your Windows machine. In order to use the Altera USB-Blaster, you must install its drivers, which can be achieved by either 1) downloading the USB-Blaster folder if it is available online, or 2) grabbing this folder inside a Quartus 13.1 installation. After, the USB-Blaster must be updated using the “Device Manager” utility. Instructions for installing the required and optional software will be included at the end of this document.

NOTE: If you have any questions for parts II-V, please refer to the FAQs at the end of this document or the HaHa Board Manual.

PART I: Test the Power Supply and Voltage Regulators

1. Connect the board to the USB Power supply (J1) with a USB A-to-B cable to your machine. Turn on the power to see if the power light is on. If not, check the voltage across the fuse (F1) using an oscilloscope or Analog Discovery board. If it is not 0V or close to it, the fuse should be replaced.
 - a. Take a picture of the board while it is on with LED D9 lighting up.
2. Using an oscilloscope or Analog Discovery board, check the voltage of the input (left pin) and output (right pin) of the voltage regulator (U3) against ground. If the input is 5V but the output is not 3.3V, the voltage regulator should be replaced.
 - a. Take a picture showing the voltage of the input pin (5V).
 - b. Take a picture showing the voltage of the output pin (3.3V).
3. Check the voltage on the test point 3.3A. Use button SW5 and SW6 to change the voltage.
 - a. Answer this question: what is the adjustable voltage range?
 - b. Take a picture showing the minimum voltage of the 3.3A test point.
 - c. Take a picture showing the maximum voltage of the 3.3A test point.

NOTE: After doing voltage measurement, remember to set both S13 and S14 to “3.3F” to make sure you can program both chips successfully.

PART II: Test the FPGA and the Peripherals

1. Connect the 10-pin JTAG connector of the USB Blaster to the JTAG header (P5) of the board. Connect the USB-B cable of the USB Blaster to your machine.

Open Quartus and navigate to **Tools > Programmer** in the menu bar. In the *Programmer* window, make sure “USB-Blaster [USB-X]” (X is the number that Quartus assigned to your USB-Blaster) is shown next to a button called *Hardware Setup*. If not, click on this button, change the *Currently selected hardware* to “USB-Blaster [USB-X]”, and close this window. Click on the *Auto Detect* button to detect the two chips in the JTAG chain. In the *Select Device* window, keep the selected device and click OK. Do NOT close this window. You will need this for the next steps.

If this is not a success, there are two possible remedies: 1) If there is no current sensing sensor mounted in the JP2/JP3 sockets, please mount it; OR 2) reinstall your USB-Blaster driver

Note: The chip with the name starting with *UNKNOWN_* is the ATmega16U4 microcontroller. Whenever you create a new Quartus project and compile your HDL code for programming, this chip will not be included on the *Programmer* window at first. You must click on the *Auto Detect* button. In order to program correctly, Quartus must account for all the chips in the JTAG chain.

- a. **Deliverable:** Attach a screenshot of the Programmer Window showing Quartus detecting two chips in the JTAG chain
2. Click on the “10M50SA” chip in the JTAG chain picture. Then, click on *Change File...* in the left side of window. In the new window, navigate to where you downloaded the Self-Test code binaries and select the POF file (**test.pof**). Check the *Program/Configure* on the first row. This will check the two boxes underneath the first row.

Right click on the microcontroller chip (starts with *UNKNOWN_*) in the JTAG chain. Click on *Edit* and select *Change Device*. Click on the name of the chip (again starts with *UNKNOWN_*). Select *Edit...* option, change its name to *ATMEGA16U4*, and click OK.

You are now ready to program the board. Click on *Start* on the left side. Wait until the programming is complete.

If this is successful, the 8 LEDs of on the board should be blinking, and the 7-segment display a hexadecimal digit counting from 0 to F. To check the function of the 10 user switches, just switch any of them and you will see the 8 LEDs will change their pattern of blinking. There are two different patterns of blinking and whenever any of the switches is switched, the pattern should change. If not, replace the switch.

To check the function of the pushbuttons, just push any of them, and you should see the 7-segment display stops counting and start to repeat blinking with a fixed number. If you do not see this, the button should be replaced.

- a. **Answer these questions:** What are the two LED patterns occurring in the board? Which patterns of switches trigger either of the LED patterns?
- b. **Answer this question:** What happens to the digit on the 7-segment display when you keep pressing any of the key buttons?
- c. **Deliverable:** Attach a screenshot of the programmer window showing that you successfully programmed it.
- d. **Deliverable:** Take a picture of the board with one of the LED patterns and a hexadecimal digit displayed in 7-segment display.

NOTE: Before you proceed to the next parts, please complete Part II. Make sure switch S12 is switched down.

PART III: Test the Microcontroller and the Accelerometer

1. For HaHa SEP 1, ground the PE2 pin. For HaHa SEP 3, place switch S15 to *TEST*.
Reset the microcontroller by toggling switch S12 up once then immediately down.

