Exercise work – Alarm system

In the exercise work the task is to make an alarm system using both Arduino UNO (ATmega328p) and Arduino MEGA 2560 (ATmega2560). Either the UNO or the MEGA should use the motion sensor to sense movement and the keypad module for the user to input the password. Connect the UNO and MEGA using a communication protocol (**USART, I2C/TWI or SPI**). The other microcontroller is used to display the alarm system state (communication connection, movement detection, password given, incorrect password, no password given fast enough, etc.). Add a buzzer to sound the alarm when needed. The exercise work must fulfill at least all of the minimum requirements. The exercise work will be done using the Atmel Studio environment. Libraries and header files etc. that have been used or provided during the course are all allowed. You can find a keypad library on Moodle that you can use.

However, the use of Wiring language is not allowed (*digitalWrite()*, *analogRead()*, *pinMode()* etc.).

Use of the following libraries/header files is not allowed:

- Wire.h
- SPI.h
- Serial

The minimum requirement for the exercise work are:

- Motion detection, which causes an alarm
- buzzer for the alarm (PWM)
- **USART, I2C/TWI or SPI** communication between the MEGA and UNO, where one micro-controller is used to control the sensors and the other is used to give system information for the user (one is the master device and the other is the slave device)
- user must give a four (4) number password via a keypad
- "submit password" – button
- a backspace button
- timeout for the password input
- the alarm system can identify whether a password is given and if it's correct or incorrect
- the user is informed whether the given password is correct, incorrect or a timeout has occurred
- upon receiving the correct password, the alarm system is turned OFF
- use of **state machine structure** (one or both)

Additional functionalities for the exercise work:

- Information is displayed to the user via LCD
- the system check whether the communication connection can be found
- use of interrupts
- possibility to arm/ rearm the alarm system
- use of sleep modes when possible (one or both)
- use of ADC
- using EEPROM to store and change the password
- adding your own justified extra feature

Additional grading criteria:

- use of subroutines (if needed / possible)
- user-friendly "UI"
- use of Embedded C coding standard

The groups will have to demo your circuit to exercise instructor before deadline. After the demo, return the commented whole Atmel Studio projects (*main.c* and other used .c or .h files), circuit diagram and brief report of around 5 pages, in which you explain what did you do and why, what was good or what could be improved in the program software, etc. **THE DEADLINE FOR THE DEMO AND REPORT IS FRIDAY 14.5 AT 23:59.** You can demo your work either in the exercises or by arranging a demo time with the instructor via email or by submitting a demo video. Return the material to the instructor via Moodle. For demo video instructions see below. Include your group number (number of the Elegoo box) or group name as well as the names and student numbers of the participants in the report. If you have problems or questions you can contact the instructor via email at lauri.jarvinen@lut.fi or alejandro.ibanez.rioja@lut.fi

Exercise work is graded with a 0-50 point scale. The grading includes your program (how it works, usage of variables and functions, etc.), the comments of your code, the circuit for your project and the report. More information on the grading can be found in Moodle, under "Exercise Work".

Demo instructions for distance students

The video of your exercise work demo should contain the following steps (and the response to them):

1. upload your code to the Arduino boards.
2. a very brief overview of the included components
3. triggering of the motion system
4. inputting the correct password
5. inputting the incorrect password
6. timeout occurring with no user keypad inputs
7. use of backspace to delete given inputs
8. use of backspace when there are no inputs
9. submitting password when there are no keypad inputs from the user
10. more than 4 inputs before submitting password
11. less than 4 inputs before submitting password
12. inputs given and timeout occurs before submitting password
13. additional features and their function

You must start with step 1. but you can include other steps at a random order, as long as they are included. The alarm system should clear the steps with no crashing or bugs. You can choose the appropriate exception handling for the incorrect user inputs, but explain your choice.

Try to keep the video under 10 minutes. You can film your demo using your phone or similar devices. Submit your demo video along with your codes and report via Moodle. If the demo video is too large for Moodle, you can upload the video to your OneDrive and share it to the instructors email at lauri.jarvinen@lut.fi or add a link to a hidden YouTube video in your report. If you encounter problems or have questions, about the exercise work, you can contact the exercise instructor.