

Ultrasonic Sensor Measurement

SLG47910V, SLG46582V

This abstract discusses how ForgeFPGA along with Power GreenPAK SLG46582V is used to assemble a demo board along with an Ultrasonic Sensor and a 7-segment display to measure distance and read them out. A stand-alone demo board is created to demonstrate the ForgeFPGA functionality.

This application note comes complete with a design file which can be found in the Reference section

Contents

| | |
|------------------------------------|---|
| 1. Terms and Definitions | 1 |
| 2. References | 1 |
| 5. Verilog Code..... | 3 |
| 6. Power GreenPAK SLG46582 | 6 |
| 7. Floorplan: CLB Utilization..... | 7 |
| 8. Schematics..... | 8 |
| 9. Design Steps | 8 |

1. Terms and Definitions

| | |
|------|-------------------------------|
| FPGA | Field Programmable Gate Array |
| GPIO | General Purpose Input Output |
| CS | Chip Select |

2. References

For related documents and software, please visit: <https://www.renesas.com/eu/en/products/programmable-mixed-signal-asic-ip-products/forgefpga-low-density-fpgas>. Download our free ForgeFPGA™ Designer software [1] to open the .ffpga design files [2] and view the proposed circuit design.

[1] [ForgeFPGA Designer Software](#), Software Download and User Guide

[2] [AN-FG-016 Ultrasonic Sensor Measurement.ffpga](#), ForgeFPGA Design File

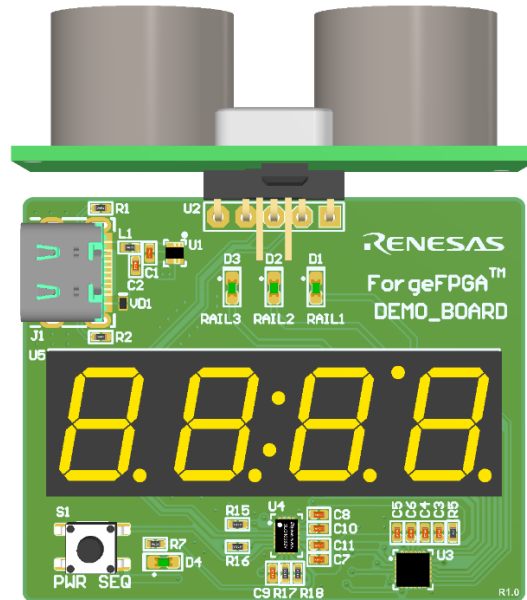
[3] ForgeFPGA SLG47910, Datasheet, Renesas Electronics

[4] Power GreenPAK SLG46582V Datasheet, Renesas Electronics

[5] Ultrasonic Sensor US-100, <https://www.adafruit.com/product/4019>

3. Introduction

ForgeFPGA Demo Board is designed with ForgeFPGA SLG47910 and an Ultrasonic Sensor US-100. The board is used to measure the distance between the sensor and an object which it then displays it in *cm* to the 4-bit seven-segment display attached to the board.



