MSP430 Colour Sensor

Team #34

Nikolaos Zapantis

Minying Lu

Boston University Spring 2015

EC 450 Microprocessors

Professor Giles

Goal

Use the MSP430 in order to build a color sensor with the ability to detect the color of an object and provide the color's hex number using a combination of the various aspects of a microprocessor that we have learned about throughout the semester. Including use of interrupts such as the watchdog timer and the adc interrupt, analog to digital signal conversion, the use of pins as both input and output, and some other topics that we have learned about.

Implementation

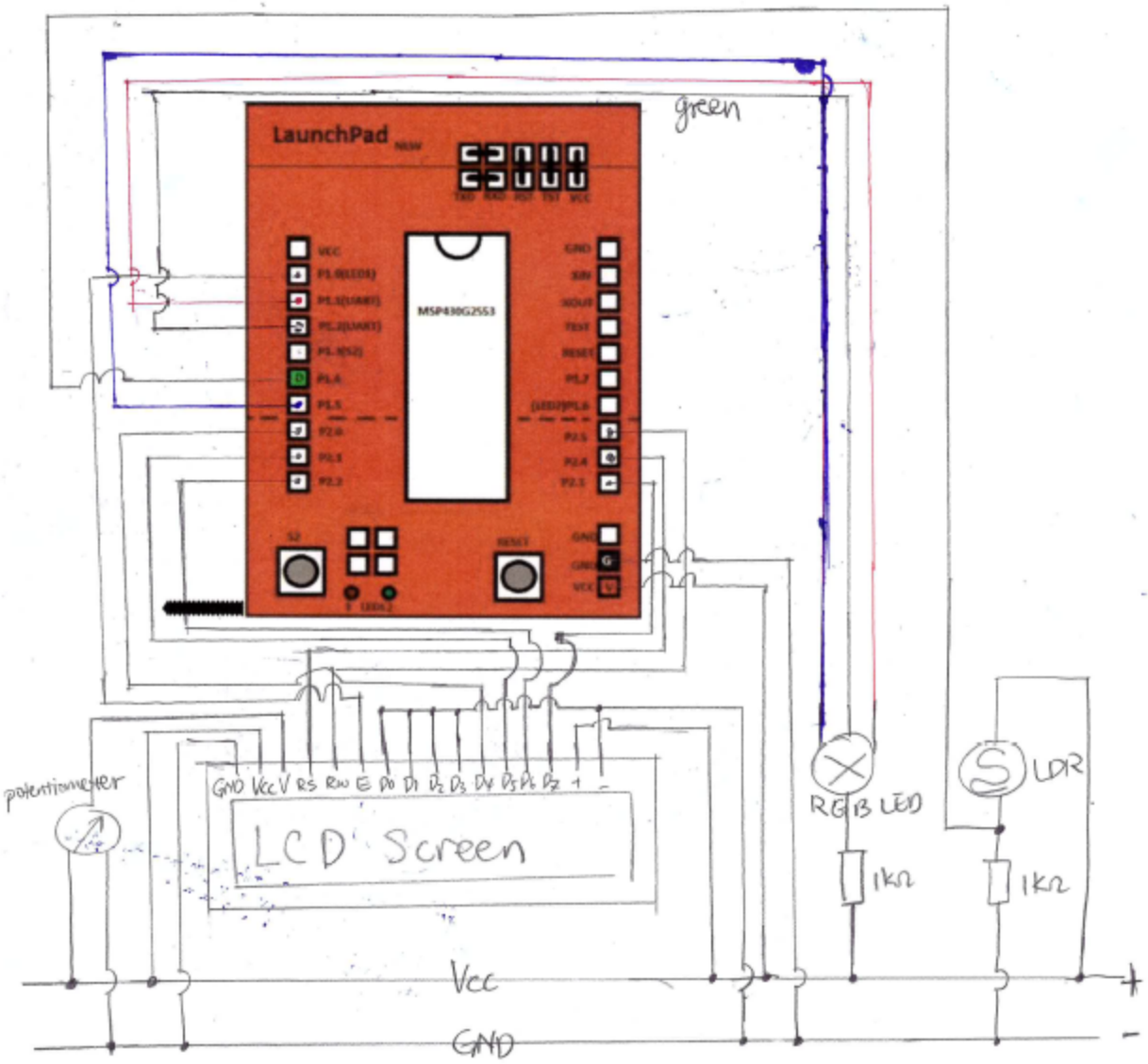
Our three main components were the LDR, the RGB LEDs and the LCD screen. The watchdog timer controls the blinking pattern of the RGB LEDs with them blinking red, green, and blue for 5 cycles. Each color activates a flag that will be used in the ADC interrupt, and each color is on for 100 interrupts while the clock set to 8MHz. The ADC interrupt reads the analog signal from the LDR and converts it to digital signal. In the ADC interupt, one reading is received per color per time, with a total of 5 readings for each color led that's blinked. For a more accurate and stable result we take the average of the five readings of red, green and blue storing the three averages into an array of the color. The calibration colors white and black are measured first and the corresponding readings will be stored in whiteArray[] and blackArray[], and the future readings of the color that will be measured will be stored in colorArray[]. In order to get number within the range of 0 to 255 (which correspond to hex values from 00 to ff, or 2 bytes), we set the blackArray readings to be the lower bound and whiteArray to be the upper bound, and applied a linear math equation on readings from colorArray. The behavior of the LCD is defined in lcd.c, and the lcd.h file refers to professor Giles' example from piazza. We

