

# Galileo & DE2i-150 Interface: Photocell Circuit

Team Alpha Centauri:

Phathom Donald

Alexis Wells

4.7.14

EECE494: Computer Bus and SoC Interfacing  
Department of Electrical and Computer Engineering  
Howard University  
Instructor: Dr. Charles Kim

# Idea Formation

Simulation of a monitoring device that shows the state of photocell as low or high.

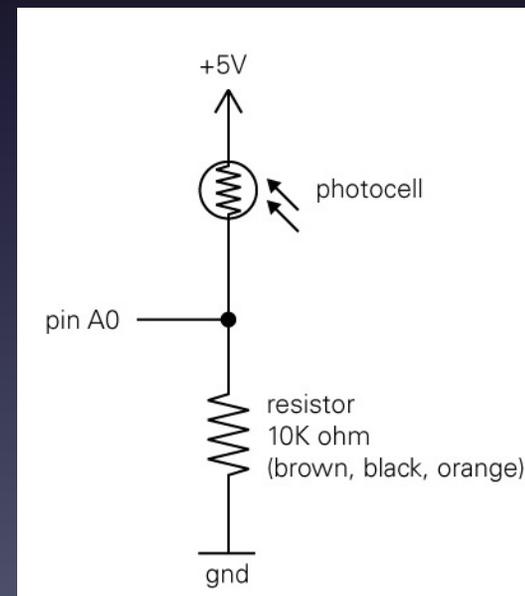
Accomplished through interface of Galileo board & DE2i-150 board, using GPIO expansion of DE2i-150 board.

# Introduction

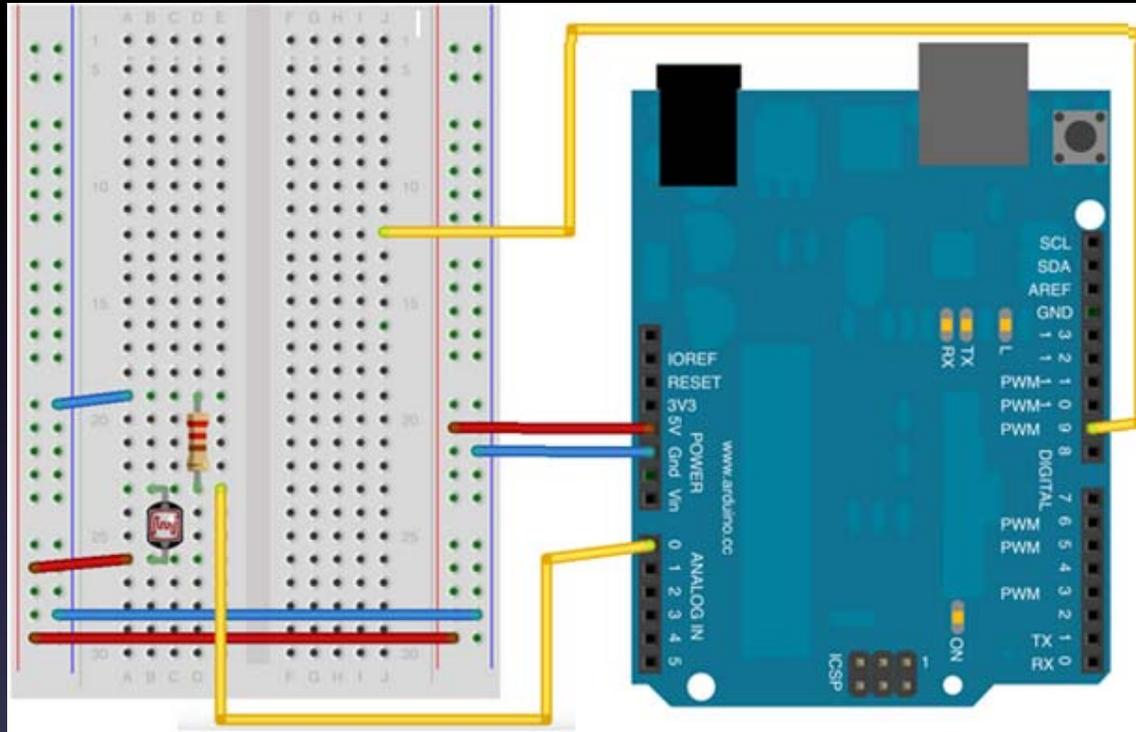
- A message, determined by the Galileo board, will appear on the DE2i-150, notifying the user if a photocell is detecting light or not.
- The photocell is detecting light if the message reads High.
- The photocell is not detecting light if the message reads Low.

# Principle

- **Photocell:** A light-controlled variable resistor. Its resistance decreases with increasing incident light intensity.