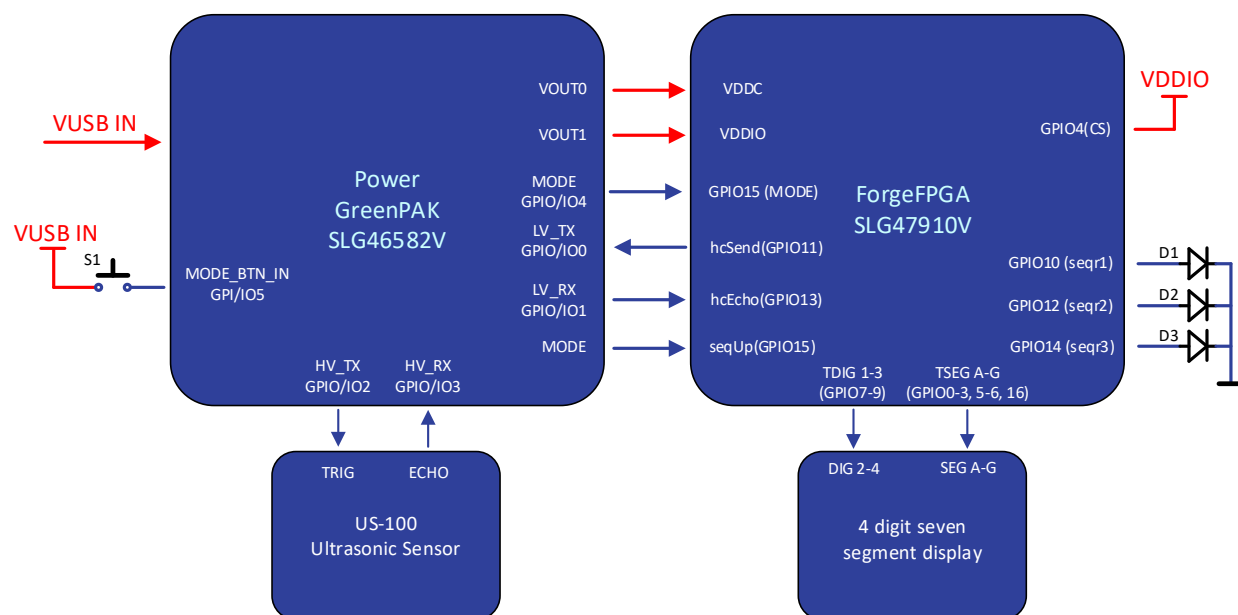


Figure 2. Functional Block diagram

The 4-bit seven-segment display not only shows the distance between the sensor and the object, but also displays a specific code if the object is too close or too far. When an object is within 5 cm, the display shows "CLS," and if the object is more than 3 meters away, the display shows the code "FAR."

The sensor measurement and the power rail LEDs are separate entities and doesn't depend on each other. S1 button marked on the board is used the turn ON/OFF the power sequence.

4. Ingredients

- ForgeFPGA Demo Board
 - ForgeFPGA Workshop Software
 - ForgeFPGA Device SLG47910
 - Power GreenPAK SLG46582V
 - Ultrasonic Sensor US-100
 - 4-bit seven-segment display

