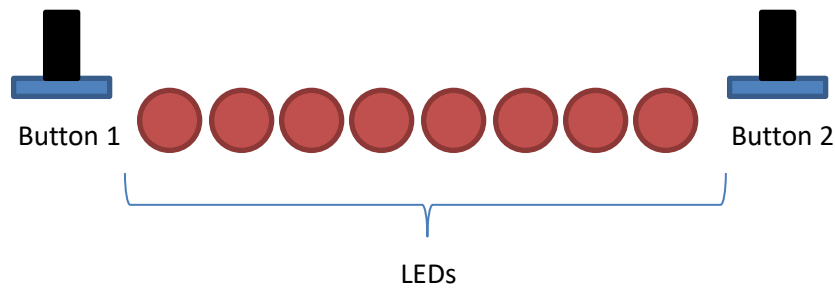


ME 220 – Introduction to Mechatronics

Arduino Assignment 02: GAME of LEDS

You need 8 LEDs and 2 switches/buttons that are properly interfaced to your Arduino (i.e. USE resistors for each component).

At the end you will have something as follows:



Player X will use Button X. Game scenarios are defined below.

When the program runs, it will display a choice menu on Serial Monitor and let the user choose game 1 or game 2. Details of this menu and response to invalid user choice are up to you.

Game 1:

In this game, it is about *sort of* shooting the other side. When a player pushes the button, LEDs will start to move (i.e. get lit) towards the other player. If other player can press the button before the LEDs reach to the other players end LEDs will move from that end as well and they will meet somewhere in the middle and cancel each other. In this case no one gets any points. If LEDs shot from one end reaches the other end before the other player shoots LEDs on his/her end (i.e. press the button), shooter gets 1 points. If a player shoots before an already in progress shot is finalized either by hitting the opponent or being cancelled by the upcoming shot, that players shot will be terminated immediately and that player will not be allowed to press again for a period of time where LEDs move for 5 positions.

Through serial monitor proper user feedback should be given for:

- a- Score changes
- b- Clashing shots cancelling each other – unprotected player warning
- c- Cancelled shots due to double firing

When a player reaches 5 points, that player is declared as winner and game ends and goes back to the main selection menu.

HINT: Remember, *polling* is when you check an input continuously and when there are interrupts you do not need to poll. Use INTERRUPTS! Recall how to [attach external interrupts to Arduino](http://arduino.cc/en/Reference/AttachInterrupt)¹. It is important to note that interrupt handling functions (a.k.a. interrupt service routines, i.e. ISRs) should be written as short as possible so that another interrupt does not happen while the previous interrupt is being handled! If there is such a probability, your handler function better [disable](http://arduino.cc/en/Reference/NoInterrupts)² interrupts when it starts execution and [re-enables](http://arduino.cc/en/Reference/Interrupts)³ them before exiting the handler function.

Game 2:

This is about timing. In one set, first one player will be prompted on the Serial Monitor to set target time. That is, after the prompt, indicated player will press the button, hold it down for a while, and then release. When the button is down, all LEDs will be turned on, and then when button is released, all LEDs will be off. Total time the LEDs are on will be measured as `time_target`. Then the other player will be prompted to press the button when LEDs are turned on. The second player is expected to keep the button down (if possible) exactly as much as the first player did! As soon as the second player presses the button, time will start to tic and as soon as the button is released, this interval will be measured as `time_guess`.

Again the lights will be off as the second player release the button.

If the second player manages to be within +/- 10% of the time set by the first player, player 2 gets the set, otherwise player 1 will get the set.

At the end of each set results including `time_target`, `time_guess`, winner and current points will be displayed on the Serial Monitor.

In the next set, roles will change, setter will become guesser etc. Whoever reaches 5 first, will be declared as the winner and goes back to the main selection menu.

¹ <http://arduino.cc/en/Reference/AttachInterrupt>

² <http://arduino.cc/en/Reference/NoInterrupts>

³ <http://arduino.cc/en/Reference/Interrupts>