also used a button to control when we would begin the start the blinking of our RGB LEDs and the reading of LDR analog signal.

Project Assessment

As a whole the project was somewhat of a success, however there were a few issues that we encountered that ended up interfering with the accuracy of our readings. Our main issue when it came to getting accurate measurements for our readings was the amount of ambient light in the surrounding area. What the ambient light would end up doing to our readings was that it would end up basically as if it was reading a darker colour. For instance say if we wanted to measure red, the colour red would be at one corner while black would be at the opposing corner, so when we tried to measure our readings would end up being closer to black than red. However, when we tried testing it in a nearly pitch black area our readings ended up being considerably more accurate and were close to the colour we were trying to read. The reason of this is that there is not enough contraction between the base colors (black and white) and the color we are measuring because the ldr is always reading ambient light as part of the input. As a result, the reading of the color came out to be much lower than it suppose to be and the color end up in the darker corner of the color range.

Schematic


LaunchPad
MSP430G2553
green
potentiometer
GND Vcc V RS RW E D0 D1 D2 D3 D4 D5 D6 D7 + -
LCD Screen
RGB LED
LDR
1kΩ
1kΩ
Vcc
GND
+
-

Component List

- 1x MSP430 Launchpad
- 1x LCD Screen
- 1x Common Anode RGB LED
- 1x LDR
- 2x 1000Ω Resistors
- 1x potentiometer
- several male-female jumper wires

Next Steps

There are several things that we could implement if we wanted to improve upon our design, some of which would be fairly trivial to put in, while others would be considerably more complicated, One of the easiest additions that we could put in that would be a fairly large quality of life improvement would be to add a hard reset button. Since currently our application requires reprogramming from a computer in order to be able to recalibrate it, which is inconvenient if someone wanted to move it around since the measurements will change depending on various variables such as the height it is measured and the ambient light. So adding a reset button that would reset it to its default state would be an easy and helpful addition.

One of the harder things that would greatly improve our accuracy would be something that would help us take care of the issue of ambient light, there may be some sort of software solution where we would be able to take it into account, but that sort of addition is a bit more obscure on how to go about and would likely require a better understanding of the LDR. On the other hand there may be a sort of hardware solution that may be able to cut back on the ambient

light problem. One thing that we did try was covering the entire thing in a box, however the LED would reflect around the box and would still mess with our readings. An alternate solution that follows the same idea and may be a bit more effective would be to create some sort of individual encasing for the LDR that was shaped in such a way that it would cut out any surrounding light while only reading the reflection of the LED, while minimizing any ambient light that may be read.

There are also a few other things that we could attempt to do that would help improve the quality of our project. One thing would perhaps be a better math model when it comes to dealing with the way we calculate the colour array since the current way we do it is linear, however the LDR is more sensitive to green light so somehow taking that into account may help make our results more accurate. Another thing that would be an interesting addition, which I am unsure on how you would be able to go about it, would be to somehow have the MSP430 communicate with a computer. Since currently we just output the color values in hex on the LCD screen, a more interesting use of those values would be to send them to a computer which would then display the colour.

Code

See Github repository.

Summary of Contributions

Work was largely equally distributed between the two of us.