5. Verilog Code

Shown below is the main top module of the design. This top module works in coordination with two other sub modules called `customsevenseg.v` and `threeRailSequencer.v` which can be found in the design file attached to this Application Note.

```
// main module
```

```
(* top *) module hcsr04_test #( //HEXFORMAT 0 - Decimal Output 1 - Hex Output
    parameter HEXFORMAT = 0 ) (
```

```
// Main inputs
```

```
(* iopad external pin, clkbuf inhibit *) input clk,
```

```
(* iopad external pin *) input nreset,
```

```
// OSC config outputs
```

```
(* iopad_external_pin *) output osc_en,  
// Custom IO  
(* iopad_external_pin *) input hcEcho, // from the sensor  
(* iopad_external_pin *) output hcEcho_oe,  
(* iopad_external_pin *) output hcSend,  
(* iopad_external_pin *) output hcSend_oe,  
(* iopad_external_pin *) output hcSamplingPin,  
(* iopad_external_pin *) output tssoA,  
(* iopad_external_pin *) output tssoB,  
(* iopad_external_pin *) output tssoC,  
(* iopad_external_pin *) output tssoD,  
(* iopad_external_pin *) output tssoE,  
(* iopad_external_pin *) output tssoF,  
(* iopad_external_pin *) output tssoG,  
(* iopad_external_pin *) output tssoDEC,  
(* iopad_external_pin *) output tdig1,  
(* iopad_external_pin *) output tdig2,  
(* iopad_external_pin *) output tdig3,  
(* iopad_external_pin *) input seqUp, // sequence for the LEDs  
(* iopad_external_pin *) output seqUp_oe,  
(* iopad_external_pin *) output seqr1,  
(* iopad_external_pin *) output seqr2,  
(* iopad_external_pin *) output seqr3,  
(* iopad_external_pin *) output tssOE,  
(* iopad_external_pin *) output tdigOE,  
(* iopad_external_pin *) output seqUpOE  
);  
//OSC  
assign osc_en = 1'b1;  
//OE  
assign hcEcho_oe = 1'b0;  
assign hcSend_oe = 1'b1;  
assign seqUp_oe = 1'b0;  
  
assign tssOE = 1'b1;  
assign tdigOE = 1'b1;  
assign seqUpOE = 1'b1;  
  
//Ultrasonic Sensor Interface  
  
reg [14:0] measure = 15'b000000000000000;  
reg [14:0] rangeVal = 15'b000000000000000;  
reg [3:0] hundreds = 4'b0000;  
reg [3:0] tens = 4'b0000;  
reg [3:0] ones = 4'b0000;  
integer pgmCnt = 0;  
integer samples = 0;  
integer range = 0;  
reg usClk = 0;  
reg pgmClk = 0;  
reg hcOut = 0;  
reg sampling = 0;  
reg readyToSampling = 0;  
reg closeDet = 0;  
reg calculateValue = 0;
```

```
assign hcSend = hcOut;  
assign hcSamplingPin = sampling;
```

```
always @(posedge clk) begin  
    pgmCnt <= pgmCnt + 1;
```

```
    if(pgmCnt < 2700) begin  
        hcOut <= 1;  
    end
```

```
    if(pgmCnt >= 2700) begin  
        hcOut <= 0;  
        readyToSampling <= 1;  
    end
```

```
    if(readyToSampling && hcEcho) begin  
        sampling <= 1;  
        samples <= samples + 1;  
        if(samples > 2941) begin // counts for measure 1 centimeter  
            measure <= measure + 1;  
            ones = ones + 1;  
            if(ones > 9) begin  
                ones = 4'b0000;  
                tens = tens + 1;  
                if(tens > 9) begin  
                    tens = 4'b0000;  
                    hundreds = hundreds + 1;  
                end  
            end  
            samples <= 0;  
        end  
    end
```

```
    end  
  
    if(sampling && !hcEcho) begin  
        sampling <= 0;  
        range <= samples;  
        if(HEXFORMAT) begin  
            rangeVal <= measure;  
        end else if (measure < 5) begin // Error code "CLS" if the distance less than 5  
            centimeter  
            rangeVal[3:0] <= 4'b1101; // S  
            rangeVal[7:4] <= 4'b1110; // L  
            rangeVal[11:8] <= 4'b1100; // C  
        end else if (measure > 300) begin // Error code "FAR" if the distance more than 3 meter  
            rangeVal[3:0] <= 4'b1011; // R  
            rangeVal[7:4] <= 4'b1010; // A  
            rangeVal[11:8] <= 4'b1111; // F  
        end else begin  
            rangeVal[3:0] <= ones;  
            rangeVal[7:4] <= tens;  
            rangeVal[11:8] <= hundreds;  
        end  
        samples <= 0;
```

```
        sampling <= 0;
    end

    if(pgmCnt > 5000000) begin // Overall program timer. Resets all values every x millisecond
(5000000 clocks = 100ms)
        samples <= 0;
        pgmCnt <= 0;
        measure <= 0;
        ones <= 4'b0000;
        tens <= 4'b0000;
        hundreds <= 4'b0000;
        readyToSampling <= 0;
    end
end

//Seven Segment Display Module
customSevenSeg css1 (clk, rangeVal, tssoA, tssoB, tssoC, tssoD, tssoE, tssoF, tssoG, tssoDEC, tdig1,
tdig2, tdig3);

//Three Rail Power Sequencer Module
threeRailSequencer trs1 (clk, seqUp, seqr1, seqr2, seqr3);

endmodule
```

6. Power GreenPAK SLG46582

Below is the circuit implemented on the Power GreenPAK for power sequencing.

