Assignment #4 – 40-Click Dash Game

In this assignment you will use the InterruptIn, Timer, and Serial interface on the mbed module to implement a virtual running game. (Depending on how you choose to implement this, you may also want to use Ticker and/or Timerout as well.) The two players will each have a pair of switches to control their running. Player 1 will use pushbutton switches connected between p30 and VOUT and between p25 and VOUT. Player 2 will use pushbutton switches connected between p11 and VOUT and between p16 and VOUT.

In this game, the players will compete to see who can race from the starting position to the finish line in the least amount of time. Each time a player presses the switch that was not most recently pressed, their position advances by 1 step (that is, the player needs to alternate between their pair of switches). Initially, your software should display some sort of message (via Serial to be displayed on your computer) to let the players prepare (for example, “On your marks”, “Get ready”, etc. with a short delay between each message). When the race begins, your software should display “Go!” At the point, the player should begin alternately clicking their switches as fast as possible and your software should use interrupts to count the steps. At least once per second, your software should display the position of the players. Minimally, this could be as simple as:

```
Player 1: 10 steps,     Player 2: 14 steps
```

or if you want to do something fancier, you could draw a mini racecourse with text:

```
S|..........1.............................|F
S|..............2.........................|F
```

(The “1” and “2” represent player 1 and 2 on a track that is 40 dots long from the starting line to the finish line.)

Whichever player reaches 40 steps first is the winner of the race. Once there is a winner, the mbed should display a message identifying the winner and the amount of elapsed time between the “Go!” and the 40th step. Show the elapsed time in seconds with at least 2 decimal places.

The serial interface should be configured for 9600 bits/second (“baud”), 8 bits, no parity, and 1 stop bit.

Submit your “main.cpp” to the appropriate dropbox on http://canvas.ou.edu by the end of April 2nd.