Open Atmel Studio and navigate to **Tools > Device Programming** in the menu bar. Alternatively, you can press *Ctrl + Shift + P* to open the same window. Select *Atmel Mega DFU* for Tool and *ATmega16U4* for Device. Click Apply. On the left sidebar, click on *Memories*. In the Flash section, click on the “...” next to the path and navigate to the location of the provided file (U_ACC.elf). Click on *Program*.

2. For HaHa SEP 1, connect the PE2 pin to VDD. For HaHa SEP 3, place switch S15 to *APP*.
Reset the microcontroller by toggling switch S12 up once then immediately down.

The microcontroller will read the acceleration data from the accelerometer and send it to the FPGA. The FPGA will show the value in binary on the eight LEDs if you place the most significant four switches to “1001”. Please rotate the board and see if the value showing on the LEDs are changing. If not, the accelerometer is probably not working well.

- a. **Deliverable:** Attach a photo of the board showing the acceleration value on the LEDs in your report.

PART IV: Test the Temperature Sensor, RAM Module, and PLL Module

1. Finish Parts II and III before proceeding. Open Quartus and navigate to **Tools > In-System Memory Content Editor** in the menu bar. Set *Hardware* to “USB-Blaster [USB-X]” and click on *Scan Chain*. Select Index 0 with Instance ID of *RAM1*. Navigate to **Processing > Continuously Read Data from In-System Memory** in the menu bar. Alternatively, you can press *F6*. The status will change from *Ready to Acquire* to *Acquisition in Progress*. You should see 3 bytes of info:

[7-Segment Display Digit] [Accelerometer] [Temperature]

The In-System Memory Content Editor is reading values off a RAM module where each of the three pieces of information are written into. Also, the FPGA chip has an internal ADC Temperature Sensor Diode. It can detect the temperature of the chip, so you can apply heat or cold on the chip. Some ways to do this are using a hairdryer or an ice pack that is relatively dry. Do NOT place anything wet on the chip. The ADC module is sampled at 10MHz which is derived using a PLL that takes in the FPGA’s 50 MHz clock as the input.

NOTE: Before closing the In-System Memory Content Editor, you must stop the continuous reading of data. Go to Processing > Stop In-System Memory Analysis or press *Esc* key. Make sure the status returns to *Ready to Acquire*. If so, you can safely close the window.

- a. **Deliverable:** Attach a screenshot of the In-System Memory Content Editor window showing that 3 types of data.

PART V: Test the EEPROM

1. Finish Part II before proceeding. Refer to Part III for microcontroller programming steps. Program the microcontroller with the provided file (U_EEPROM.elf). The microcontroller will read a signature of the EEPROM and send it to the FPGA. Put switches S9 – S6 to “1001” to see the values on the LEDs. The LEDs should be showing 00101001 (0x29).
 - a. **Deliverable:** Attach a photo of the board showing the value 0x29 on the LEDs in your report.
 - b. **Deliverable:** Attach a screenshot of the In-System Memory Content Editor window, showing 0x29 for the 2nd byte of data.

PART VI: Feedback

Please note this part is *not* optional. **Answer these questions:**

1. What problems have you met when doing the tests? How did you solve them?
2. What suggestions do you have to make the test process easier and more friendly to students?

Lab Report Guidelines and Demonstration

1. In your report, answer ALL the questions.
2. Give screenshots and photos as required to your written report.
3. Record a video demonstration of your working HaHa board by showing all parts of the Self-Test.

References

- [1] <https://fpgasoftware.intel.com/?edition=lite>
- [2] <http://studio.download.atmel.com/7.0.2389/as-installer-7.0.2389-full.exe>
- [3] <https://mautic.digilentinc.com/asset/110:waveforms-windows-64-bit-download>
- [4] <http://www.microchip.com/wwwproducts/en/MCP2221>
- [5] <http://logmett.com/tera-term-the-latest-version>

Software Installation Instructions

PART I: Install Quartus

1. Go to link [1] and download the following files into the same folder for installation:
 - a. Quartus Prime (includes Nios II EDS)
 - b. MAX 10 FPGA device support
 - c. *Optional* - ModelSim-Intel FPGA Edition (includes Starter Edition)
2. Start installing Quartus. Click “Next” in the following picture.
3. Click on “I accept the agreement” and then Next.
4. This is the installation directory. You can change your directory here, especially if you have a dual drive configuration. By default, it will install in the C:\ drive. Then, click Next.
5. Click Next. Please note that ModelSim is optional.
6. Click Next.
7. Wait until the software finishes installing.

PART II: Install driver for the USB Blaster

1. Get the “usb-blaster.zip” file from downloading this folder or grabbing from Quartus 13.1 and unzip it.
2. Plug in the USB Blaster to your computer.
3. Open the Device Manager of your computer.
4. Find the “USB-Blaster(Altera)” in the listed devices.
5. Right-click this device and click “Update driver”.
6. Select “Browse my computer for driver software”.
7. Input the directory that contains your unzipped files from step 1 and click “Next”.
8. The USB-Blaster should now be installed on your computer.