PIN 6 in the above circuit is the S1 button on the demo board that is used to turn the sequencer ON or OFF. The circuit above provides the PWR ON for three rail power sequencing (PIN5). This circuit is also used to provide trigger to the Ultrasonic Sensor US-100 and also receives the ECHO back from it through PIN3 and PIN4 respectively.

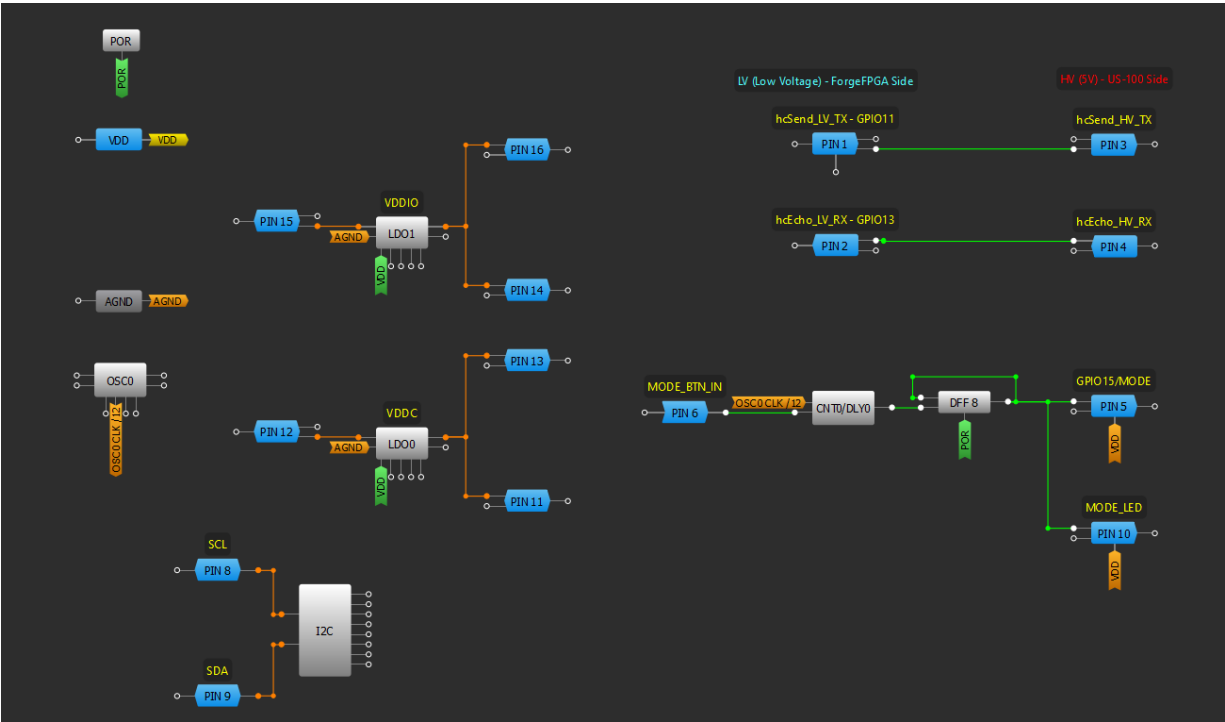


Figure 3. SLG46582 circuit

7. Floorplan: CLB Utilization

The Floor planner tab in the FPGA Editor shows the placement of CLBs and FFs (Figure 4). The resource utilization is shown in the top left corner.

