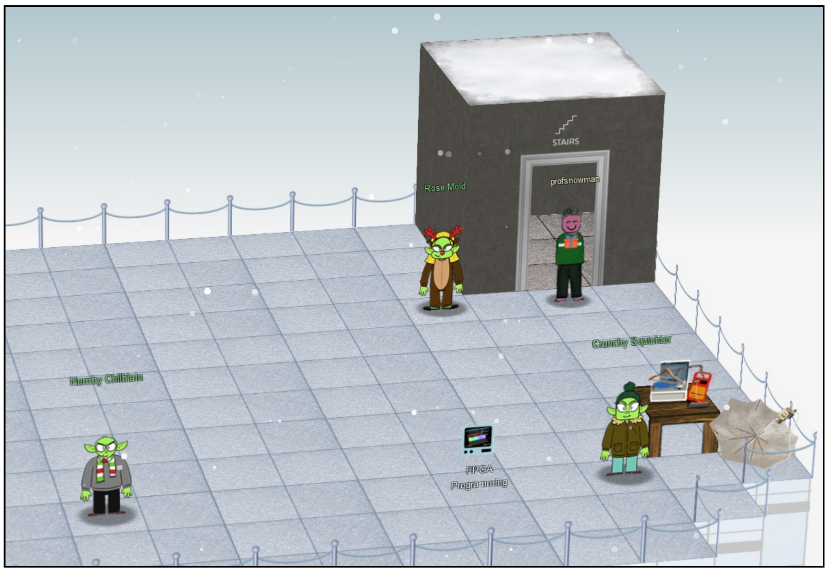
KringleCon 4: Calling Birds!

Frost Tower Rooftop



1. Click to talk to `Rose Mold`

```
I'm Rose Mold. What planet are you from?

Hey, way to go climbing the stairs. You do know you can teleport, right?

Or just use the Frostavator.

n00bs...
```

2. Click to talk to `Crunchy Squishter`

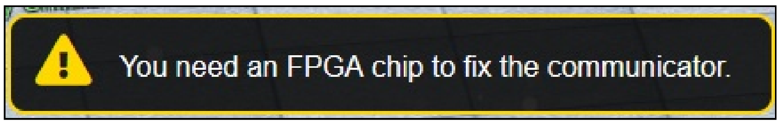
```
Greetings Earthling! I'm Crunchy Squishter.

Hey, could you help me get this device on the table working? We've cobbled it together with
primitive parts we've found on your home planet.

We need an FPGA though - and someone who knows how to program them.

If you haven't talked with Grody Goiterson by the Frostavator, you might get some FPGA tips
there.
```

3. Click the `Communicator`

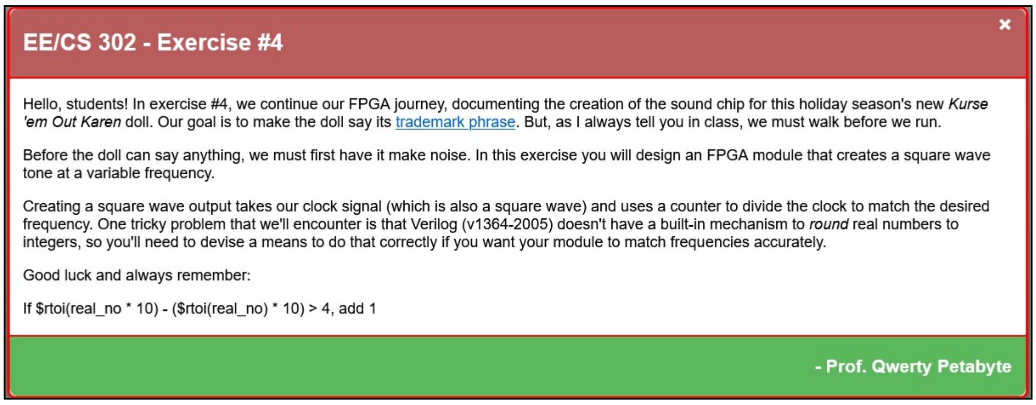

You need an FPGA chip to fix the communicator.

4. Click to talk to `Numby Chilblain`

```
Klatu Barada Nikto!

I'm Numby Chilblain.
```

5. Click `FPGA Programming`



EE/CS 302 - Exercise #4

Hello, students! In exercise #4, we continue our FPGA journey, documenting the creation of the sound chip for this holiday season's new *Kurse 'em Out Karen* doll. Our goal is to make the doll say its trademark phrase. But, as I always tell you in class, we must walk before we run.

Before the doll can say anything, we must first have it make noise. In this exercise you will design an FPGA module that creates a square wave tone at a variable frequency.