# Setup & Materials



## Materials:

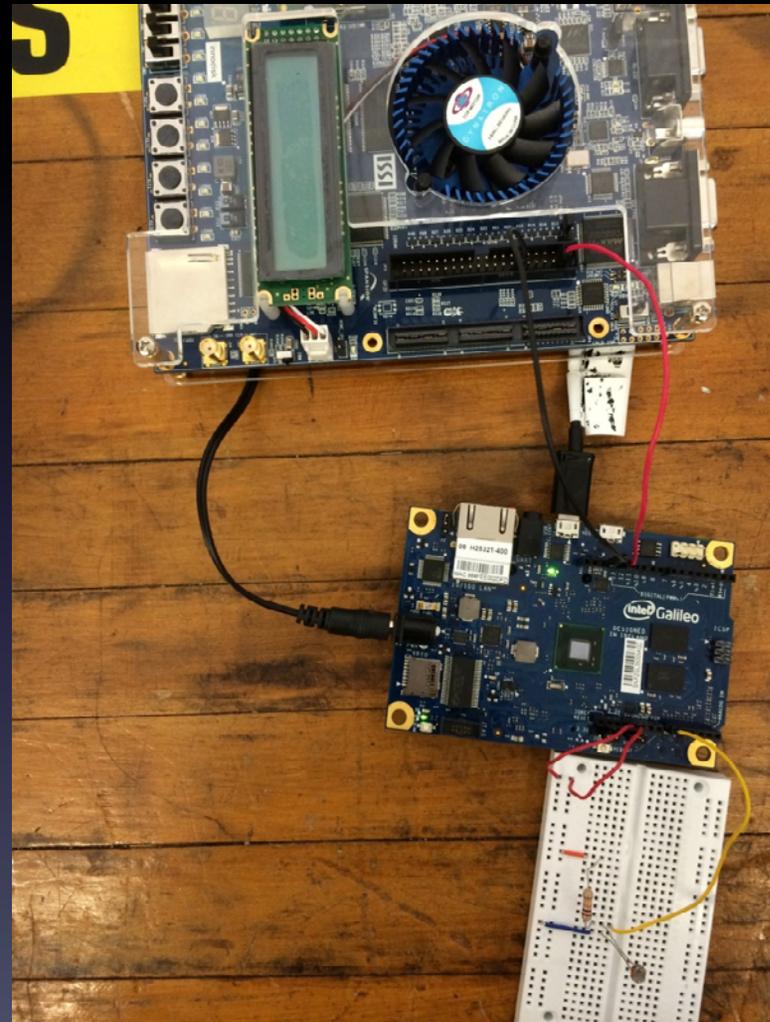
- 1 Galileo board
- 1 Photocell
- 1 Resistor
  - 10 k $\Omega$
- Wires
- 1 DE2i-150 board

## Setup:

- Connect GPIO to pin 13 (PWM) of Galileo board
- Connect of photocell to pin AO (analog input) of Galileo board

# Step 1 – Set Up Circuit

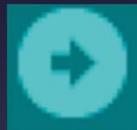
- One pin of the photocell is connected to both GND, through a resistor, and the analog input pin, AO.



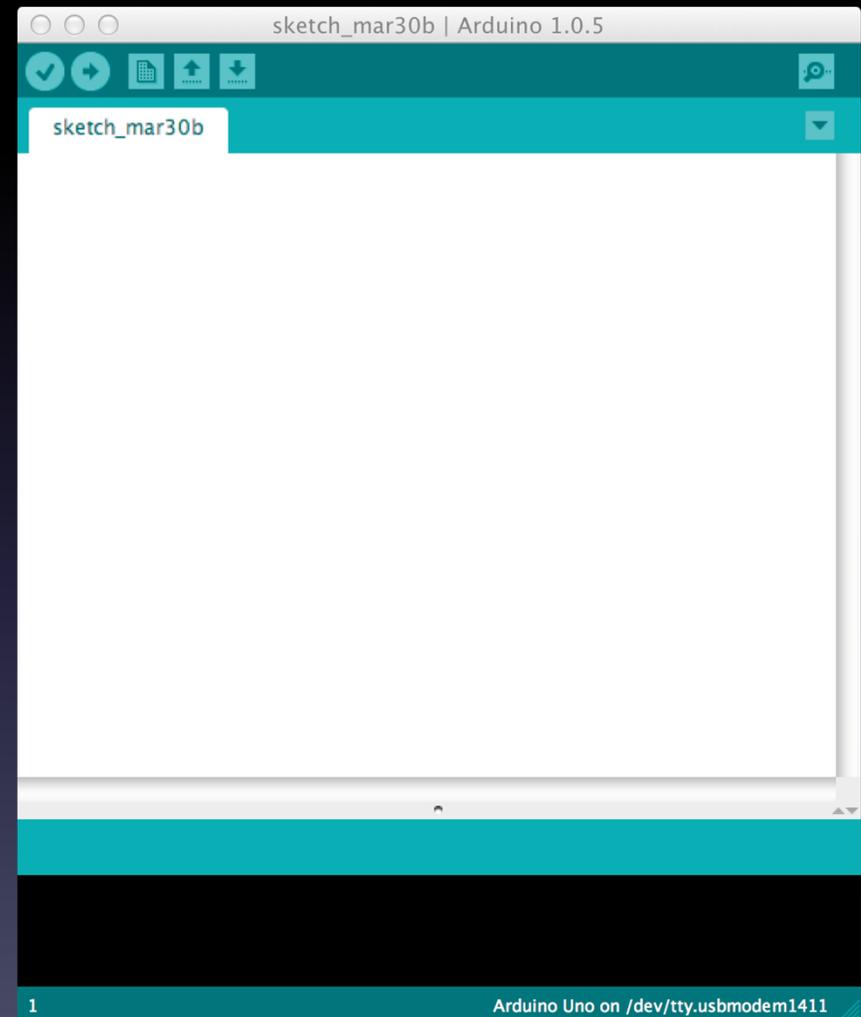
# Step 2 – Open New Galileo Sketch



- Verify: To check for errors in the code



- Upload: To send code to Galileo board



# Step 3 – Pin Assignments

```
int photoPin = A0; // The photocell is connected to pin A0
int gpioPin = 13;  // The GPIO is connected to pin 13
int photoValue;   // The analog reading from the photocell
```

- The photocell acts as an analog input to the Galileo board, and is therefore connected to the analog pin, A0.
- The GPIO acts as an digital output, and is therefore connected to the digital pin, 13.

