In this project, you will develop and deploy embedded software to control a small elevator system. The tasks include to develop a model for a controlled process (a DC motor), to develop driver blocks for an actuator and sensor, to design a control algorithm to control the elevator, to generate and deploy code for an embedded processor, and to tune your design. Designing the Driver Block for the distance sensor is also the first Problem of Assignment 3, so it is recommended you start with that.

This assignment is to be solved by groups of two students. Each group should hand in their solution. If there are difficulties to form groups, get back to us, and we will try to resolve this.

The elevator system has been built physically, starting in August 2016 by Derrick Alabi and Christoff Ellmer. In its basic structure, the elevator consists of an elevator cabin that is hanging on a wire in its shaft. The wire is controlled by a 12V DC motor. At the bottom of the shaft is a distance sensor. The distance sensor and the 12V DC motor are connected to a controller, realized by an STM32 system (in the original design, this was an Arduino Uno processor mounted on a board). The board gets its power from a power-supply, on which the voltage that is supplied to the board can be controlled. You should set this voltage to 0 volts before connecting it to the board. Thereafter you turn it up to 12 volts.

The voltage from the power supply must NEVER exceed 12V. This may damage the board. Please be careful with the voltage.

The control unit (i.e., the STM32 board) has five buttons (four, numbered 1, 2, 3, 4, correspond to four floors, and the fifth is an emergency stop button) that can be pressed by users (for simplicity, we do not distinguish between buttons at floors and buttons inside the elevator, there is just one button for each floor). Whenever a button for floor $n$ has been pressed, the elevator must eventually (possibly after passing and stopping at some other floors) arrive to floor $n$, and the doors open. The elevator then stays at this floor until the control unit orders it to move to some other floor. Once the doors are open, they remain open for at least 5 seconds. The elevator must close the doors before leaving a floor. There is a red lamp, intended to represent that the emergency stop has been pressed, and a yellow lamp, which is intended to go on when “doors are opened”. The board controls
a DC motor, by a pulse-width modulated signal. The width of the signal corresponds to
the torque exerted by the motor on the wire, which makes the elevator itself go up and
down. The ultrasound distance sensor at the bottom of the shaft measures the height of
the elevator carriage. On the shaft, there are markings that are intended to correspond
to the four floors.

Your task is to model the elevator system, including the elevator, motor, sensor, and
controller in Simulink/Stateflow, and generate code that runs on the STM32. The same
model should also be able to simulate the behavior of the elevator system in pure simulator
mode inside Simulink.

The requirements are quite similar to those for the last problem in Homework 2. The
intention is that you can reuse most of your design for that problem (from one of the team
members). They include

• Typical “elevator-behavior” requirements, including that: whenever the button for
  floor $n$ has been pressed, the elevator must eventually arrive to floor $n$. When this
  happens, the yellow lamp is lighted, signalling that the doors are open. Once the
doors are open, they remain open for at least 5 seconds. The doors must close (i.e.,
the yellow lamp goes off) before leaving any floor. Whenever the “stop” button is
pressed, the elevator is stopped and the red lamp is lit. The lamp is switched off
and the elevator resumes to normal when the “stop” button is pressed again.

• The elevator should run smoothly, and make smooth stops and starts at floors. It
  should thus exhibit a suitable maximum speed and a suitable maximum absolute
  value of acceleration.

• The elevator should stop only at the precise locations of the floors, as marked on the
  shaft. You should find out how these locations correspond to distances, as measured
  by the distance sensor.

• The elevator should be robust to faults/disturbances, at least in the following sense:
  if the distance sensor provides clearly wrong data (e.g., if an object enters under
  the elevator cabin), then the elevator should detect that “something is wrong”, and
  perform suitable action (maybe this could be just stopping, and/or possibly lighting
  up some lamp, etc. Feel free to design this in a suitable way).

You can reuse your floor-planning algorithm from Assignment 2, but if its performance
is not good, you are encouraged to optimize it (this has lower priority than making the
elevator functional).

Some of the subtasks that are involved include:

• You should make yourself familiar with the setup, construction, etc. of the elevator
  (e.g., which pins correspond to which hardware features). Some of this is described
  in the document assemble.pdf.

• Developing Simulink Driver Blocks that allow Simulink models to receive signals
  from the distance sensor and send signals to the Motor. This task is included in
  Homework 3, so you can reuse a team member’s solution to this problem.
• Making a Simulink model of the correspondence between the input to the DC motor and the speed of the elevator (as seen by the sensor). This correspondence includes, e.g., a dead zone around 0, where the torque from the motor is too weak to move the elevator, and how increasing the input signal is translated into greater speed. Note that a given voltage produces more speed in the downwards direction than in the upwards direction.

What you should hand in  You should produce a report (in .pdf), where you describe your overall design effort (what design decisions where applied to the various parts), include the models, driver blocks, the procedure you used for producing a model of the Motor-elevator behavior.

You should also present the outcome of a usage scenario with a given sequence of button-pushes by users. This should be done in two ways:

• You first demonstrate a usage scenario on the physical elevator, and log and save some input and output signals.

• Thereafter you do the same scenario in pure simulated mode: apply the logged input signals onto your Simulink model of the system, and record the outputs.

You should compare the logs (mainly the movement of the elevator) and check that the simulated behavior is close to the logged “real” behavior, also with respect to timing.

To produce a model that can be used both for code generation and simulation, the block Environment Controller (see Simulink documentation) can choose to forward either of two signals, depending on whether code is generated or the model is simulated. You may consider whether you want to use it (using it is not at all required).

Acknowledgment: Thanks to Derrick Alabi and Christoff Ellmer for assembling the elevator and producing the first versions of documentation. Thanks to Magnus Lång for producing updated documentation for STM32.

Submission

Solutions (all files) to this assignment are to be submitted via the Student Portal by October 26, 2018. You should also prepare a demonstration for Magnus and me (times decided later)