

Smart Night Light

Matt Ball, Aidan Faraji-Tajrishi, Thomas Goold, James Wallace
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI

mjball@oakland.edu, atfarajitajrish@oakland.edu, tegoold@oakland.edu, jhwallace@oakland.edu

Abstract— Our project is called Smart Night Light. It is intended to replace traditional night lights. Design and implementation proved to be more involved than originally anticipated, requiring several concepts from higher level classes, such as Serial Peripheral Interfaces (SPI). While in the preliminary stages of development, Smart Night Light shows promise of being a reliable alternative to traditional night lights. With more time and resources, the Smart Night Light could render the old way of lighting obsolete.

I. INTRODUCTION

The smart night light is meant to be an upgrade to existing night lights. A typical night light lacks customization by offering only one color at one level of brightness. Our smart night light attempts to give users more freedom by offering many colors at varying levels of brightness. Our commitment to customization is taken even further by also offering two different operating modes: Automatic and Manual. Auto senses the ambient light and adjusts the brightness accordingly, while Manual gives the user the ability to set the light to full brightness. Auto utilizes a PMOD Ambient Light Sensor (ALS) which converts light to digital data with 8-bit resolution. The 8 bits are stored in registers which can then be multiplexed to select the color. While the brightness of the LED's in Auto is controlled by the sensor, the color is selected by the user through switches. Once the user selects the desired color, the light is output to the LED's through Pulse Width Modulation (PWM). If the user selects Manual mode with the integrated switches, the light sensor is bypassed and the chosen color is set to maximum brightness by separate registers. This is the advantage of the Smart Night Light over typical night lights.

II. METHODOLOGY

The design and implementation of this project required the application of concepts taught throughout the semester. This project also required some topics more advanced than the scope of this course. Through the use of combinational circuitry, synchronous sequential circuitry, and digital system design, the group was able to implement the proposed idea into a functioning product. The individual building “blocks” of the project will be described in the proceeding sections.

A. Finite State Machine (FSM)

The design of the FSM for the Ambient Light Sensor (ALS) was the main design hurdle the group faced. The use of the ALS required the creation of a circuit that would utilize a Serial Peripheral Interface (SPI), which is above the scope of this course. After receiving guidance from their professor, the group was able to design and implement their FSM. The state diagram for the FSM can be seen below in Figure 1.

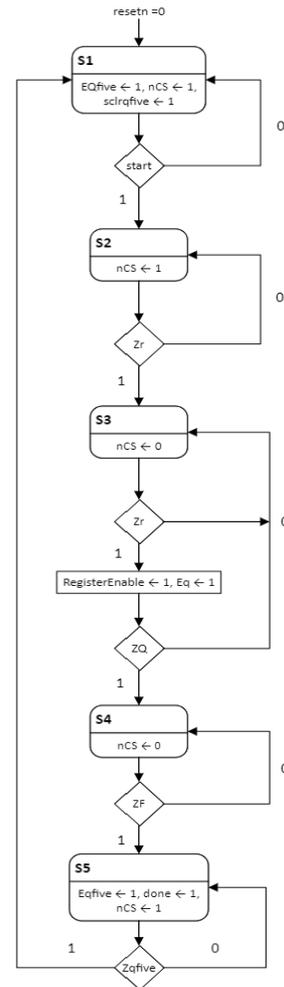


Figure 1: The state diagram for the FSM of the ALS.

The FSM enters State 1 (S1) when the signal resetn is equal to 0. While in S1, the signals Eqfive, nCS, and sclrfive are all set to 1. The signal Eqfive is the enable for the 10-bit counter contained in S5, nCS is chip select for the ALS, sclrfive is the synchronous clear for the 10-bit counter in S5. The transition between S1 and S2 is made when the condition start is equal to 1. In our application, start will always be 1, but start could be configured in future applications to be 0. If this were the case, there could be other conditions that must be met before S2 would be reached. While in S2, nCS remains high and the condition Zr must be 1, before S3 can be reached. Zr is used to detect the rising edge of the SPI clock, sclk. When a rising edge is detected, nCS will be switched to 0 and the ALS will begin reading data. Whenever Zr is 1, the signals RegisterEnable and Eq are 1. RegisterEnable is the enable on the shift register that alerts the register to begin collecting data and Eq is the enable on the 16-bit counter. The FSM will now wait for the signal ZQ, which is the output of the 16-bit counter, to equal 1 before entering S4. While in S4, nCS is still equal to 0. The sensor will continue to read until a falling edge is detected on sclk. When a falling edge is detected, signal ZF will equal 1 and the FSM will enter S5. When in S5, signals Eqfive, done, and nCS will be equal to 1. The signal done infers that 16-bits of data have been received and S5 has been reached. The final condition in S5 is that Zqfive must be equal to 1 before the FSM returns to S1. Zqfive is the output of the 10-bit counter used in S5 to cover the T-Quiet period for the ALS's ADC. Once the conditions have been met, the FSM returns to S1.