# Step 4 – Setup Routine

```
void setup() {  
    pinMode(gpioPin, OUTPUT); // Initialize the digital pin as an output.  
}
```

- `pinMode()`: Configures the specified pin to behave either as an input or an output.
- GPIO will be a digital output.

# Step 5 – Loop Routine

```
void loop() {  
  // Read the analog in value:  
  photoValue = analogRead(photoPin);  
  
  if (photoValue < 500) {  
    digitalWrite(gpioPin, LOW); // Send no signal to the GPIO expansion of the FPGA board  
  }  
  else {  
    digitalWrite(gpioPin, HIGH); // Send a signal to the GPIO expansion of the FPGA board  
  }  
  
  delay(10);  
}
```

- analogRead(): Reads the value from the specified analog pin.
- digitalWrite(): Writes a HIGH or a LOW value to a digital pin. Output pin set to 0V (ground) for LOW.
- delay(): Pauses the program for specified amount of time (milliseconds)

# Step 6 – Verify Code

Done compiling.



Binary sketch size: 49,680 bytes (of a 262,144 byte maximum) - 18% used



8

Intel® Galileo on /dev/tty.usbmodem1a121

- Verify the code to check for errors
- Errors would be displayed at the bottom of the window

# Step 7 – Upload to Galileo

```
Done uploading.

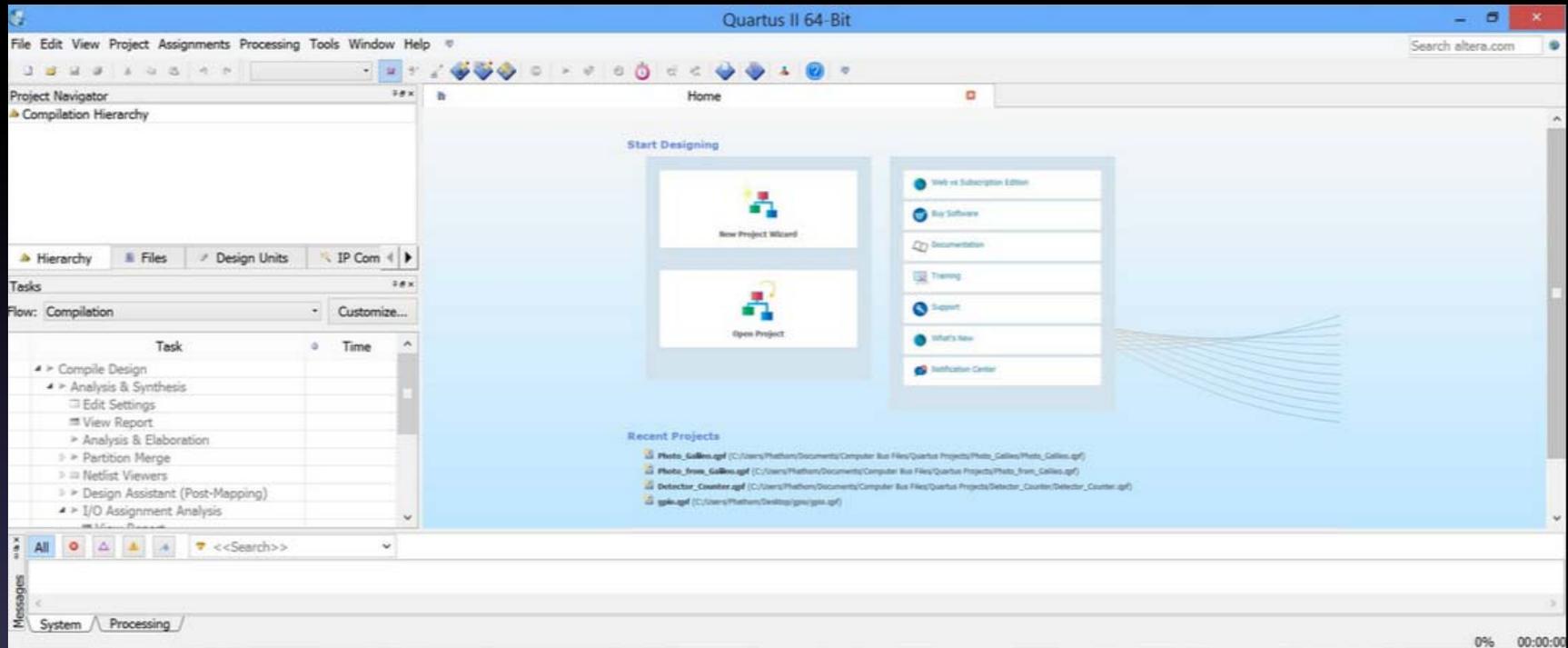
Transfer complete
#
Moving downloaded file to /sketch/sketch.elf on target
#mv the downloaded file to /sketch/sketch.elf
target_download_name="${host_file_name##*/}"
echo "Moving downloaded file to /sketch/sketch.elf on target"
$fixed_path/lsz --escape -c "mv $target_download_name /sketch/sketch.elf; chmod +x /sketch/sketch.elf"
< $tty_port_id > $tty_port_id

Transfer complete
#
#
```