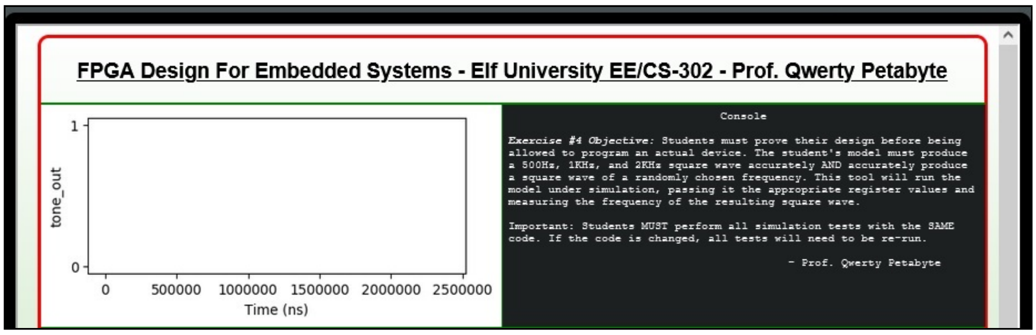
Creating a square wave output takes our clock signal (which is also a square wave) and uses a counter to divide the clock to match the desired frequency. One tricky problem that we'll encounter is that Verilog (v1364-2005) doesn't have a built-in mechanism to *round* real numbers to integers, so you'll need to devise a means to do that correctly if you want your module to match frequencies accurately.

Good luck and always remember:

If $rtoi(real_no * 10) - ($rtoi(real_no) * 10) > 4, add 1

- Prof. Qwerty Petabyte

6. Close the `EE/CS 302 - Exercise #4` window



FPGA Design For Embedded Systems - Elf University EE/CS-302 - Prof. Qwerty Petabyte

```
Console

Exercise #4 Objective: Students must prove their design before being
allowed to program an actual device. The student's model must produce
a 500Hz, 1KHz, and 2KHz square wave accurately AND accurately produce
a square wave of a randomly chosen frequency. This tool will run the
model under simulation, passing it the appropriate register values and
measuring the frequency of the resulting square wave.

Important: Students MUST perform all simulation tests with the SAME
code. If the code is changed, all tests will need to be re-run.

                                       - Prof. Qwerty Petabyte
```

7. Scroll `Down` the window

8. Scroll to the `Top` of the window

9. Replace line 28 onwards with the following code

```
    reg sq_wave;
    integer counter;

    assign wave_out = sq_wave;

    always @(posedge clk or posedge rst)
    begin
        if(rst==1)
           begin
              counter <= 0;
              sq_wave <= 0;
           end
        else
           begin
              if(counter <= 0)
                begin
                   counter <= ((125000000 / $rtoi(freq / 100)) / 2) - 1;
                   if ($rtoi(freq * 10) - ($rtoi(freq) * 10) > 4)
                      begin
                         counter <= counter + 2;
                      end
                   sq_wave <= sq_wave ^ 1'b1;
                end
              else
                begin
                   counter <= counter - 1;
                end
           end
    end
endmodule
```

10. Click the `Simulate 500Hz` button

```
                           Console

Sending code for analysis...
Verilog parsed cleanly...
Beginning FPGA simulation. This may take a few seconds...
Code changed! Resetting some simulation results...
Congratulations!
Simulation results indicate a frequency of exactly: 500.0000Hz
```

11. Click the `Simulate 1KHz` button

```
                              Console

Sending code for analysis...
Verilog parsed cleanly...
Beginning FPGA simulation. This may take a few seconds...
Congratulations!
Simulation results indicate a frequency of exactly: 1000.0000Hz
```

12. Click the `Simulate 2KHz` button

```
                              Console

Sending code for analysis...
Verilog parsed cleanly...
Beginning FPGA simulation. This may take a few seconds...
Congratulations!
Simulation results indicate a frequency of exactly: 2000.0000Hz
```

13. Click the `Simulate Random` button

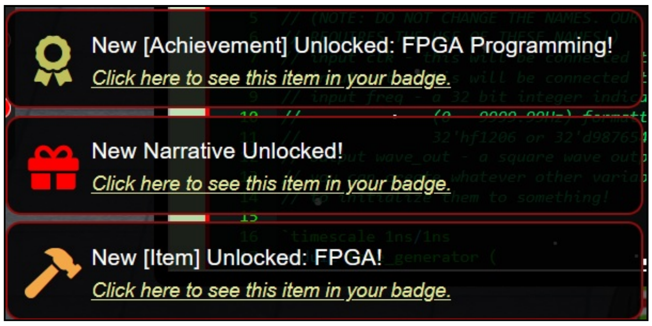
```
                              Console

Sending code for analysis...
Verilog parsed cleanly...
Beginning FPGA simulation. This may take a few seconds...
Random target frequency: 3337.04
Using a clock frequency of 125MHz, the closest you could get to the target frequency is 3337.0709
Simulation results indicate a frequency of: 3337.0709Hz
Congratulations! Your model matches the best-fit value!
```

14. Click the `Program Device` button

```
                              Console

Sending code for analysis...
Verilog parsed cleanly...
Synthesizing/implementing design and generating bitstream.
Bitstream will then be sent to device.
This will take SEVERAL seconds...
The device has been successfully programmed!
```



