

# Project proposal: Motion-deactivating my alarm system

Long-Hin Tang\*  
(Dated: April 29, 2021)

The goal of this project is to build an alarm that is deactivated by the completion of a motion-based task, where the user is required to move back and forth from the alarm multiple times under a time constraint. The exact parameters of the task can be set by the user. The separation between the user and the alarm is determined by an ultrasonic distance sensor.

## I. DESIGN CONCEPT

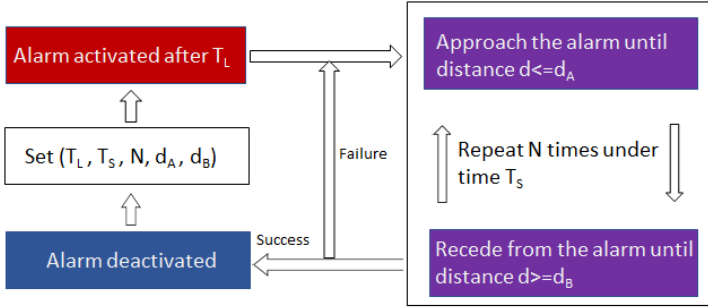


FIG. 1. A flow diagram of the device's operation.

- The alarm is activated after a countdown time  $T_L$  (up to a maximum of 24 hours), after which the user is prompted to complete the motion-based task.
- In the task, the user is required to move across a point at a distance  $d \leq d_A$  (down to a minimum of 2 cm) and one that is  $d \geq d_B$  away (up to a maximum of 4 m) for  $N$  times under time  $T_S$  (up to a maximum of 59 seconds). These limits ( $\pm 3$  mm) for  $d_A$  and  $d_B$  are based on the specifications of the ultrasonic sensor. See FIG. 2 for an illustration of the geometry. The timer for the task is initiated by the user's approaching the alarm down to  $d \leq d_A$ . The user is prompted to retry up to 100 times or until the task is completed under the time limit  $T_S$ .
- Upon the completion of the task, the alarm is deactivated and enters a setting phase, where the user can set the parameters ( $T_L, T_S, N, d_A, d_B$ ) with a keypad switch and initiate the countdown from  $T_L$ .

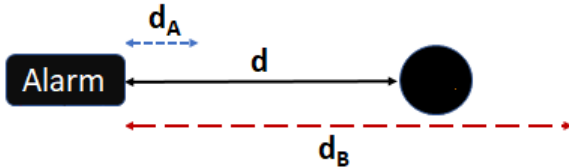


FIG. 2. An illustration of the configuration of the alarm and the user (represented by a black disk) during the motion-based task.

- A LCD is used to prompt the user to follow the instructions and show the times and distances during the different phases. For example, it shows the countdown time from  $T_L$  when the alarm is inactive.

## II. COMPONENTS REQUIRED

Every item on the list is contained in the Arduino kit.

- Arduino UNO R3 microcontroller
- Breadboard
- Active piezoelectric buzzer
- Resistors/ potentiometer
- LCD (LCD1602) (Data sheet.)
- Ultrasonic sensor (HC-SR04) (Data sheet.)
- Keypad switch

## III. SCHEMATICS

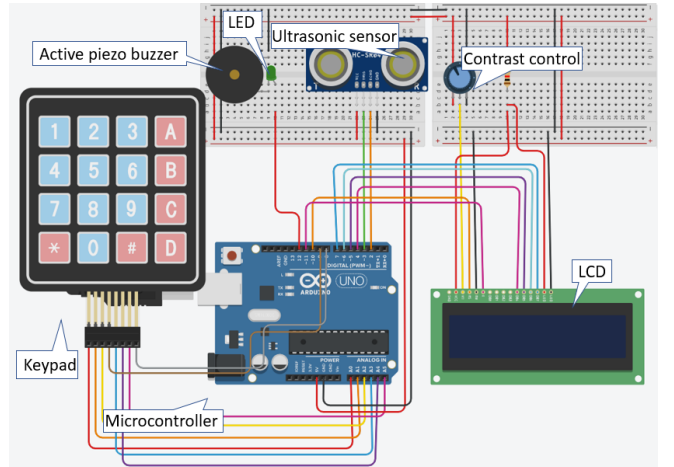


FIG. 3. The electronic schematic diagram of the device.

## IV. PLAN OF IMPLEMENTATION

1. Connect an active buzzer and the LCD to the Arduino such that the buzzer is activated after a user-set countdown time. At this stage, the alarm automatically enters reset mode after a certain duration.

\* Physics Department, Boston University.

2. Test the ultrasonic sensor by plotting the logged data. In particular, test the range and precision of the sensor and check that they are consistent with the manufacturing specifications.
3. Build the motion-based mechanism by using the distances as a trigger for the deactivation.
4. Addition of a keypad switch which enables the user to set the parameters ( $T_L, T_S, N, d_A, d_B$ ) without modifying the code.

## V. IMPROVEMENTS

1. Reducing noise and correcting discontinuities in the distance readout with the application of low-pass fil-

ters and data processing.

2. Powering the device with a battery pack and a power module so that the device can function independently of a computer. The device can then be housed in a protective case.
3. The addition of a Real Time Clock (RTC) module allows the device to function as a regular clock that displays the current date and time. It also gives the user the option to trigger the alarm based on a set local time, rather than just a countdown timer.
4. Connecting the microcontroller to an IR receiver module allows the user to set the parameters using a remote control, instead of the keypad switch. With this addition, the device can function remotely, preventing the user from deactivating the alarm with brute force.