8 Intel® Galileo on /dev/cu.usbmodem1d111

- Send code to Galileo board via USB
- If unknown, serial port can be found by:
  - Tools -> Serial Ports -> /dev/tty.usbmodel1411

# Step 8 – Quartus II Home Page



- Select File -> New Project Wizard
- Click Next

# Step 9 - Directory

**Directory, Name, Top-Level Entity [page 1 of 5]**

What is the working directory for this project?

C:\altera\13.1

What is the name of this project?

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

Use Existing Project Settings...

< Back   Next >   Finish   Cancel   Help

- Enter information about your project
  - Working Directory
  - Name of New Project
- Click Next

# Step 10 – Device Settings

**Family & Device Settings [page 3 of 5]**

Select the family and device you want to target for compilation.  
You can install additional device support with the Install Devices command on the Tools menu.

Device family  
Family: Cyclone IV GX  
Devices: All

Target device  
 Auto device selected by the Fitter  
 Specific device selected in 'Available devices' list  
 Other: n/a

Show in 'Available devices' list  
Package: Any  
Pin count: Any  
Speed grade: Any  
Name filter:   
 Show advanced devices

Available devices:

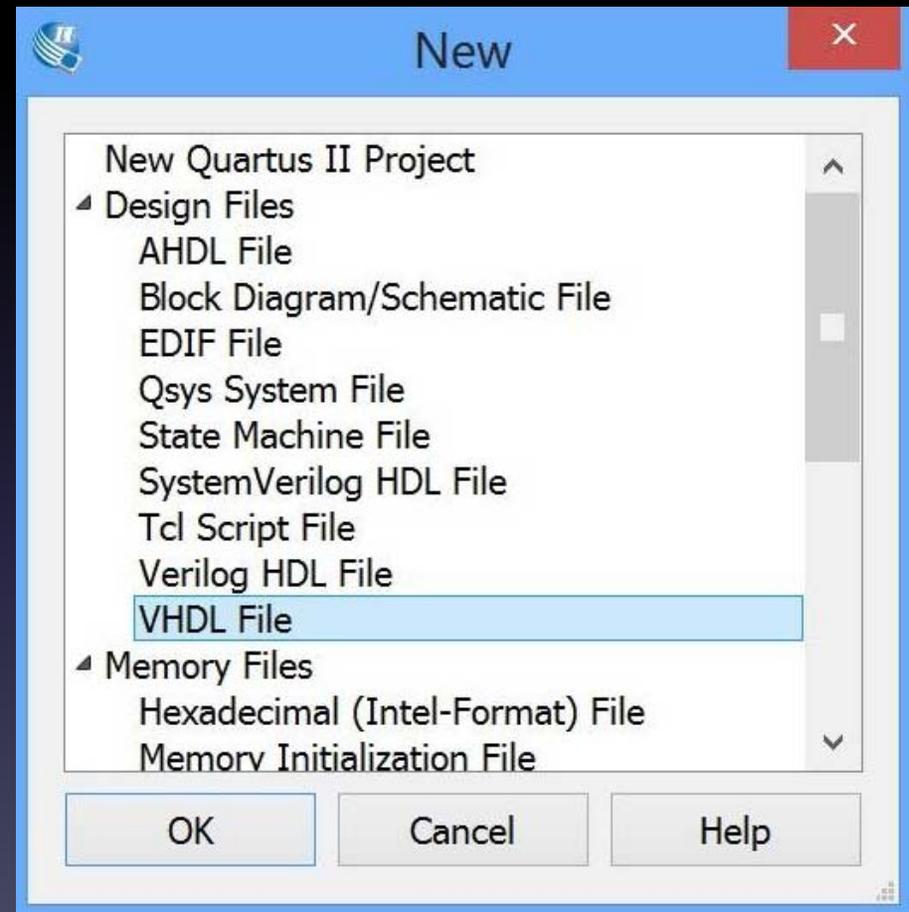
Name	Core Voltage	LEs	User I/Os	GXB Transmitter
EP4CGX150DF31C7	1.2V	149760	508	8
EP4CGX150DF31C8	1.2V	149760	508	8