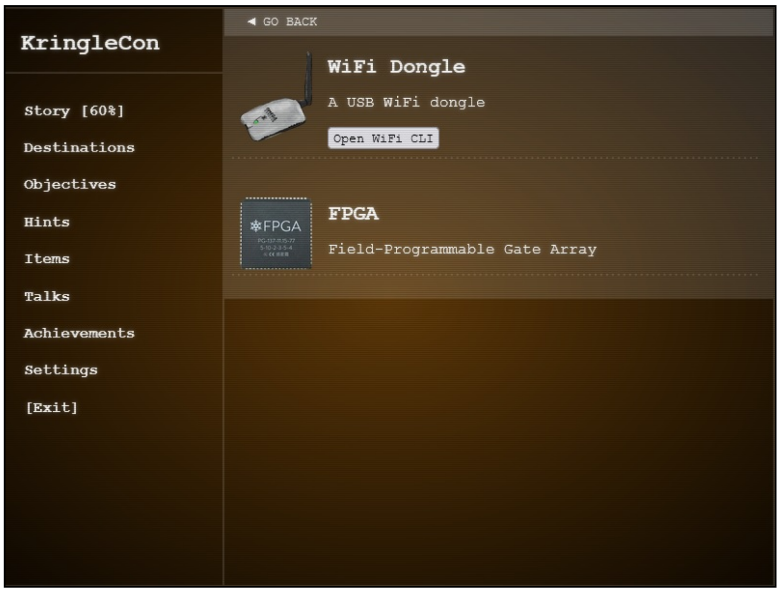
New [Achievement] Unlocked: FPGA Programming!
Click here to see this item in your badge.

New Narrative Unlocked!
Click here to see this item in your badge.

New [Item] Unlocked: FPGA!
Click here to see this item in your badge.

15. Click the `Close` button

16. Click the `Hammer` (Items) icon

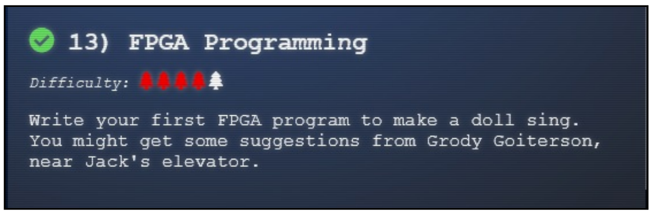


WiFi Dongle
A USB WiFi dongle

FPGA
Field-Programmable Gate Array

17. Click `Objectives` and then scroll `Down`

18. Click `13) FPGA Programming`



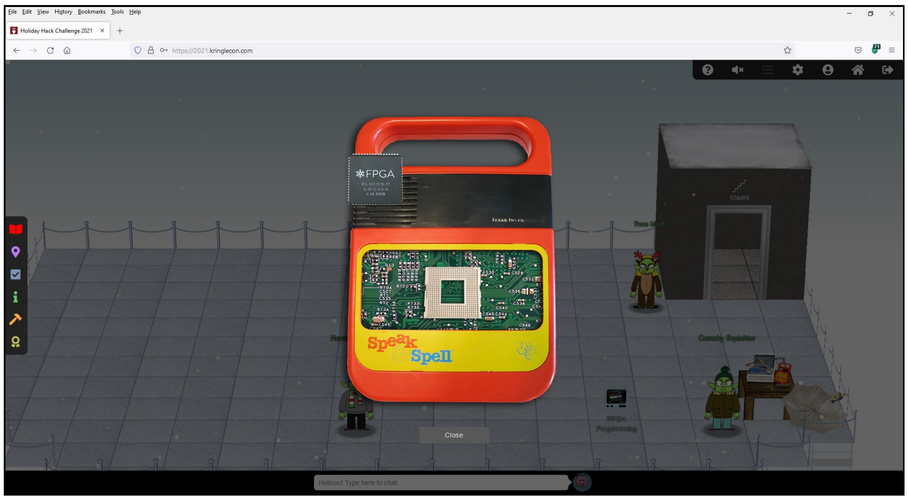
13) FPGA Programming
Difficulty:
Write your first FPGA program to make a doll sing. You might get some suggestions from Grody Goiterson, near Jack's elevator.

19. Click `[Exit]`

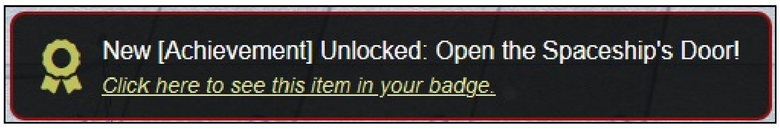
20. Click to talk to `Crunchy Squishter`

```
Thank you! Now we're able to communicate with the rest of our people!
```

21. Click the `Communicator`



22. Place FPGA into the socket


New [Achievement] Unlocked: Open the Spaceship's Door!
Click here to see this item in your badge.

23. Click the `Spaceship`