

**1.**

Use an active buzzer to make a one second beep when a tilt switch is tilted. Realize this by attaching the switch to an interrupt. (1p)

**2.**

Make an RGB LED cycle through red, green, blue, yellow, magenta, and cyan. Use an interrupt attached to a timer instead of the `delay()` function. (1p)

**3.**

- a) Use the passive buzzer to play nokia tune (or other monophonic melody of your choice). (1p)
- b) Add the keypad to the system so that you can use it to play notes. (1p)

**4.**

- a) Do a bitwise not on a positive integer. What is the resulting number? Explain how the microcontroller presents negative numbers on bit level. (1p)
- b) Why doesn't the following code work as one might expect? How would you make the comparison work with floating point numbers? Make the modifications to the code. (1p)

```
float f = pow(sqrt(2),2);  
Serial.println(f);  
if(f == 2.0){  
    Serial.println(" True");  
}else{  
    Serial.println("Why do we end up here?");  
}
```

There are many ways to achieve the described functionalities but in tasks 1 and 2 we want you to familiarize yourself with interrupts and timers. These enable faster response times and free processor resources in more demanding applications.