CPE355 - Real time embedded kernels - Spring'11
Prof. Nuno Alves (nalves@wne.edu), College of Engineering
Homework Assignment #8
Due date: 11am, Thursday, February 16th 2012

WESTERN NEW ENGLAND UNIVERSITY | WNE

- Homework turned in after the deadline, will not be graded.
- Submit you code to manhattan.
- Bring your question 2 code loaded in your Arduino to the Feb. 16th class for a live demo.
- Make sure your code is very clean and fully commented.

Question #1 (50 points)

The following piece of Arduino friendly code shows how to play a very annoying buzzer through a piezo speaker connected between digital pin 12 and ground.

```
void setup()
{
  pinMode(12, OUTPUT);
}
void loop()
{
  digitalWrite(12, HIGH);
  delay(500);
  digitalWrite(12, LOW);
  delay(2000);
}
```

I would like you to use this code as inspiration and create two different functions (call it taskA and taskB), that play two very distinct sounds for 3 seconds each. Ambitious students feel free to program more complex and longer melodies (e.g. Super Mario soundtrack or other retro sounds shown in http://www.superflashbros.net/as3sfxr/).

For this question, I would like you to enable both interrupts and create a very simple queue system with a fixed array of size 10. Every time interrupt0 is triggered (e.g. through a button), then the TaskA will be stored in the queue. Interrupt1 will have the same effect for TaskB.

Design constraints:

* TaskA and TaskB have the same priority. For example if I press 0,0,1 then taskA will be played twice and taskB will be played once after.

* While a buzzing sound is playing (either TaskA or TaskB), the queue through interrupts can still capture future buzzing sounds.
* You do not have to use pointers, function to pointers or dynamically allocate memory in this exercise.
* You may have to do some clever button de-bouncing, such that the same interrupt is not "accidentally" triggered multiple times.
* Every time the queue is updated, you will have to perform some general cleaning.
* If more than 10 function calls are queued, you may discard any additional function calls.

Question #2 (50 points)

Repeat the same exercise as question #1, with the difference that TaskA now has a higher priority than TaskB.