Case Study of Mechatronic System Design: Automatic Padlock Opener Control (APOC)

1. Learning outcomes:

- Digital I/O processing.
- Internal and external interrupts.
- Free Time System.
- Feedback motor control (P-control/PI-control).

2. Introduction

This project is to design a mechatronic system to open a padlock automatically/manually (dial lock) by controlling a DC motor coupled to the knob of the lock. One of the padlocks (dial lock) is shown in Figure 1. To solve the padlock from the lock state, we must follow the manual on the lock (while we are facing the lock):

- 2 complete turns RIGHT (CW) to the 1st number.
- 1 complete turn LEFT (CCW) to the 2nd number.
- Turn RIGHT (CW) to the 3rd number.

The three numbers in the steps are preset for each padlock, but the rotation patterns are the same for all these locks. The rotation actions are executed by implementing a position control of a DC motor with an incremental encoder. After the three rotations, the shackle can be pulled out manually.

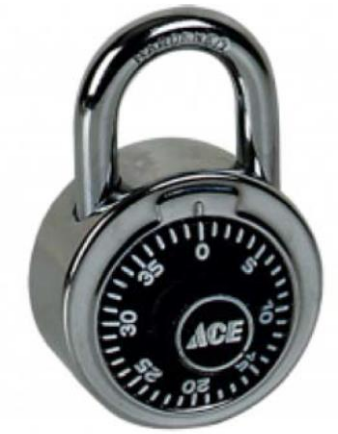

Figure 1. ACE Padlock

In this project, we cannot build the whole system of Automatic Padlock Opener Control given the limited materials, but we will implement the motion control and the user interface of our system based on the contents we learned in this course. The small demo system we are going to build is shown in Figure 2.

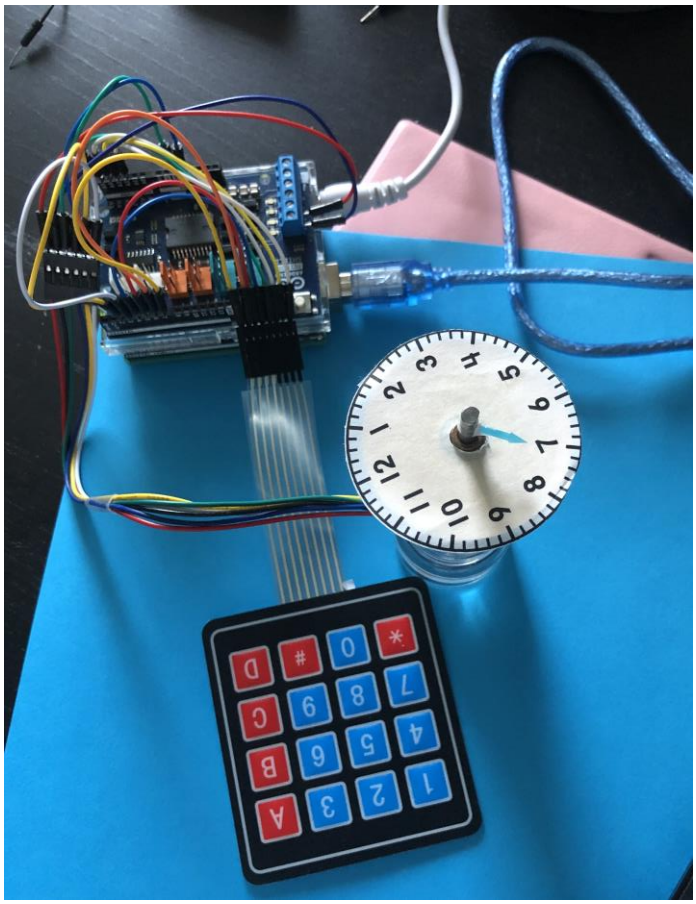

Figure 2. System of the Padlock Opener

A paper plate clock is attached to the motor to mimic the scales on the padlock. A tape pointer attached to the shaft of the motor is to represent the arrow mark on the padlock. As the incremental encoder does not give us the absolute position, we must know the initial pointing position of the shaft by inputting the initial pointing position to the system. Given that, we can know the absolute position of the shaft based on the incremental position read from the encoder. In the automatic mode, the user needs to input three numbers from the Padlock manual to solve the lock and the motor will execute the three rotation steps by itself. In the manual mode, the motor will rotate one tick in CW/CCW direction after the user push a key on the pad.