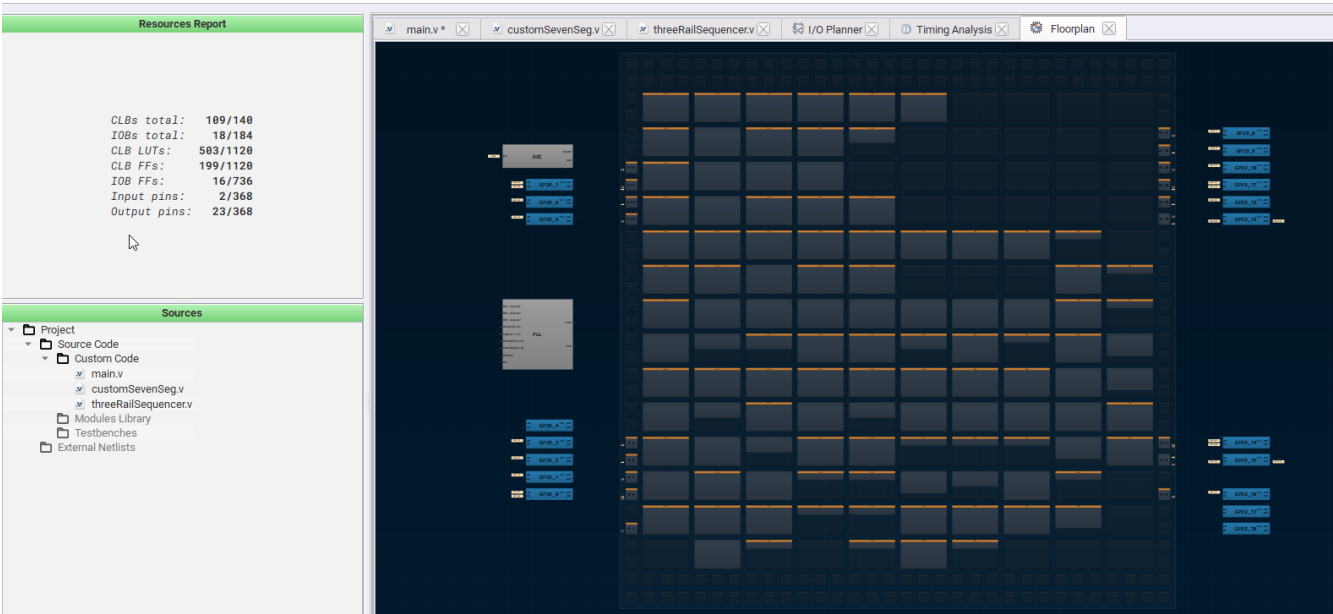
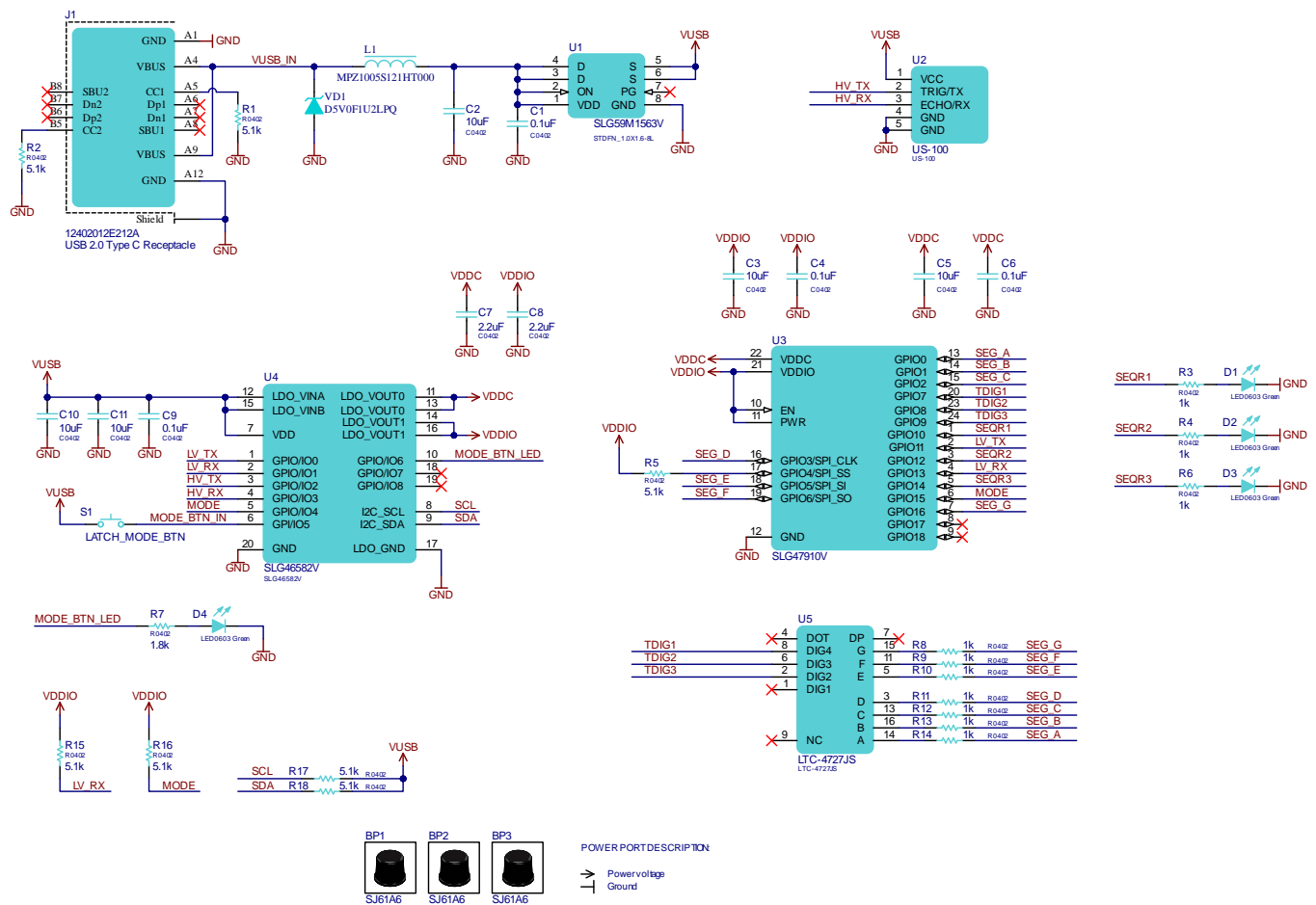


