Open your mind. LUT.
Lappeenranta University of Technology

BL40A1812 Introduction to Embedded Systems

Exercise Report

Group 07
Srinivas Babu Pandi 874650
Vili Raunola 0543366
Eero Suomalainen 0565289

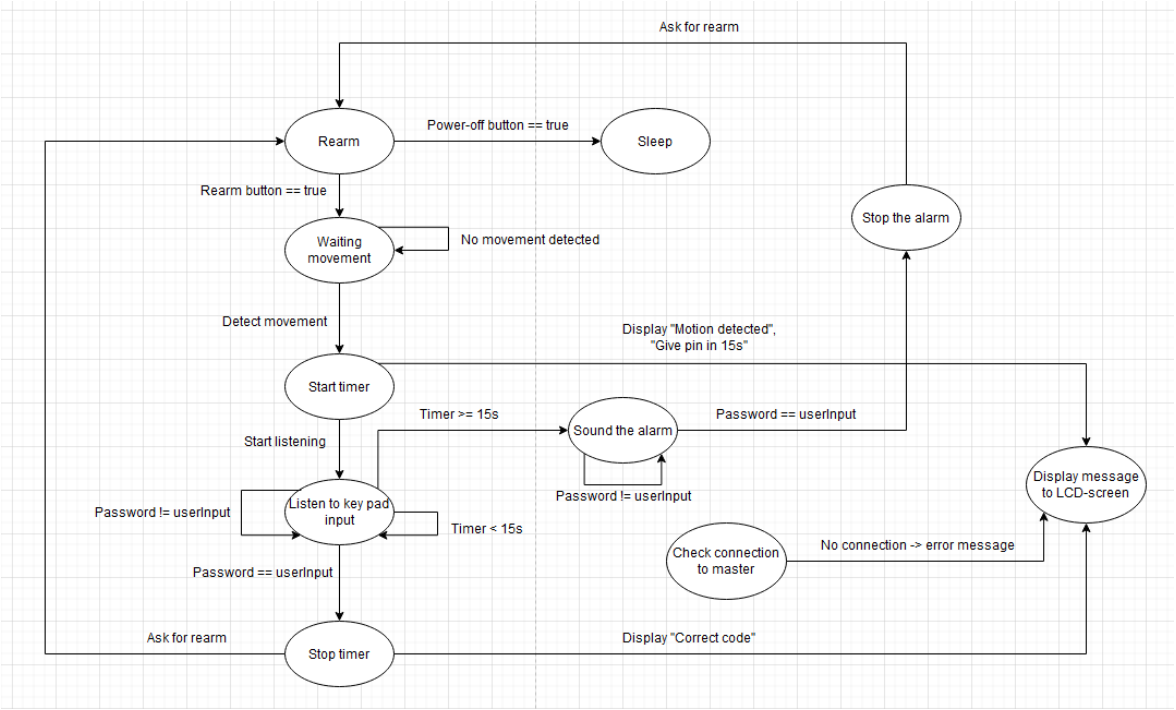
1. OVERVIEW OF PROJECT

This exercise work (later referred as the project) was done as a part of Introduction to Embedded Systems-course. The project is an alarm system built using Arduino Kit and its components. The system is built around motion detection and it sounds the alarm after 15 seconds when motion is detected. User can input a four number password during these fifteen seconds and if correct, the alarm is deactivated. Correct password deactivates the alarm whatever it's triggered or not. The system informs user on whatever the password is correct or incorrect. The system uses both Arduino MEGA and UNO boards with MEGA being a master, UNO being a slave and both feature a state machine structure. The project was done in C using Atmel studio.

The project also features following additional features:

- LCD screen to display information
- Use of interrupts
- Rearming the alarm system
- Use of sleep modes
- EEPROM to store and change password

2. STATE MACHINE



State machine diagram of the system

The system starts at rearm state. Here user can press (on keypad) button A to arm the system or button B to power-off the system. If button B is pressed the system goes to sleep and can be activated again only by pressing reset button on MEGA and UNO boards. After arming the system, it starts to wait for movement and stays in this state until movement is detected.