< Back   Next >   Finish   Cancel   Help

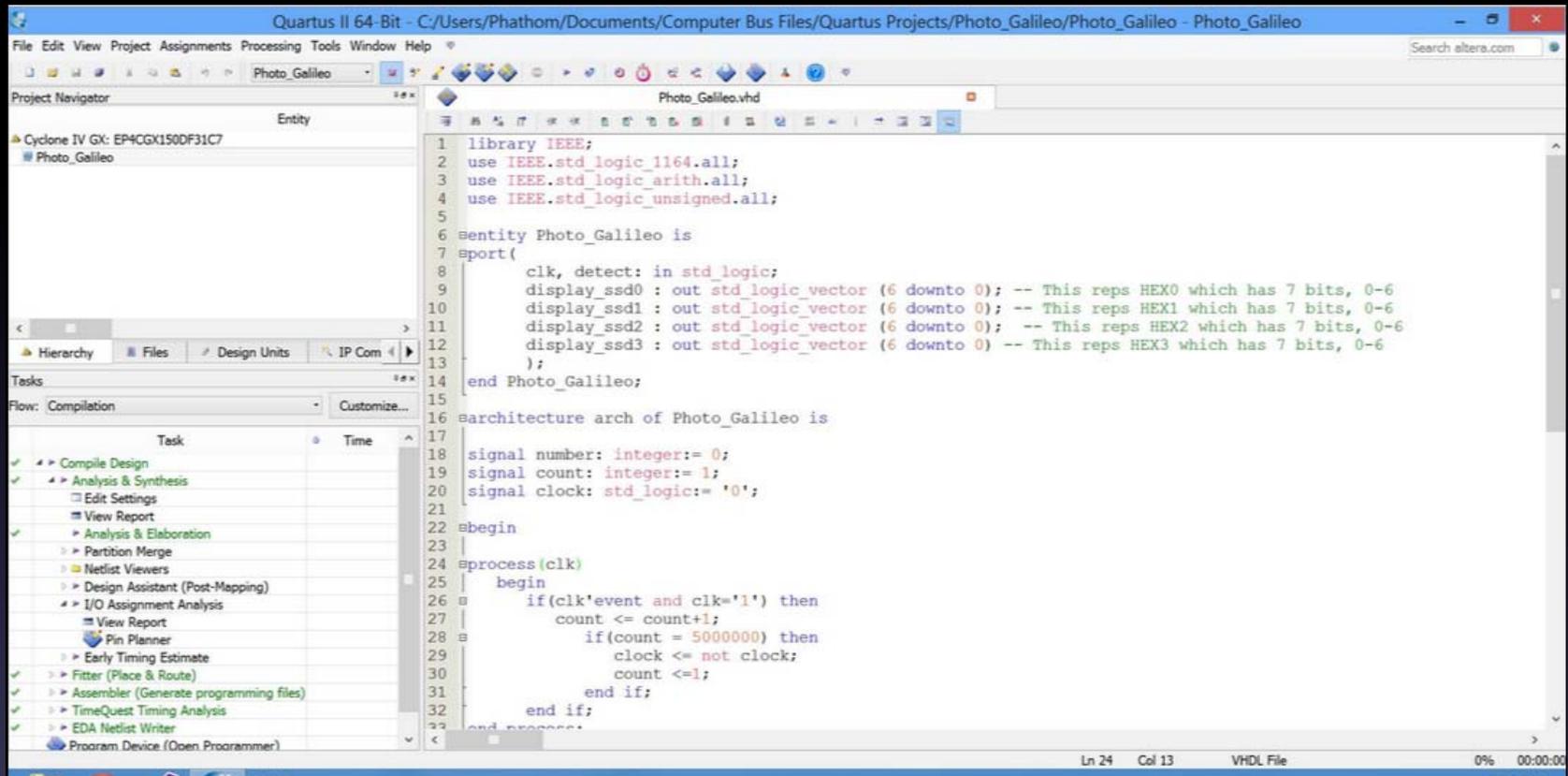
- Assign a Specific FPGA to design and make Pin Assignments
  - Click Finish
  - Click Yes to create project directory

# Step 11 – New VHDL File

- Select File -> New -> VHDL File
- Click OK.



# Step 12 – Write VHDL Code



The screenshot shows the Quartus II 64-bit IDE interface. The main window displays the VHDL code for a project named "Photo\_Galileo". The code defines an entity "Photo\_Galileo" with four outputs: "display\_ssd0", "display\_ssd1", "display\_ssd2", and "display\_ssd3", each of type "std\_logic\_vector (6 downto 0)". The architecture "arch" of "Photo\_Galileo" includes three signals: "number" of type "integer" initialized to 0, "count" of type "integer" initialized to 1, and "clock" of type "std\_logic" initialized to '0'. A process block "process (clk)" is defined, which increments "count" on each clock edge. When "count" reaches 5,000,000, the "clock" signal is toggled, and "count" is reset to 1. The "display\_ssd" outputs are not explicitly assigned in the visible code, but their types suggest they are intended to display the current count.

```
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity Photo_Galileo is
7   port(
8     clk, detect: in std_logic;
9     display_ssd0 : out std_logic_vector (6 downto 0); -- This reps HEX0 which has 7 bits, 0-6
10    display_ssd1 : out std_logic_vector (6 downto 0); -- This reps HEX1 which has 7 bits, 0-6
11    display_ssd2 : out std_logic_vector (6 downto 0); -- This reps HEX2 which has 7 bits, 0-6
12    display_ssd3 : out std_logic_vector (6 downto 0) -- This reps HEX3 which has 7 bits, 0-6
13  );
14 end Photo_Galileo;
15
16 architecture arch of Photo_Galileo is
17
18   signal number: integer:= 0;
19   signal count: integer:= 1;
20   signal clock: std_logic:= '0';
21
22 begin
23
24   process (clk)
25     begin
26       if (clk'event and clk='1') then
27         count <= count+1;
28         if (count = 5000000) then
29           clock <= not clock;
30           count <= 1;
31         end if;
32       end if;
33   end process;
```

