CS 240E Programming Assignment 1

April 2, 2014

1 Getting Started

The purpose of this assignment is to help you get started with setting up the Arduino Galileo and explore some of its basic features through a sample application. We will be using the Danger Shield provided in class. Before starting with the given problem please ensure you have set up the environment by following the tutorials mentioned below.

1.1 References

1. Intel: Getting Started
2. Setup plus Hello World
3. Danger Shield
4. Shift Register
5. Arduino Guide
6. Language Reference for Arduino

2 Secure Byte Lock

With the spate of break in's and robberies that have occured recently on campus you have decided to add an extra level of security and build a secure lock sytem of your own.

The lock requires a combination of 8 bits (Might we suggest you use *00101010*). Pressing *Button 1* and *Button 2* will consitute as having entered *0* and *1* respectively.

If the combination entered is incorrect then have the *Red LED* start to blink (at 1 second intervals) and the buzzer keeps on ringing until the lock is reset.Press *Button 3* to turn off the alarm and reset the lock.

Provided the combination entered was correct, turn on the *Yellow LED* and let it stay on till the lock is reset or locked again. Further, immediately

after the correct combination has been entered display a count down timer on the 7 segmement LED. Let the countdown begin from '5' and provide half a second between consecutive numbers. The user has to enter his/her house before the countdown hits zero. This is done by pressing *Button 1* and *Button 2* simultaneously, once this has occured the lock is reset to the locked state again. If however, you did not enter within the alotted time then you have to start the whole process again (i:e enter the combination again).

Leaving nothing to chance, the correct combination is not enough to gain entry. As a busy Stanford student, you leave home early in the morning and come back late in the night. Thus, using *Potentiometer 3* set the upper limit for acceptable light levels and if the light level recorded by the Photocell is higher than the threshold then deny entry. If entry is denied because of the light level then blink both LED's for 2 seconds before resetting the lock

2.1 Summary

1. Enter the combination to unlock using Button 1 and Button 2
2. If you entered a wrong combination then
   (a) Red LED starts to blink (1 second interval)
   (b) Buzzer keeps on ringing
   (c) Press Button 3 to disable alarm and return to locked state
3. If you entered the correct combination then
   (a) Yellow LED stays on
   (b) Immediately start a count down timer which is displayed on the 7 segment LED (Starts from 5, count down in half second intervals)
   (c) Press Button 1 and Button 2 simultaneously before the timer hits zero to 'enter'
   (d) If you enter the correct combination but fail to enter within the alloted time you have to start again
   (e) The correct combination is not enough to gain entry.Set the upper limit for acceptable light levels using Potentiometer 3. Thus, the lock only opens if ambient light levels are lower than the threshold.
   (f) Blink both LED's for 2 seconds if the user is denied entry because of the readings from the photodetector

3 Hints

1. Use the shift register to control the 7-segment LED. However, the built in shiftOut() function is not yet supported on the Galileo and you are expected to write your own implementation. Reading the forums and the datasheet for the shift register are a great place to start.

2. To ensure that the comparison between the readings obtained from the Potentometer and the photo-resitor are valid create a voltage divider for the photo-resistor as mentioned here. Be sure to read the comments at the end of the article.

4 Bonus Points

1. The Built in tone() function to control the buzzer is not yet supported on the Galileo. Thus, students who write their own version of the tone() function will receive upto 5% extra credit.