Figure4. Floorplan

8. Schematics



From the schematic above we can observe the different elements used in the Demo Board. It explains the pinout for each element used and also how it connected to each other including the 3 LEDs.

9. Design Steps

1. Launch the latest version of the Go Configure Software Hub. Select the SLG47910V device and the ForgeFPGA Workshop software will load.
2. Download the design example Ultrasonic Sensor Measurement.ffpga. If you are not familiar with the ForgeFPGA Workshop software, review the Four-Bit Counter application notes that cover the basic design steps.
3. Open the Ultrasonic Sensor Measurement.ffpga file after downloading.
4. Open the FPGA editor and review the Verilog code. There is a main code with the module name hcsr04 which is the top module defining the whole design. This code has 2 submodules called customSevenSeg.v and threeRailSequencer.v
5. Open the IO planner tab on the FPGA editor and review the pin assignment.
6. Once the user has familiarized themselves with the code, connect the provided demo board to the system via a USB-C wire.

7. Once connected, the board is ready to use. Select an object from which you need to measure the distance.
8. The onboard Ultrasonic Sensor US-100 will calculate the distance and display it in cm on the 4-bit Seven-Segment display.
9. If the object is too far (more than 3m) then it will display the code FAR on the seven-segment display and if the object is too close, then it will display the code CLS.
10. The user can also observe three LEDs on the board which represent the three rail sequences. The rail sequences are set in the Verilog code which is also programmed in the IC on board.

10. Conclusion

This application note shows how to use the ForgeFPGA demo board to measure the distance between 2 objects using the Ultrasonic Sensor. The demo board also displays three different rail sequences and displays how two independent functionalities can be implemented on the same board using one single ForgeFPGA SLG47910. The design file is available for download (AN-FG-016 Ultrasonic Sensor Measurement.ffpga). If interested, please contact the ForgeFPGA Business Support Team.

11. Revision History

| Revision | Date | Description |
|----------|----------------|------------------|
| 1.00 | August 6, 2024 | Initial release. |