- Type VHDL code into Blank Project Space

# Step 12 – Write VHDL Code

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity Photo_Galileo is
port(
    clk, detect: in std_logic;
    display_ssd0 : out std_logic_vector (6 downto 0); -- This reps HEX0 which has 7 bits, 0-6
    display_ssd1 : out std_logic_vector (6 downto 0); -- This reps HEX1 which has 7 bits, 0-6
    display_ssd2 : out std_logic_vector (6 downto 0); -- This reps HEX2 which has 7 bits, 0-6
    display_ssd3 : out std_logic_vector (6 downto 0) -- This reps HEX3 which has 7 bits, 0-6
);
end Photo_Galileo;

architecture arch of Photo_Galileo is

    signal number: integer:= 0;
    signal count: integer:= 1;
    signal clock: std_logic:= '0';

begin

    process(clk)
    begin
        if(clk'event and clk='1') then
            count <= count+1;
            if(count = 5000000) then
                clock <= not clock;
                count <=1;
            end if;
        end if;
    end process;
```

# Step 12 – Write VHDL Code

```
process (clock)
  begin
    if (clock'event and clock='1') then
      if (detect = '1') then
        number <= 1;
      elsif (detect = '0') then
        number <= 0;
      end if;
    end if;
  end process;

process(number)
  begin

    if (number = 0) then
      display_ssd0 <= "1000000";
      display_ssd1 <= "1000111";
      display_ssd2 <= "1111111";
      display_ssd3 <= "1111111";

    elsif (number = 1) then
      display_ssd0 <= "1111001";
      display_ssd1 <= "0001001";
      display_ssd2 <= "1111111";
      display_ssd3 <= "1111111";

    end if;
  end process;
end arch;
```

# Step 13 – Analysis & Elaboration

The screenshot displays the Quartus II 64-Bit software interface. The main window shows the 'Photo\_Galileo.vhd' file. The 'Table of Contents' pane on the left lists various reports, with 'Analysis & Elaboration' selected. The 'Flow Summary' pane on the right provides a detailed overview of the compilation process, including the flow status, version, and various resource counts. The 'Messages' pane at the bottom shows the results of the analysis and elaboration, indicating that the process was successful with 0 errors and 4 warnings.

Flow Status	Successful - Sun Apr 06 19:26:17 2014
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Web Edition
Revision Name	Photo_Galileo
Top-level Entity Name	Photo_Galileo
Family	Cyclone IV GX
Device	EP4CGX150DF31C7
Timing Models	Final
Total logic elements	N/A until Partition Merge
Total combinational functions	N/A until Partition Merge
Dedicated logic registers	N/A until Partition Merge
Total registers	N/A until Partition Merge
Total pins	N/A until Partition Merge
Total virtual pins	N/A until Partition Merge
Total memory bits	N/A until Partition Merge
Embedded Multiplier 9-bit elements	N/A until Partition Merge
Total GXB Receiver Channel PCS	N/A until Partition Merge
Total GXB Receiver Channel PMA	N/A until Partition Merge
Total GXB Transmitter Channel PCS	N/A until Partition Merge
Total GXB Transmitter Channel PMA	N/A until Partition Merge
Total PLLs	N/A until Partition Merge

