

Chapter 1

User manual

1.1 General

There are two folder in the source file folder: "arduino" and "python-gui".

Arduino folder:

This contains Arduino source code.

main.ino: Containing state-machine and interrupt callbacfunction for the servo encoder.

dac.cpp and dac.h: Driver for dac used on stepper motor.

motor.h: Contaning elevator (servor) class and stepper (door) class.

led.h: Containing the LED class for updating onboard LED on All-In-One servo-lab to indicate which floor the elevator is.

Python GUI:

This contains GUI source code.

1.2 Arduino

Open this project in Arduino IDE. Select board, serial bound-rate and program the Arduino.

Commands to run from serial monitor

Inside cabin:

- **Go to 1. floor:** 11 (Ctrl + Enter)
- **Go to 2. floor:** 22 (Ctrl + Enter)
- **Go to 3. floor:** 33 (Ctrl + Enter)
- **Emergency button:** 911 (Ctrl + Enter)
- **Reset alarm:** 999 (Ctrl + Enter)

Outside cabin:

- **At 1. floor, order elevator here:** 1 (Ctrl + Enter)
- **At 2. floor going down, order elevator here:** 21 (Ctrl + Enter)
- **At 2. floor gOUNg up, order elevator here:** 23 (Ctrl + Enter)
- **At 3. floor, order elevator here:** 3 (Ctrl + Enter)

1.3 Use the elevator with the Python GUI

Open project in PyCharm. Install "pyserial" and "tkinter".
Edit the serial port in "main.py":

```
6
7 arduino = serial.Serial(port='COM3', baudrate=9600, timeout=.1)
8 # windows: port="COM4" or port="COM3"
9 # linux: port="/dev/ttyACM0"
```

Exit the serial monitor in Arduino IDE (this prevents others using the port) and run the Python script.

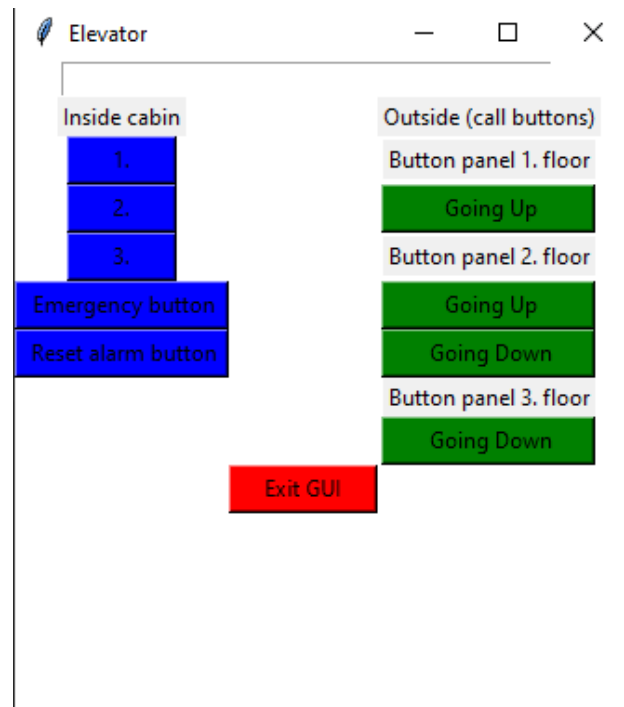


Figure 1.1: Python GUI

Figure 1.1 shows the GUI. The blue buttons are inside the elevator and the green buttons are outside the elevator.

NB! The preferred method to exit the GUI is to press the red button "Exit GUI". This button ensures that the serial port is closed after exiting.