3. Building Materials

We build the FTS following the structure in Lab 4. The KeyPad is used for a user to input the information, like initial position, numbers on the manual, or menu selection. The motor in the kit cannot be used directly in this project, so we must buy the extra motor and driving board online. The items used in this project are listed below:

- Arduino UNO (in the kit);
- DC +9V power (in the kit);

- KeyPad (in the kit);
- Arduino Motor Shield L298 ($29.95, extra order);
- DC motor with incremental encoder ($16.99, extra order);

4. Circuits Design

In this project, the Arduino motor shield is used to drive the DC motor. Arduino motor shield can control two motors simultaneously. When we use Part A, the direction is controlled by D12; Enable Pin (PWM signal) is on D3. The brake is engaged/disengaged from D9. When we use Part B, the direction pin is on D13. The Enable pin is on D11. The brake pin is on D8. As we must use D2 and D3 as our external interrupts, we must use Part B to control the motor in this project. The pins are used in this project is listed below:

- Motor control pins fixed from the Shied:
  - Direction: D13.
  - Motor Enable (PWM): D11;
  - Motor Brake: D8.
- Keypad 1~8 is respectively connected to the Pins: D7, D6, D5, D4, A5, A4, A3, and A2;
- Motor pins are connected following the color-coding:
  - Red (Motor +) -- B+ on the shield.
  - Black (Motor -) -- B- on the shield.
  - Green (GND for encoder) -- GND on the shield.
  - Blue (Vcc for encoder) -- 5V on the shield.
  - Yellow (Channel A) -- D2.
  - White (Channel B) -- D3.

Pin D2 and D3 are used as external interrupts for detecting the encoder signals (rectangle wave). In this project, all the user interface information will be displayed on the Serial Monitor of Arduino IDE.

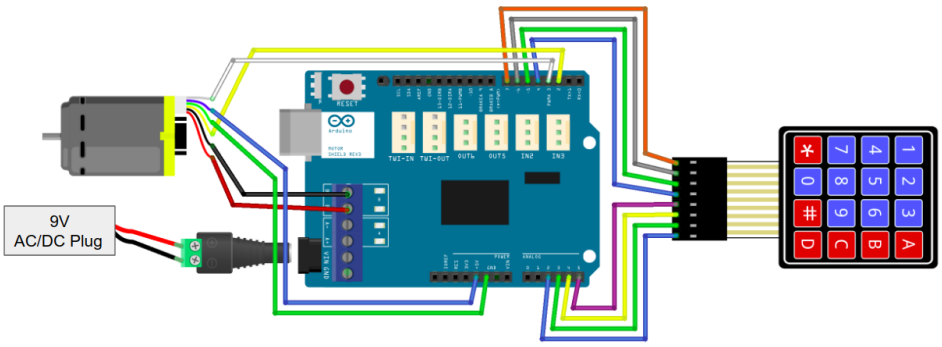

Figure 3. The circuit of the Padlock Opener Motor Control System

5. Motor and encoder testing

After building the circuit, we need to check the motor and encoder in the system. You can follow the instruction pages on the Canvas: DC Motor with incremental encoder.

6. P control tuning

After basically checking the motor and encoders. Let's tune the P value by following the instruction pages on the Canvas: Position control of DC motor.

7. Free Time System Design

In this part, we had an explanation on the requirements for the system we must use on each module. You can add more modules depending on your system design.