B. Multiplexer

The Smart Night Light utilizes an 8-to-1 Multiplexer (MUX) allowing the user to select between eight display options. The select line is user controlled using three switches on the control board. The select options 0 through 3 utilize the ALS to control the intensity of the chosen color. The select options 4 through 7 do not use the ALS. When these options are selected, the given color is activated at full intensity. The table of MUX select options can be seen below in Table 1.

Table 1: Select line options for the 8-to-1 MUX.

SW ₂	SW ₁	SW ₀	sel	Output
0	0	0	0	Red w/ALS
0	0	1	1	Green w/ALS
0	1	0	2	Blue w/ALS
0	1	1	3	White w/ALS
1	0	0	4	Red w/o ALS
1	0	1	5	Green w/o ALS
1	1	0	6	Blue w/o ALS
1	1	1	7	White w/o ALS

C. Counters

The Smart Night Light utilizes (2) separate counters as internal components of the FSM.

The first counter is a 16-bit counter with enable and synchronous clear. The purpose of this counter is to counter the bits being output from the ALS. The ALS outputs data in 16 bit packets, 1-bit at a time. After this counter reaches the required count, a done signal is sent to a shift register enabling it to record the data.

The second counter is a 10-bit counter with enable and synchronous clear. This counter is used cover the "T-Quiet" period required from the Analog to Digital Converter (ADC) of the ALS. Once the count is reached, the machine return to State 1 and the cycle resets.

D. Registers

The Smart Night Light utilizes (8) separate registers (A-H) and (1) shift register with enable and synchronous clear contained within the FSM.

The registers indicated by A-D in the block diagram are used when the ALS is in operation. The 8-bits from the FSM are received and input into each register A-D. These 8-bits are used to determine the intensity of a selected color. The output of these registers is 24-bits. These 24-bits correspond to the three colors red, green, and blue; 8-bits for each color. In each application, for a single color, 16 of the 24-bits will be zero. The only application that requires more than 8 of the 24-bits to be active is when the color white is selected. The blocks E-H do not receive any data as inputs. These blocks are activated when the user selects to have a solid color displayed at full intensity. The blocks house the required code to activate the RGB LED at full intensity for a given color.

The shift register contained within the FSM is vital in filtering the data received from the ALS. This data is sent out in 16 sequential bits. Only 8 out the 16-bits are important for the applications of this project. The shift register essentially filters out the data needed and this is the overall output of the FSM.

E. Pulse Width Modulation (PWM)

The application of PWM was another topic that required the group to seek some guidance; being that it is not in the scope of this class. PWM uses 8-bit resolution to vary the brightness intensity on a scale of 0 to 255. The 24-bit output of MUX, depending on the selected option, was split into three different data packets of 8-bits. Each of the packets corresponded to a certain color. These packets would then be subject to PWM and the intensity of the output would vary accordingly.

III. EXPERIMENTAL SETUP

To verify the functionality of our project, we used the simulation tools in Vivado to test the compatibility of the ALS with our board. This was done by creating a test bench file to examine the communications between the chip and the Nexys board. In addition to simulating the circuit, implementing the circuit aided in testing and troubleshooting the project. The only software used in the project was Vivado. The hardware used to create the project is the Nexys board and a PMOD ALS chip which is a passive light sensor add-on to the board. The chip converts light levels to 8-bit digital data.

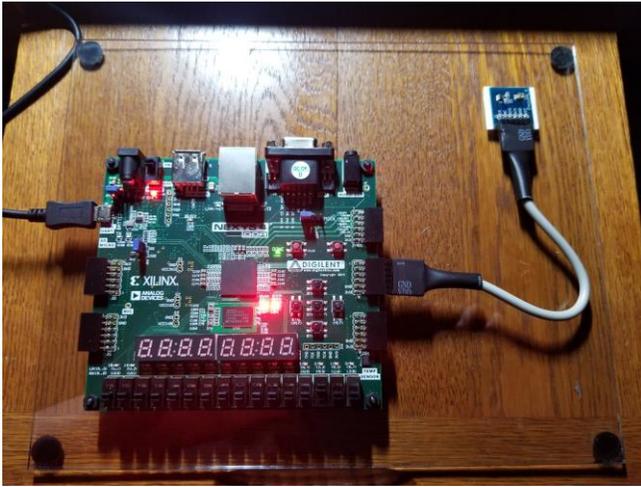


Figure 2: The Smart Night Light operating in auto mode with the red option selected.