```
10041 Inferred latch for "display_ssd0[1]" at Photo_Galileo.vhd(46)
10041 Inferred latch for "display_ssd0[2]" at Photo_Galileo.vhd(46)
10041 Inferred latch for "display_ssd0[3]" at Photo_Galileo.vhd(46)
10041 Inferred latch for "display_ssd0[4]" at Photo_Galileo.vhd(46)
10041 Inferred latch for "display_ssd0[5]" at Photo_Galileo.vhd(46)
10041 Inferred latch for "display_ssd0[6]" at Photo_Galileo.vhd(46)
Quartus II 64-Bit Analysis & Elaboration was successful. 0 errors, 4 warnings
```

- Select Process -> Start -> Start Analysis & Elaboration
- Click OK.

# Step 14 – Pin Planner

The screenshot shows the Pin Planner interface for a Cyclone IV GX - EP4CGX150DF31C7. The main window displays a list of node assignments with columns for Node Name, Direction, Location, I/O Bank, REF Group, Standby, Reserved, Output Str, Slew Rate, and Differential. The 'Location' column contains pin numbers from the DE2i-150 board. A 'Top View - Wire Bond' diagram is visible in the upper right corner.

Node Name	Direction	Location	I/O Bank	REF Group	Standby	Reserved	Output Str	Slew Rate	Differential
clk	Input	PIN_AJ16	4	B4_N2	2.5	ult	16m..lt		
detect	Input	PIN_F17	7	B7_N1	2.5	ult	16m..lt		
display_ssd0[6]	Output	PIN_F14	8	B8_N0	2.5	ult	16m..lt	2 (d..ult)	
display_ssd0[5]	Output	PIN_D16	8	B8_N0	2.5	ult	16m..lt	2 (d..ult)	
display_ssd0[4]	Output	PIN_F16	8	B8_N0	2.5	ult	16m..lt	2 (d..ult)	
display_ssd0[3]	Output	PIN_F11	8	B8_N1	2.5	ult	16m..lt	2 (d..ult)	
display_ssd0[2]	Output	PIN_G11	8	B8_N1	2.5	ult	16m..lt	2 (d..ult)	
display_ssd0[1]	Output	PIN_E12	8	B8_N1	2.5	ult	16m..lt	2 (d..ult)	
display_ssd0[0]	Output	PIN_E15	8	B8_N0	2.5	ult	16m..lt	2 (d..ult)	
display_ssd1[6]	Output	PIN_G10	8	B8_N2	2.5	ult	16m..lt	2 (d..ult)	
display_ssd1[5]	Output	PIN_J9	8	B8_N2	2.5	ult	16m..lt	2 (d..ult)	
display_ssd1[4]	Output	PIN_G12	8	B8_N1	2.5	ult	16m..lt	2 (d..ult)	
display_ssd1[3]	Output	PIN_F12	8	B8_N1	2.5	ult	16m..lt	2 (d..ult)	
display_ssd1[2]	Output	PIN_G13	8	B8_N0	2.5	ult	16m..lt	2 (d..ult)	
display_ssd1[1]	Output	PIN_B13	8	B8_N0	2.5	ult	16m..lt	2 (d..ult)	
display_ssd1[0]	Output	PIN_G14	8	B8_N0	2.5	ult	16m..lt	2 (d..ult)	
display_ssd2[6]	Output	PIN_F10	8	B8_N2	2.5	ult	16m..lt	2 (d..ult)	
display_ssd2[5]	Output	PIN_F4	8	B8_N2	2.5	ult	16m..lt	2 (d..ult)	
display_ssd2[4]	Output	PIN_F6	8	B8_N2	2.5	ult	16m..lt	2 (d..ult)	
display_ssd2[3]	Output	PIN_AG30	5	B5_N2	2.5	ult	16m..lt	2 (d..ult)	
display_ssd2[2]	Output	PIN_F7	8	B8_N2	2.5	ult	16m..lt	2 (d..ult)	
display_ssd2[1]	Output	PIN_G7	8	B8_N2	2.5	ult	16m..lt	2 (d..ult)	
display_ssd2[0]	Output	PIN_G8	8	B8_N2	2.5	ult	16m..lt	2 (d..ult)	
display_ssd3[6]	Output	PIN_D4	8	B8_N1	2.5	ult	16m..lt	2 (d..ult)	
display_ssd3[5]	Output	PIN_D5	8	B8_N2	2.5	ult	16m..lt	2 (d..ult)	

- Select Assignments -> Pin Planner
- In location Column add the pin numbers for values of DE2i-150 board
- The pin assignments can be found in the DE2i-150 FPGA System Manual

# Step 15 – Start Compilation

Quartus II 64-Bit - C:/Users/Phathom/Documents/Computer Bus Files/Quartus Projects/Photo\_Galileo/Photo\_Galileo - Photo\_Galileo

File Edit View Project Assignments Processing Tools Window Help

Photo\_Galileo

Tasks

Flow: Compilation Customize...

Task	Time
Compile Design	00:04:26
Analysis & Synthesis	00:00:26
Edit Settings	
View Report	
Analysis & Elaboration	
Partition Merge	
Netlist Viewers	
Design Assistant (Post-Mapping)	
I/O Assignment Analysis	
View Report	
Pin Planner	
Early Timing Estimate	
Fitter (Place & Route)	00:02:22
Assembler (Generate programming files)	00:00:55
TimeQuest Timing Analysis	00:00:36
EDA Netlist Writer	00:00:07
Program Device (Open Programmer)	

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Assembler
- TimeQuest Timing Analyzer
- EDA Netlist Writer
- Flow Messages
- Flow Suppressed Messages

Flow Summary

Flow Status: Successful - Sun Apr 06 19:33:01 2014

Quartus II 64-bit Version: 13.1.0 Build 162 10/23/2013 SJ Web Edition

Revision Name: Photo\_Galileo

Top-level Entity Name: Photo\_Galileo

Family: Cyclone IV GX

Device: EP4CGX150DF31C7

Timing Models: Final

Total logic elements: 55 / 149,760 (< 1 %)

Total combinational functions: 53 / 149,760 (< 1 %)

Dedicated logic registers: 34 / 149,760 (< 1 %)

Total registers: 34

Total pins: 30 / 508 (6 %)

Total virtual pins: 0

Total memory bits: 0 / 6,635,520 (0 %)

Embedded Multiplier 9-bit elements: 0 / 720 (0 %)

Total GXB Receiver Channel PCS: 0 / 8 (0 %)

Total GXB Receiver Channel PMA: 0 / 8 (0 %)

Total GXB Transmitter Channel PCS: 0 / 8 (0 %)

Total GXB Transmitter Channel PMA: 0 / 8 (0 %)

Total PLLs: 0 / 8 (0 %)

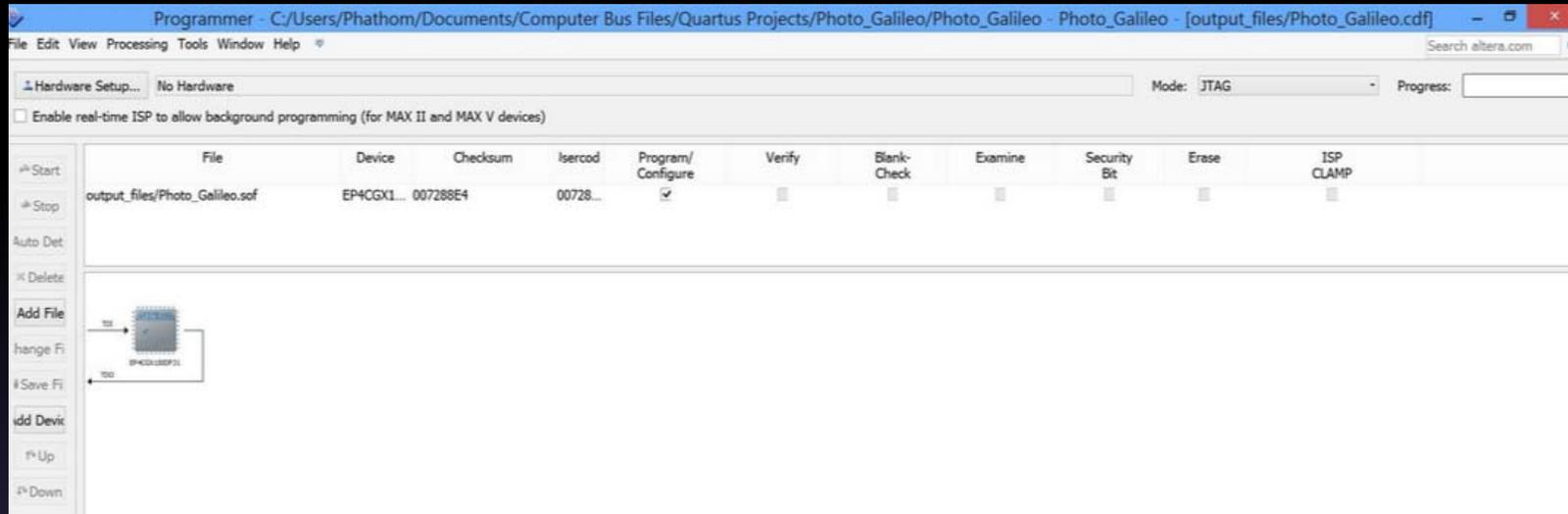
Messages