Once movement is detected it switches to start timer state during which it sends messages “Motion detected” and “Give pin in 15s” to slave to display in LCD screen. After this it starts to listen user input and switches to next state. In this state the system waits for user input for 15 seconds after which it triggers the alarm. Once alarm is triggered user can still input the password and if correct the alarm stops and the system asks for rearm.

If correct password is given in 15 seconds the timer is stopped, message “Correct code” is displayed in LCD screen and system asks for rearming. Also during these states the slave checks the connection to the master and if no connection is found it displays the error message.

3. MASTER

Arduino MEGA-board is the master of the system. It controls keypad and sensor and hence it uses keypad library including all files and headers in it (as they are in the course page). The master uses USART to communicate with computer and ISR functions to sense movement and trigger the alarm. In addition, the master uses following functions of its own:

- send_command_to_slave: Uses SPI to send commands to the slave byte at the time
- comparePassword: Compares user's input to password stored in EEPROM
- appendCharToCharArray: Appends a character to a character array
- createUserInputString: Creates a string to for LCD to display
- removeLastChar: Removes the last char of the array
- getPassword: Reads input from keypad and sends it to comparePassword
- askToRearm: Handles rearming and powering off the system
- Interrupt_init: Initialising the interrupt and timer
- start_timer: Initialises and starts the timer

The master's main function initialises the system (saves the password to EEPROM, sets MEGA as master, etc.) and the starts the while loop. The while loop uses switch-case to alter between different states. It is initialised to start with "WAIT_MOVEMENT"-state during which it uses sleep mode. When movement is detected, it switches to "MOTION_DETECTED"-state, sends information about it to the slave to display in LCD screen and then switches to "KEYPAD_INPUT"-state. In this state it uses getPassword- and comparePassword-functions to run password functionalities and if correct password is given it switches to "DEACTIVATE_TIMER"-state. Here information about deactivation is send to the slave after which state is changed to "REARM". In this state the system asks user to either rearm the system, switching back to "WAIT_MOVEMENT"-state, or to power-off the system, entering sleep mode. In most of these states the master uses *send_command_to_slave* -function to swich states on the slave side which in turn perform functionalities, such as activating deactivating the buzzer.

4. SLAVE

Arduino UNO-board is the slave in the system. It controls buzzer and LCD-screen and hence it uses lcd-library including all files and headers in it (as they are in the course page). The slave uses USART for communication. In addition the slave has following functions of its own:

- topCalculation: Calculates frequency for the buzzer
- receive_command_from_mega: Checks from payload from master to which state to switch and what message to display on LCD-screen.

As with main function on the master's side, the slave begins by initialising values on its own end (setting buzzer's frequency, LCD-screen, timer, etc.) and after that a while loop which uses switch-case to alter between different states. Slave is initialised to start with "WAIT_COMMAND"-state in which the slave waits for the command from the master. Based on command received the slave switches to "BUZZER_ON" or "BUZZER_OFF" to activate or deactivate the buzzer, "DISPLAY_FIRST_ROW" or "DISPLAY_SECOND_ROW" to display message on the right row of LCD-screen, "DISPLAY_CLEAR" to clear LCD-screen or "POWER_OFF" to enable sleep mode. Excluding the last one all states switch back to "WAIT_COMMAND"-state after all other operations are completed and, as with the master, powering off the slave puts it to sleep and then it can only be activated by restarting the slave.

5. COMPONENTS

List of components (as they are in Uno Starter Kit) used in the project:

- Arduino Mega
- Arduino Uno
- Membrane Switch Module
- Passive Buzzer
- LCD1602 Module
- PIR Motion Sensor (HC-SR501)
- Potentiometer (10kΩ)
- 1 x 220 kΩ resistor
- wires