IV. RESULTS

The waveform below (Appendix B) represents one cycle of the Ambient Light Sensor and the SPI with the board. The waveforms in teal are main inputs and outputs of the circuit. TBMISO is meant to emulate what the sensor may output as data, keeping in mind the bits that are automatically zero from the Analog to Digital Converter (ADC) embedded on the ALS. The waves in magenta are exclusively from the TBSCLK is meant to be the SPI clock, which should be a period of $1\mu\text{s}$ (1MHz). TBnCS is an output called chip select. The chip select bit is meant to go low when you want the ALS to start sending out data. The orange waves are for the two counters embedded in the circuit. The topmost counter counts to make sure that all bits have been received from the ALS. The bottommost counter makes the circuit wait a mandatory period of time (called the “quiet time” in the datasheet). It can be noted that one cycle of reading data takes place every 17.225us.

The resulting output of our project is a light that can cycle through different colors based on digital inputs. We

chose to challenge ourselves on this project by including an add-on that was not taught about in class, but the fundamental methods of analysis and design that are taught in class provided us with the skills needed to begin understanding this more complex concept. The ability to interpret the timing diagram of the sensor enabled us to write the code required to properly utilize the hardware. The obtained results were mostly as expected, because the work put into the circuit and the planning that went into the coding resulted in the overall functionality of the project. One result that was not entirely expected was the fact that the light remains on in a regularly lit room. This is the result of a relatively cheap and unprecise light sensor. The problem could be remedied by utilizing a higher quality chip, or if a filter or lens could be placed on the sensor that could make it sense dimmed lights more effectively.