```
204019 Generated file Photo_Galileo_7_1200mv_0c_slow.who in folder "C:/Users/Phathom/Documents/Computer Bus Files/Quartus Projects/Photo_Galileo/simulation/modelsim/"
204019 Generated file Photo_Galileo_min_1200mv_0c_fast.who in folder "C:/Users/Phathom/Documents/Computer Bus Files/Quartus Projects/Photo_Galileo/simulation/modelsim/"
204019 Generated file Photo_Galileo.who in folder "C:/Users/Phathom/Documents/Computer Bus Files/Quartus Projects/Photo_Galileo/simulation/modelsim/"
204019 Generated file Photo_Galileo_7_1200mv_85c_vhd_slow.sdo in folder "C:/Users/Phathom/Documents/Computer Bus Files/Quartus Projects/Photo_Galileo/simulation/modelsim/"
204019 Generated file Photo_Galileo_7_1200mv_0c_vhd_slow.sdo in folder "C:/Users/Phathom/Documents/Computer Bus Files/Quartus Projects/Photo_Galileo/simulation/modelsim/"
204019 Generated file Photo_Galileo_min_1200mv_0c_vhd_fast.sdo in folder "C:/Users/Phathom/Documents/Computer Bus Files/Quartus Projects/Photo_Galileo/simulation/modelsim/"
204019 Generated file Photo_Galileo_vhd.sdo in folder "C:/Users/Phathom/Documents/Computer Bus Files/Quartus Projects/Photo_Galileo/simulation/modelsim/"
Quartus II 64-bit EDA Netlist Writer was successful. 0 errors, 0 warnings
293000 Quartus II Full Compilation was successful. 0 errors, 32 warnings
```

System Processing (147) 100% 00:04:26

- Select Processing Menu -> Start Compilation or Click Play button
- View Compilation Message & Report.
- Click OK.

# Step 17 - Programmer



- Connect Power supply cable to board and power outlet, Connect USB-Blaster to J<sub>9</sub>, USB cable to USB-Blaster, Connect other end of USB cable to host computer
- Turn on DE2i-150 board
- Select Tools -> Programmer
- Click Hardware Setup
- Select Project file from Directory
- Press Start

# Conclusion

- Very fun and easy project.
- Can be modified for more features.
- Unsure if analog values can be displayed on 7-segment display.

# References

- <http://arduino.cc/en/Reference/analogRead>
- <http://arduino.cc/en/Reference/digitalWrite>
- <http://arduino.cc/en/Reference/delay>
- <http://arduinoarts.com/2011/08/tutorial-led-controlled-by-photo-sensor/>
- <http://learn.adafruit.com/photocells/using-a-photocell>
- <http://arduino.cc/en/Reference/pinMode>