

Temperature Display with FPGA

Andy Alvarez
6140523

Robert Lopez
6058102

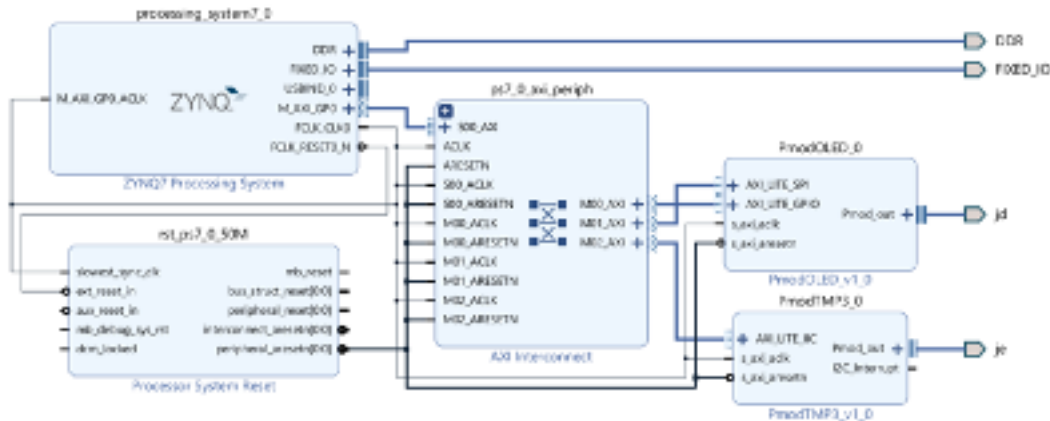
Dr. Atoussa Tehrani
EEL 4740



Objectives

- To display temperature on a user-friendly display.
- To control a temperature sensor with the FPGA board.
- To control an OLED display with the FPGA board.
- To interface the temperature sensor with the oled display.

Schematic



Materials

- Digilent Zybo Z7 FPGA Board
- Digilent Pmod TMP3
- Digilent Pmod OLED
- 4 Male to Female Jumper Wires



Method

- Add Vivado Library with necessary Pmod source files into IP Repository
- Create Block Design with Zybo Z7 Processing System and Pmod IPs
- Run Block Automation
- Validate Design
- Configure Pmod TMP3 to one of the Pmod ports (JE) on FPGA
- Configure Pmod OLED to one of the Pmod ports (JD) on FPGA
- Create VHDL Wrapper
- Export Hardware to Vitis
- Create Empty Application
- Implement C code to output the data obtained from the sensor onto the display

Temperature Sensor

```
122 void DemoRun() {
123     double temp = 0.0;
124     double temp2 = 0.0;
125     double temp3 = 0.0;
126     u8 *pat;
127     while (1) {
128         temp = TMP3_getTemp(&myTMP);
129         temp2 = TMP3_CtoF(temp);
130         temp3 = TMP3_FtoC(temp2);
131
132         // Temperature 2 is in Fahrenheit
133         int temp2_round = 0;
134         int temp2_int = 0;
135         int temp2_frac = 0;
136         // Round to nearest hundredth, multiply by 100
137         if (temp2 < 0) {
138             temp2_round = (int) (temp2 * 1000 - 5) / 10;
139             temp2_frac = -temp2_round % 100;
140         } else {
141             temp2_round = (int) (temp2 * 1000 + 5) / 10;
142             temp2_frac = temp2_round % 100;
143         }
144         temp2_int = temp2_round / 100;
145
146         // Temperature 3 is in Celsius
147         int temp3_round = 0;
148         int temp3_int = 0;
149         int temp3_frac = 0;
150         if (temp3 < 0) {
151             temp3_round = (int) (temp3 * 1000 - 5) / 10;
152             temp3_frac = -temp3_round % 100;
153         } else {
154             temp3_round = (int) (temp3 * 1000 + 5) / 10;
155             temp3_frac = temp3_round % 100;
156         }
157         temp3_int = temp3_round / 100;
```

Pmod Interfacing

```
OLED_ClearBuffer(&myOLED);
OLED_SetCursor(&myOLED, 0, 0);
OLED_PutString(&myOLED, "77 Degrees F");
OLED_SetCursor(&myOLED, 0, 1);
OLED_PutString(&myOLED, "25 Degrees C");
```

```
OLED_PutString(&myOLED, "%d.%d Degrees F", temp2_int, temp2_frac);
too many arguments to function 'OLED_PutString'
OLED_PutString(&myOLED, "%d.%d Degrees F", temp2_int, temp2_frac);
```

```
163 // Casting temperature values to string
164 char temp2intSTR[10] = {0};
165 char temp2fracSTR[10] = {0};
166 char temp3intSTR[10] = {0};
167 char temp3fracSTR[10] = {0};
168 itoa(temp2_int, temp2intSTR, 10);
169 itoa(temp2_frac, temp2fracSTR, 10);
170 itoa(temp3_int, temp3intSTR, 10);
171 itoa(temp3_frac, temp3fracSTR, 10);
```

Integer to
be
converted

Variable
where string
will be stored

Number
Base (10 for
Decimal)

OLED

```
173 // Choosing Fill pattern 0
174 pat = OLED_GetStdPattern(0);
175 OLED_SetFillPattern(&myOLED, pat);
176 // Turn automatic updating off
177 OLED_SetCharUpdate(&myOLED, 0);
178
179
180 OLED_ClearBuffer(&myOLED);
181 OLED_SetCursor(&myOLED, 0, 0);
182 OLED_PutString(&myOLED, temp2intSTR);
183 OLED_PutString(&myOLED, "F");
184 OLED_SetCursor(&myOLED, 0, 2);
185 OLED_PutString(&myOLED, temp3intSTR);
186 OLED_PutString(&myOLED, "C");
187 OLED_Update(&myOLED);
188 } // end while
189 } // end DemoRun
```

Troubleshooting

- We originally did not know that we had to include the Vivado Library in the IP Repository in the settings of Vivado
- After not being able to implement the Pmod onto our block design, we realized this was the issue
- Our implementation of the OLED_PutString function was originally buggy as we tried to pass the temperature values as integers
- Through the itoa function in C we were able to convert int to string and pass it onto the OLED display

- Originally we had planned to work solely in Vivado without having to export to Vitis
- We realized we needed to output the temperature data in a user-friendly manner + interface with OLED
- Therefore C code needed

Conclusions

- Digilent provides very valuable information about each Pmod available through the internet
- Vivado Library folder must be added to IP Repository
- OLED reflects changes in temperature and refreshes automatically
- Important to analyze header files and function prototypes in C
- Overall, this project showed us how to implement multiple Pmod through a fun and interesting way

