Refrigerator Thermometer Keeping Food at Safe Temperatures

Jacob Monks Rafal Astifo Nathan Gilmer Joseph Asteefan

Introduction

- A Thermometer is an essential tool used to measure temperature.
- They mainly consist of having a sensor to measure temperature and some form of visual to display the temperature reading.
- Many applications, such as refrigeration, rely on thermometers to ensure the appropriate temperatures are maintained.
- □ For our application, a standard thermometer system was developed for refrigerators to help indicate hazardous temperature readings.



Methodology

- The Nexys-A7 board readily comes preinstalled with a temperature sensor.
- □ The output signal readings from the sensor are to be displayed onto a monitor through the use of VGA interfacing.
- □ The visual is developed in an image of a horizontal bar that increases and decreases in accordance with the temperature.
- The temperature range is indicated through the use of colors:
 - $\Box \quad \text{Red} = \text{Overheating (hotest)}$
 - $\Box \quad Blue = Freezing (coldest)$



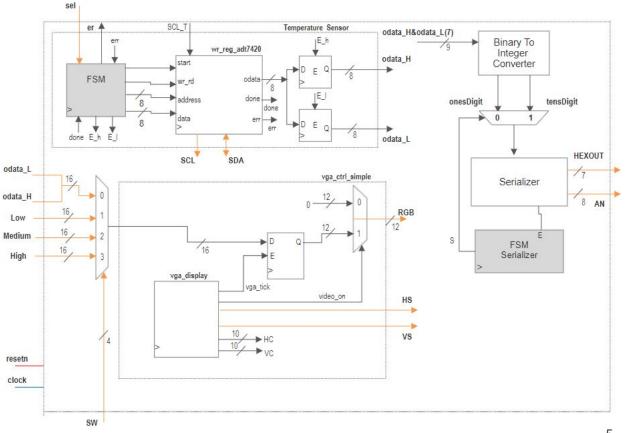
Experimental Setup

- Application was created through the use of VHDL.
- Temperature sensor, VGA controller, and Seralizer are each created having their own set of components, and then interfaced within the top file.
- The VGA display will display the output readings from the temperature sensor, and a set of switches to demonstrate each temperature range:
 - $\Box \qquad \text{Blue (Freezing):} < 28^{\circ}\text{C}$
 - □ Cyan: 28-31°C
 - □ Green (ideal temperature): 31-34 °C
 - □ Orange: 34-37°C
 - $\Box \quad \text{Red (Overheating):} > 37^{\circ}\text{C}$



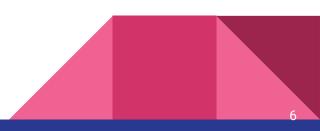
Block Diagram

Inputs/Outputs	Ports
Temperature	Low, Medium, High
Temperature Data Select	SW[0-3]
Temperature Range Select	SW[4]
VGA RGB	R[0-3] G[0-3] B[0-3]
VGA HS	HS
VGA VS	VS
Temperature Sensor SCL	SCL
Temperature Sensor SDA	SDA
7 Segment Display	HEXOUT[6-0]
Serializer Out	AN[7-0]



Temperature Sensor

- The preexisting basic controller code for the temperature sensor was used.
- The code is set to write/read to/from the temperature value registers (8-bit):
 - Register 0x00: Temperature value MSB
 - Register 0x01: Temperature value LSB
- Resulting temperature measurement is recorded into a 16-bit, two's complement, data format.
- Data is set to refresh at 32 ms per reading.
- ADT7420 reads temperatures between -40° C to 150° C.