PART III: Install Atmel Studio

1. Go to link [2]
2. Open the installer. If needed, select your destination folder. Otherwise, keep it the same. Click on “Agree to the terms” and click Next.
3. Select the first option and then Next. (The other 2 options are optional.)
4. Select the option and Next.
5. Click Next.
6. Click Install and wait until it is installed.

Optional: Install Waveforms (Only if you have AD2 board)

1. Go to [3]. It should direct to the download link for Waveforms 2015. If it does not, go to [4] and provide your information to get the download link.
2. Open the installer. Click Next.
3. Click Next.
4. Click Next.
5. If needed, select your destination folder. Otherwise, keep it the same. Click Install.
6. Wait until the software finishes installing.

References

- [1] <https://fpgasoftware.intel.com/18.1/?edition=lite&platform=windows>
- [2] <http://studio.download.atmel.com/7.0.2389/as-installer-7.0.2389-full.exe>
- [3] <https://mautic.digilentinc.com/asset/110:waveforms-windows-64-bit-download>
- [4] <https://mautic.digilentinc.com/waveforms-download>

Frequently Asked Questions

FAQ about the Microcontroller

1. Q: After I program the Microcontroller and turn off the board, will the program remain in the Microcontroller?
A: Yes.
2. Q: Why is my program for the Microcontroller not working as soon as I succeeded in programming it?
A: Your program will not be running until you reboot the board.
3. Q: I cannot program the Microcontroller. Why?
A: You could have made these mistakes:
 You use the wrong USB port. You must use J1, instead of J18.
 Or, you turned on S12 and reset the chip. You should turn it off.
 Or, you turned on S14 and power the Microcontroller with an adjustable voltage source. You should turn S14 to 3.3F.
4. Q: Why is the Microcontroller not running the code I programmed after I reboot the board?
A: To run the code, you should:
 - a. Connect the PE2 pin to VDD before you reboot for HAHA SEP 1.
 - b. Put switch S15 to *APP* before you reboot for HAHA SEP 3.
5. Q: Why I cannot re-program the Microcontroller? It seems it is always running the code I programmed last time.
A: To re-program the chip, you should:
 - a. Ground the PE2 pin before you reboot for HAHA SEP 1.
 - b. Put switch S15 to *TEST* before you reboot for HAHA SEP 3.

FAQ about the FPGA

1. Q: Which version of Quartus should I install to program the FPGA?
A: Version 15 or higher. You get download it from Intel Quartus website.
2. Q: Why I cannot detect the chip with the USB Blaster plugin?
A: You should install the driver file for the USB Blaster. The version you install should be compatible with the USB Blaster you use. If you are using the one you got from the TA, you should find the driver file in Quartus 13.1.
3. Q: I created the bitstream file, but why I cannot program with it?
A: You should make sure you chose the correct device part number when you were building the project. You can find the number on the marking of the chip.
4. Q: After I program the FPGA and turn off the board, will the program remain in the FPGA?
A: It depends. If you programmed it with a SOF file, it will be lost after power off. If with a POF file, it will stay.
5. Q: Which kind of file is recommended to program the FPGA?
A: The SOF file is recommended. You can program with a POF only if you are 100% sure your design will do no harm to the board. Each of the I/O ports of your design should be assigned to a pin number.
6. Q: Why that I cannot see any output from the FPGA?
A: You may forget to assign signals to the pins.
7. Q: When I am measuring the temperature with FPGA, I see the result is always 39 in hex, why?
A: You should use the single cycle mode of the sequencer of the ADC. Refer to the datasheet to see how to configure that.
8. Q: When I finish reading the values in the In-System Memory Content Editor and close the window, the whole Quartus crashes, why?
A: Stop auto updating the values before you close the window.

FAQ about the Experiments

1. Q: For a variable in Verilog, where is the LSB?
A: LSB means least significant bit. No matter how the variables are declared, the LSB is always on the right side. For example, the LSB for aaa [1:8] is aaa[8], and the LSB for bbb [63:0] is bbb[0].
2. Q: When I am trying to see the power up values in the SRAM of the Microcontroller, I find every time they stay the same. Why?
A: You must totally power off the board before you power up. Plug out both the USB cable and the USB Blaster.
3. Q: Why I cannot write or read from the EEPROM on the board.
A: You may forget to configure the 'hold' pin of the EEPROM. Please refer to its datasheet for details.
4. Q: Why I cannot detect the Bluetooth module from the computer.
A: You may forget to connect the header P11 with jumpers.
5. Q: In the bus snooping experiment, why don't I see any signal in the waveform?
A: You may try zooming in with a higher time base.
6. Q: Why the ADC in the FPGA is not working well?
A: Try to use the single cycle mode of the sequencer of the ADC.
7. Q: Do I need to use Qsys when implementing the ADC in the FPGA?
A: You do not.
8. Q: When using the temperature sensor in the FPGA, I find the value and temperature relation do not match that of the FPGA datasheet, do I do anything wrong?
A: Some of the chips have an offset in the value – temperature relation. You need to first figure out the true relation of your chip.