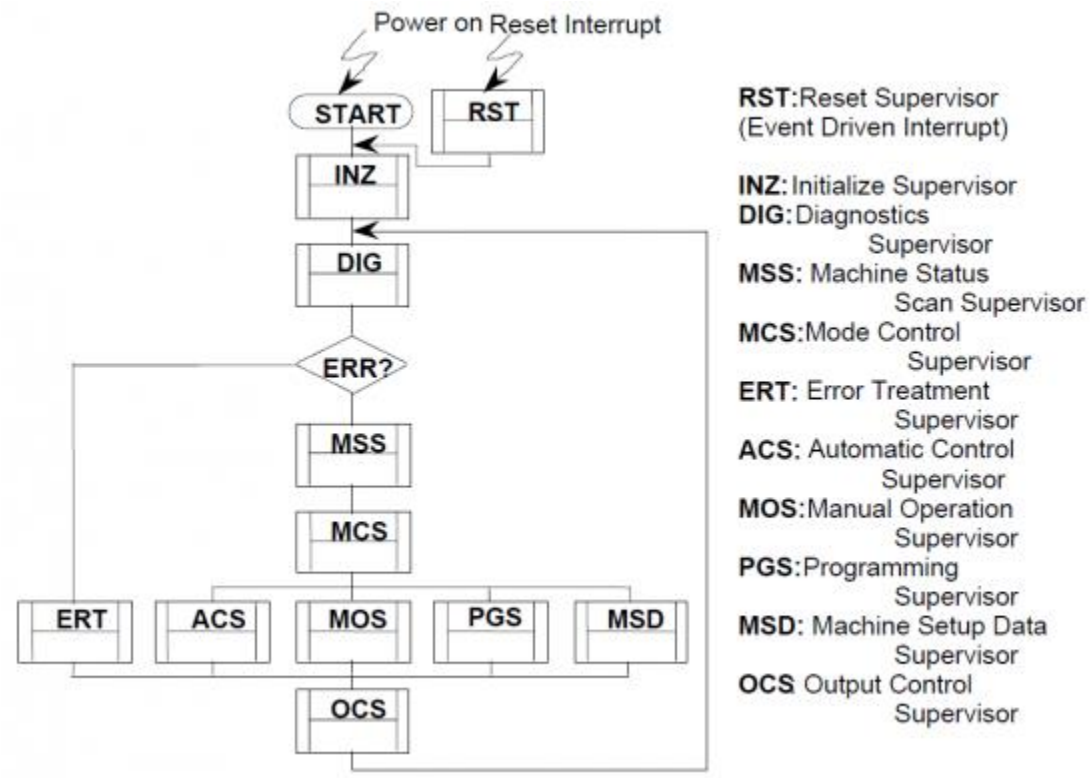

Figure 4. General Free Time System Flow Chart

A. Initialization Function (INZ):

In the initialization function, we may set up the system initialized flags including system first running flag and error detection flags.

B. Machine status selection function (MSS):

Given the user input from Keypad, **t**he machine goes into the relevant mode.

C. Machine setup data function

- **Input**: Initial dial position. The input should be given through the Keypad.
- **Output**: Display of the dial position on the Serial Monitor.
- **Process**: The control should be set to "Ready" status for Automatic or Manual Operation.

D. Automatic function (not activated unless control status is "Ready")

- **Input**: Three dial numbers to open the lock. The input should be given through the Keypad.

- **Output**: Motor rotation to position the dial to the input positions and shackle pulling prompt from the Serial monitor. Also, it displays which combination code number is moving to on the Serial monitor.
- **Process**: When the three dial numbers are given after the initial dial position is input during the automatic function, the motor should be rotated in the following manner:
  - 2 complete turns RIGHT to the 1st number
  - 1 complete turn LEFT past the 1st number to the 2nd number
  - Turn RIGHT to the 3rd number.
  - When the actions defined above are completed, the output prompt for pulling the padlock should be shown.

E. Manual Function (not activated unless control status is "Ready") :

- **Input**: The keys on the Keypad designated for the CW or CCW rotation of the dial knob.
- **Output**: Motor rotation in the specified direction.
- **Process**: When the designated key is being pressed, the motor should be rotated in one tick in certain direction.

F. Programming Function: (Not needed).

G. Output Control Supervisor:

- **Input**: State/Mode of the system.
- **Output**: Information displayed on the Serial Monitor.
- **Process**: You need to design a readable information on the monitor relevant to the user's process on the system. The information should not be too long and repeated in each state.

H. Error Function:

- **Input**: Detection of the error
- **Output**: Error message on the display
- **Process**: Any systems error should be detected when it occurs. Any man-machine communication error should be detected (i.e. unusable keyboard input).

8. Tasks for students

A. Build the circuit given the drawings we provided.
B. Evaluate the motor and encoder based on the contents in Lab 5/Lab 6 and showing how many pulses you can get with your detection method.
C. We provide the interpolated P control for the motor and the student need to try different numbers of P to get the motor response curves. They can print the encoder position into the Serial monitor and use Excel to plot the positions. After tuning P well, update the parameter into the code.
D. Finish the code of incomplete FTS to implement the whole designed functionality.
E. Draw the general flowchart of the FTS (Free Time System) of APOC by adding/deleting the modules to/from the comprehensive one that was explained in Lab 4.
F. Explain the types of interrupts they used in the system.
G. Draw the flow chart for the **user manual** in different operation modes.
H. ***Extra points: PI control implementation in the system***

9. Presentation Video (15 ~20 mins)

Students need to make a slide to show the contents below:

A. Introduction of the system modules.
B. Motor control:
    a. Evaluation of motor and encoder.
    b. P-control tuning process of the motor.
C. Human-machine interface design in each step of the flow chart.
D. User manual to process the system.
E. Developing process of the whole system.
F. Video to show the demo operation in different modes.

10. Final folder to submit:

- Built Circuit.
- Developing Code.
- Slides used in the presentation.
- Presentation Videos.