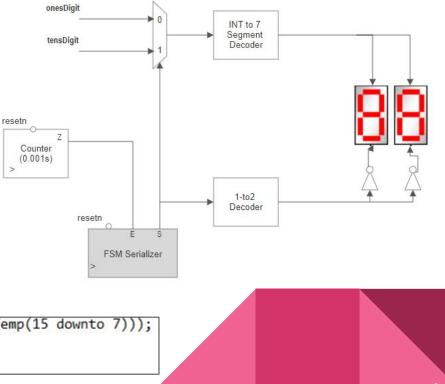
VGA Display

- The SIMPLE version for the VGA controller 12-bit core code was used.
- A 12-bit color is outputted, with each color having 4 bits (RGB).
- The horizontal bar diagram is placed in the center of the display, and each color is fixated at a specified range.
- A monitor with a refresh rate of **30hz** and a resolution of **640x480** was used to display the results.
- The color scheme for the thermometer was defined to resemble a typical thermometer.
- It consists of 5 blocks: Blue, Cyan, Green, Orange and Red. Each block has a height of 96 rows and a width of 320 columns.

Seven Segment Display Serializer

- A seven segment serializer is used to display the temperature. This value is truncated to its whole number.
- Every millisecond, the serializer switches between displaying the ones digit and the tens digit.
- This is so fast that we see both digits on at the same time.
- The temperature was converted to an integer. This integer divided by 10 to find the tens digit, then the mod operator was used to find the ones digit.

```
tempInt <= conv_integer(unsigned(Output_Temp(15 downto 7)));
tens <= tempInt / 10;
ones <= tempInt mod 10;</pre>
```



Process

- 1. The temperature sensor (continuously) reads the temperature, converts the resulting data into Celsius and stores it into the temperature value registers (0x00, 0x01).
- 2. The 9 MSB outputted from the following registers are then sent to the seven segment display, while the whole 16-bit temperature is sent to the VGA controller, to display the resulting temperature.
- 3. The data sent to the VGA control defines the output of the display through two separate processes that executes the data through the use of if-statements:
 - A. Block Height Process: This is explained in the next slide.
 - B. Color/Block Placement Process: If the temperature results are within specific ranges, and if the horizontal and vertical pixels are within a specific range, then the color defined for the temperature range will appear.



Block Height Process

- Considering the VGA has 480 rows, the 5 colors or blocks are equally divided to have 96 rows each
- The red and blue blocks do not gradually increase, as for this application it is not necessary.
- \Box The middle blocks all have a temperature range of 3°C.
- Since we needed to increase or decrease the bar every 4 bits, the temperature reading was subtracted by the minimum temperature value for that block, then divided by 4 to find that block's height.
- □ When setting the v-count parameters, the base height of the block was subtracted by the block's height. This showed the 0.03125°C per row increase on the VGA display.

if (Output_Temp > "0000111000000000" and Output_Temp < "0000111110000000") then --cyan block from 28-31 deg tempSUB <= Output_Temp - "0000111000000000"; --subtracting temp sensor reading for block's base temp i.e. temp sensor-28 cyan_block_h <= "00"&tempSUB(15 downto 2); --dividing number by 4 by bit shifting 2 bits right end if;

Expected Results (Simulation)

SW	Temperature Range (°C)	V-count Range	H-count Range	RGB	Thermostat Output Image
0010 (L)	T < 28	384 to 480	160 to 480	Blue	Blue block
0100 (M)	31< T < 34	288 to 192	160 to 480	Green	Blue, Cyan, Green,
1000 (H)	T > 37	96 to 0	160 to 480	Red	Blue, Cyan, Green, Orange, Red

**Remaining pixels set to white



Red Block Range:

																145.43833	8 us											
Name	Value		0.00002	s	0.00004	4 s	0.00006 s		0.00008	s	0.00010	s	0.0001	2 s 1 1 1 1 1	10.00	014 s	10.000	16 s	0.0	0018 s		0.00020	s	0.00	022 s	l ^o).00024 s	10.000
🔓 resetn	1																											
🔓 clock	1																											
16 sel	1																											
> 😽 SW[3:0]	1000													100	0													
> 😻 odata_H[7:0]	00													00														
> 😻 odata_L[7:0]	00													00														
14 SCL	z			П						П	TT				Т			П	TT	П	Т		П				TTT	TT
14 SDA	Z							TT																				
> 😻 R[3:0]	f	f	χ.	X	f	X	• X	f	X	0	f		XoX		f	χo	X	f		XoX		f	Х	ο χ		f	χ.	χ _f
> 😽 G[3:0]	0	f	χ	χfχ	0	χfχ	0 X f X	0	X f X	0 / f	X O) f	τ ν ο χ	fX	0	X f X O	fX	0	f	XoX	f	0	X f X	ōχ	fX	0	χf χ 0	χf χο
> 😻 B[3:0]	0	f	χo	X f X	0	X f X	0 X f X	0	f X	0 1	X O	X f	X o X	f	0	X f X O	f	0	f	XoX	f	0	X f X	ōχ	fX	0	X f X O	χf χο
HS	1													_											_			
1∎ VS	1																											
> 😻 hcount_to_top[9:0]	433																											
> Vcount_to_top[9:0]	4		0	X	1		Ĺχ	2	2	X		3	X		4		X	5		X		6		Ĺχ		7		χ 8
> 😻 hexout[6:0]	01													01														
> 😻 AN[7:0]	fe													fe														
																												12

Orange Block Range:

											3,216.471667 1	us			
Name	Value		0.00306 s	0.00308 s	0.00310 s	0.00312 s	0.00314 s	0.00316 s	0.00318 s	0.00320 s	0.00322 s	0.00324 s	0.00326 s	0.00328 s	10.003
🔓 resetn	1														
le clock	0														
🔓 sel	8			27.				D.				27,			
> 😽 SW[3:0]	1000							10	00						
> 😻 odata_H[7:0]	00)0						
> 😻 odata_L[7:0]	00		4)0						
16 SCL	0			тптт	TTTT										ТП
16 SDA	z														
> 😻 R[3:0]	e	f		f	(0) f (e	(f) (f e (f (0 / f /	e f 0	f e	f X 0 X 1	t e (f	() (f)	• (f (0	(f)(e
> 😽 G[3:0]	7	0	f (0)	f) 0 (f) 7	f X O X	f 7 1	fOff	7 (f (0	f 7	f (0 (1	:) 7 (f) 0 (f)	7 (f) 0	f/7
> 😻 B[3:0]	0	0	f ()	f) () (f) (c		f 0 (f)0)f)	0 (f (0	f (0) f X O X 1	t <u>0 t</u>) () (f)) f (0	<u>f</u>
HS	1														
1 6 VS	1														
> 😻 hcount_to_top[9:0]	409														
> Vcount_to_top[9:0]	100		95 X	96	X	97)	98	X	99	10	•	101		102	103
> 😽 hexout[6:0]	4c							4	le						
> 😽 AN[7:0]	fd								id						
															13

Green Block Range:

														6,304.105	000 us	
Name	Value		0.00610 s	10.	.00612 s	0.00614 s	0.00616 s	0.00618 s	0.00620 s	0.00622 \$	0.00624 s	0.00626 s		0.0 <mark>0630 s</mark>	0.00632 s	lo: 006
🔓 resetn	1															
谒 clock	1															
🔓 sel	1															
> 😽 SW[3:0]	1000								1	000						
> 😽 odata_H[7:0]	00									00						
> 😽 odata_L[7:0]	00									00						
16 SCL	0			Т				TTTT								
14 SDA	Z															
> 😻 R[3:0]	0	e	f)0	f	e (f)	0 (f		f 0 1	f 0 f	0 (f (0	f 0	f (0) f	X 0 X f X 0) <u>f</u> (0	<u>f</u>
> 😽 G[3:0]	0	7	f)0	f	7 (f)	0 (f		f		f (0	f		f X		f ()	f
> 😽 B[3:0]	0	0	fXOX	f	0 (f)				f 0 f	0 (f (0	f 0) f (0) f	X 0 X f X) <u>(f</u>)	<u>t</u> (0
l∎ HS	1				1999) - 1999)											
VS	1															
> 😽 hcount_to_top[9:0]	0															
> vcount_to_top[9:0]	197		.90 X		191	X	192	193		194	19	5 X	196		197	198
> 😻 hexout[6:0]	01									01						
> 😽 AN[7:0]	fe									fe						444
																14

Cyan Block Range:

													9,376.105000 us	
Name	Value	ana fara-	0.00918 s	0.00920 ≤	0.00922 s	0.00924 s	0.00926 s	0.00928 s	0.00930 s	0.00932 s	0.00934 s	0.00936 s	0.00938 s	0.00940 s
🐻 resetn	1													
1 clock	1													
16 sel	1	2												
> 😻 SW[3:0]	1000							1000						
> 😻 odata_H[7:0]	00							00						
> 😻 odata_L[7:0]	00							00						
16 SCL	Z													
16 SDA	Z													
> 😽 R[3:0]	0	0 (f)			f	VO (f)	0 (f) 0	f 0	f 0 f	0 (f)	0 f (0	f (0	f 0	f 0 f 0
> 😽 G[3:0]	0	f		f (0)	f	, 0 (f)	d (f) 0	f d) f (0 (f)	d (f)		f \ O	f d	$f \sqrt{0} \sqrt{f} \sqrt{d}$
> 😼 B[3:0]	0	0 (f)			f		f ()	f		f)		X •	f	VOV f
1 å HS	1													
l∎ vs	1													
> 😻 hcount_to_top[9:0]	0													
vcount_to_top[9:0]	293	286	X	287 X	288		289	290	X	291		292	293	X 294
> 😻 hexout[6:0]	4c							4c						
> 😼 AN[7:0]	fd							fd						
														15

Blue Block Range:

	rest:													12,466.9383	333 us	
Name	Value	6	0.01226 s	0.01228 s	0.01230 s	0.01232 s	0.01234 s	0.01236 s	0.01238 s	0.01240 s	0.01242 s	0.01244 s	0.01	46 s 0	0.01248 s	10
🐻 resetn	1															
🐻 clock	1															
16 sel	1															
> 😽 SW[3:0]	1000			1			5	1000		2	2	4				
> 😽 odata_H[7:0]	00							00								
> 😼 odata_L[7:0]	00		<u>.</u>		*			00	-		2					
1 SCL	Z															
16 SDA	Z															
> 😽 R[3:0]	0	f 0 f) 0 (f		f (0	(f) (0)	f)0 f	f	0 (f) 0	f \) 0 (f)	0	(f) (f 0	ſ
> 😽 G[3:0]	0	f 0 f	d (f		f (0	(f) (0)	f 0 f	0 (f (0 (f) 0) 0 (f)	0	(f)O	f 0	ſ
> 😽 B[3:0]	f	f 0	f		f (0	f		f	0 (f	XX	f		f		f	
14 HS	1															
1⊌ VS	1															
> 😽 hcount_to_top[9:0]	470															
> 😻 vcount_to_top[9:0]	389	382	383	<u> </u>	384	385	χ	386		87 X	388	χ	389	X	390	
> 😽 hexout[6:0]	01							01								
> 😽 AN[7:0]	fe						1	fe								
															16	

Live Demonstration



Conclusion/Future Improvements

- Temperature range for this project was determined to demonstrate the overall workings of our application, and may be adjusted accordingly for future uses.
- □ The display of the temperature value can be modified to appear on the display rather than the board to better improve the overall thermometer.
- Showing the color density change as it increases/decreases to show the different shades for each temperature block can better demonstrate the overall temperature.
- □ The temperature sensor implemented with the board may be accurate but a better sensor can be used to decrease some of the fidgeting noticed from the sensor.
- Possibly including noise, or warning texts to the display to better fit the needs of the user.

References

- "Are You Storing Food Safely?" U.S. Food and Drug Administration. 9 February, 2021. <u>Are You Storing Food Safely?</u>
- Brown, Arthur. "Nexys A7 Reference Manual." Nexys A7 Reference Manual -Digilent Reference, Digilent, Nexys A7 Reference Manual
- "Celsius to Fahrenheit (°C to °F)." *Metric Conversion*, Wight Hall Ltd., 2020, <u>www.metric-conversions.org/temperature/celsius-to-fahrenheit.htm</u>
- Llamocca, Daniel. VHDL Coding for FPGAs, www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html

