Alexandria University
Faculty of Engineering
Computer and Systems Engineering Department

CSE343 Embedded Systems
Assigned: Wednesday 19/04/2023
Due: Friday 28/04/2023

Lab Experiment 06-A

Objectives

Implement an experiment based on state machines, external interrupts, servo motors, buzzer, and potentiometers.

Problem Statement

You are required to implement a system to control the charging and discharging of a reservoir. The Charging and discharging, are controlled by setting the position of the servo angle to control the action (charging, discharging, idle and the rate). The operator has two potentiometers to change the rates, one for charging, the other for discharging. The operator has a push button to switch from charging, discharging and idle. There is an RGB led to indicate current action. The system checks the limits (fully charged, completely discharged), and switch to idle, and activate a buzzer alarm.

Implementation Details

The system behavior can be modeled according to the following state transition table:

Butt.	action old	quantity new	action new	led	servo	buzzer
yes	charge	x	idle	B	idle	no
no	charge	>=max	idle	B	idle	yes
no	charge	<max	charge	G	new rate	no
yes	discharge	x	idle	B	idle	no
no	discharge	<=min	idle	B	idle	yes
no	discharge	>min	discharge	R	new rate	no
yes	idle	>=max	discharge	R	new rate	no
yes	idle	<=min	charge	G	new rate	no
yes	idle	>min&<max	discharge	R	new rate	no
no	idle	x	idle	B	idle	no

(x)→ Don't Care

Quantity is updated according to the following function:

Quantity new = quantity old + old rate * interval (will be defined below).

Button Behavior

Button = true (1) if button interrupt occurred, otherwise false

Use interrupt handler function, assign external interrupt

The handler must first mask the external interrupt (**EIMSK &=0x00**)

(It should be restored later in the loop function)

Then it must clear any pending external interrupt (**EIFR |=0x03)**

Charge/Discharge Rates

Charge/discharge rate = potentiometer reading (normalized to 1.0)* maximum rate

New rate (operating rate):

If new action is charge rate = rate,

If new action is discharge = - *rate*,

If action is idle rate is 0.

Use two functions to: Read the two potentiometers and calculate the charge/discharge rate.

Use a function to update rates

LED

Use a function to update led.

Buzzer

Use a function to activate buzzer.

Report on Serial Monitor

report the status(every interval on the serial monitor, activating time stamp)

the report , may look for example as follows:

Action	Charging
Rate	5.2/minute
Quantity	500.4
Percentage	20.5%

Remaining time 70 minute

Use a function to display the report

```
If charging: Remaining time = (max - quantity)/rate.

If discharging: Remaining time = (min - quantity)/rate.
```

Data

```
Reservoir capacity is 12000 units.

Max charge rate 200 unit/minute

Max discharge rate 40 unit/minute

Interval is given as 60 seconds

Servo settings:

      charging angle: (90-5)*new rate/maximum rate.

      Discharging angle: 180+(90-5)*new rate,  idle angle 90.

      Quantity limits: max = 0.95* capacity,  min=0.05*capacity.
```

Suggested Program Layout

Global Section

- Use named constants for initial data.
- Includes of libraries, constants, variables, pins (one switch, one servo, three led, two pots, one buzzer), button must be volatile.
- Function decalarations.

Setup Section

- Assign interrupt, set pins input/pullup/output, initialize old values.
  Action is set to idle, quantity is set to zero, charge rate is set to zero, button is set to false.

Loop

Update quantity, update rates, set servo, update display, delay interval, reset button to false, (set all old values to new values to be ready for the next cycle), and unmask external interrupts as follows:

```
cli() ;
EIMSK |=0x03;
sei();
```

For cli() and sei() documentation check this link in arduino forum:

- https://forum.arduino.cc/t/solved-cli-soi-where-are-these-functions/669135

Resources:

- Interfacing buzzer with tone:
  - Interfacing buzzer diagram
    https://www.instructables.com/Interfacing-Buzzer-to-Arduino/
  - Tone function
    https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/

- Sample program https://drive.google.com/file/d/161i8o6N8dsUqQQkXqJwbGVjYDauukCq9/view?usp=sharing

Interfacing RGB:

- **Note**: The given RGB is common anode. See a sample interfacing diagram through this link and then map the interfacing to the next link. https://circuitdigest.com/microcontroller-projects/single-rgb-led-interfacing-with-arduino
- Sample program: https://drive.google.com/file/d/1Y1WAa6uihhf51m1G7iFtQlQpSCsXoMqI/view?usp=sharing
- Common anode RGB specifications: https://drive.google.com/file/d/1B9QJl5CN44gJ6ezO2EBfL9TEcs6WnsVX/view?usp=sharing

- Interfacing Servo motor:
  - Datasheet: https://components101.com/motors/servo-motor-basics-pinout-datasheet
  - Sample program: https://drive.google.com/file/d/1vvUsJu5K4d723LLx8c0NwdhsiRnoYHsG/view?usp=sharing

Delivery Policy

- Each group must send a 20-second video for the system at rest showing charging and discharging with all indicator outputs shown.
- You should submit a report showing your schematic diagram and the challenges you faced (if any).
- You should submit the sketch source code (.ino file(s)).
- You should cite any additional resources you used.
- Further details for the submission instructions will be posted later on MS Teams.

Good Luck