CONCLUSIONS

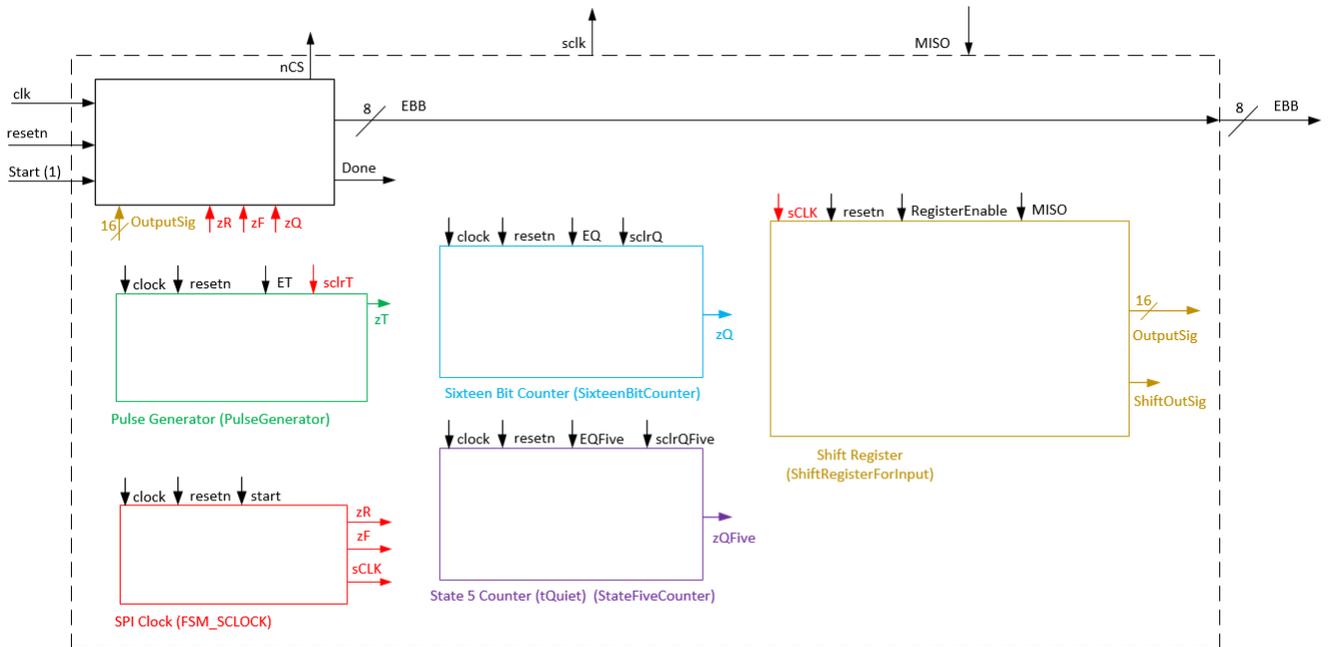
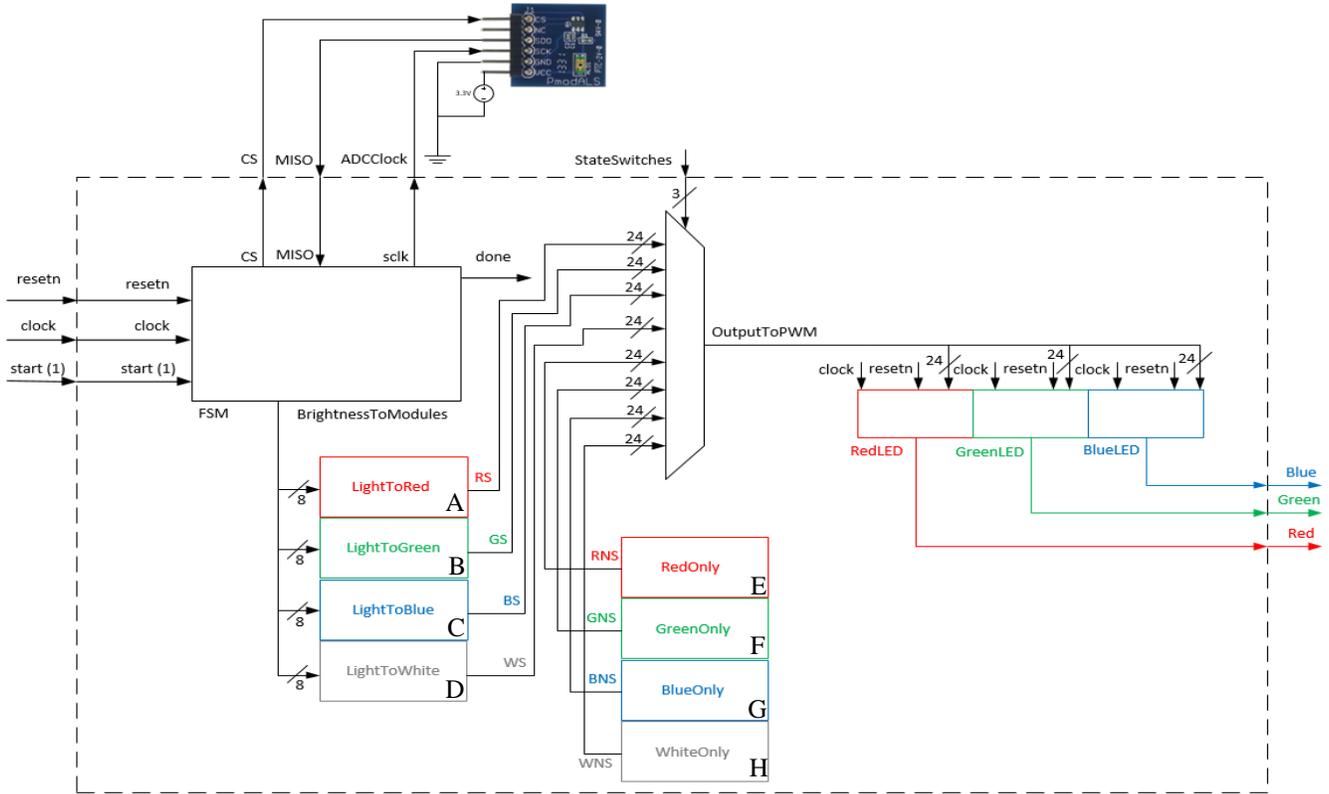
This project proved to be a useful application of what was learned about digital systems in this class. Basic components such as registers, multiplexers, counters, flip flops, and finite state machines were integrated to create a project with a useful application. Of course, this project was only a simple application of the principles learned in class. However, our work shows that the application of simple concepts can be expanded into something more realistic. One immediate improvement that could be made is the ability to manually adjust the brightness of the light. This would be in line with our stated goal of giving users more freedom and customization.

REFERENCES

- [1] *PmodALS™ Reference Manual*[PDF]. (2016, April 15). Diligent.
- [2] *ADC081S021 Single Channel, 50 to 200 ksps, 8-Bit A/D Converter datasheet (Rev. G)*[PDF]. (2005, April). Texas Instruments.
- [3] Llamocca, D. (n.d.). *VHDL Coding for FPGAs*. <http://www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html>.

APPENDIX

A. Block Diagrams for the Smart Night Light and FSM of the ALS.



B. Testbench simulation used to verify the functionality of the ALS FSM.

