Smart Greenhouse Project

Description:

- The aim of this project is to monitor and control a Greenhouse climate.
- Temperature and Light intensity are to be controlled.
- The temperature must be maintained between 20 - 25 Celsius.

System components:

- Microchip (Atmel) SAM3X8E – ARM embedded computer platform (Arduino due)
- Keypad (Port I/O)
- Photosensor (A/D)
- Temperature sensor (time)
- RC servo motor (PWM)
- Switch (HW-int)
- Display (port I/O)
- LEDs

Grade 3 requirements:

Req. 1: The system should have a calendar that allows showing date and time. The user should be able to configure it whenever it is needed using the keypad and display. The date should be represented as ***DD/MM/YYYY***, while the clock as ***hh:mm:ss***. (*Use the* ***24-hour clock system***). **SysTick** can be used as a base for the calendar.
Purpose: timestamp

Req. 2: Periodic temperature recording every minute for a duration of 7 days. The recorded temperature should be time-stamped with the time moment that it was measured. During recording the temperature, if the system memory buffer is full, the recording should manage to delete an old recording and add a new one instead (deleting an old node from the Linked List and adding a new one instead). *Linked Lists* must be used as a data structure to hold the recorded information.
Purpose: Recording of temperature.

Req. 3: Presentation of recorded data on the LCD by text. Each day is presented by the minimum, average, and maximum values for temperature. Maximum and minimum values should be presented along with their timestamp.
Purpose: Presentation of logged data.

Req. 4: The greenhouse has a large horizontal glass roof surface, shades and a motor-controlled mirrors that faces the sun to reflect the sun light through the glass roof to the plants inside. The motorized mirrors tracks the sun (with the help of the photosensor) to reflect it inside

the green house, while the shades control the amount of light that enters the greenhouse by turning it on and off. For the healthy growth, the plants should have 16 hours of light and 8 hours of darkness every day. Unfortunately, the required ratio of light to dark cannot be maintained naturally since sun light duration varies drastically between summer and winter in Sweden. To tackle this issue, a lighting system is installed (modelled by the LED) to compensate the lack of sunlight in winter. The system should track the sun and control the mirrors and shades along with the lighting system (LED) to achieve the healthy growth light to darkness ratio for the plants. The sun can be simulated by any bright light source and the lighting system can be modeled by the LED. The implemented system should show on the display the position of the sun (in degrees), the number of hours of sun light, the number of hours of Lighting system, the number of hours of darkness, the plant got in the current moment (Problem, to achieve this req. Should operate in periodically)
Purpose: light to darkness ratio, Sun position

Req. 5: The greenhouse temperature should be maintained between a defined upper and lower limit. An alarm should be raised in case these limits were crossed. The alarm signal could be turning on a led or showing a blinking alarm message on the lcd. The alarm signal should be kept on until the user acknowledge its occurrence and resets it manually. The upper and lower limits must be configurable by the user through the keypad and the lcd display.
Purpose: Alarm at high or low temperature.

Req. 6: Create a fast mode that achieves **Req. 2** where each 30 minutes are simulated by a second.
Purpose: High speed simulation to simplify testing.

Req. 7: Draw a functional block diagram (Not a flowchart or schematic or wiring diagram), which illustrates graphically how the system is composed. The functional block diagram should be included in the project documentation (not as a separate file). Also, the diagram should be followed by a paragraph that explains it in detail.
Purpose: Illustration of how things are connected from a higher abstract view.

Req. 8: Documentation of the project. A report which specifies the modules of the system, with a clear connection to the project requirements, so that it is possible to track which requirements are achieved. *The documentation is a part of the deliverables for the final exam. Missing the submission of the documentation before the mentioned deadline means failure.*
Purpose: Document what you have done according to the project instructions.

Grade 4 requirements: You need to satisfy grade 3 requirements in addition to the following.

Req. 9: Periodic temperature recording every minute. Each minute, the temperature is recorded N times and averaged to give one number; data is stored for as long as there is memory available. When memory is full an indication is made in the user interface and the oldest values

are overwritten, in a circular fashion. N is selectable within the range 1-10, in a settings screen by the keypad. The N temperature values, as well as their average value, should be stored with their timestamp. (This requirement supersedes **Req. 2**)
Purpose: Recording of temperature.

Req. 10: Use two photosensors to achieve the function of **Req. 4**. (This requirement supersedes **Req. 4**)
Purpose: Tracking of sun position.

Req. 11: Presentation of recorded data on the LCD display using graphs. Each day is presented by minimum, average, and maximum values for temperature.
Purpose: Presentation of logged data.

Req .12: Create a test mode for **Req. 11** using pre-recorded data to test the graph mode.
Purpose: Test mode for graphs.
Req .13: Documentation of grade 4 requirements.
Purpose: Document your work.

Grade 5 requirements: You need to satisfy Grade 3 and 4 requirements in addition to the following.

Req. 14: The smart greenhouse system should include a Username/Password feature to allow users to access its features. It is not secure to store the password as plaintext. A general approach is to store a hash of the password which is the output of a secure one-way function. When the user inputs the password, the hash of the password is computed and compared to the stored hash. Using a regular secure hashing function directly with passwords is problematic for multiple reasons:

1) As passwords tend to be short, using one hash function is fast and makes it easy for an attacker to perform a dictionary attack and guess the password.
2) Two users with the same password will have the same hash.

To resolve those issues techniques like iterative application of hashing and extending the password using random bits (salting) can be used. Read the following information about password hashing: https://botan.randombit.net/handbook/api_ref/passhash.html

To simplify this assignment, you are asked to implement MD5 hashing in order to hash the password. MD5 is easy to implement, however you should remember that in a real setting you should not directly use it and that it is better to use a stronger function that is already implemented in a library. However, since we are not using an operating system in this project, you should implement MD5 yourself. You can find explanation of the algorithm and a method of testing the evaluation here: http://practicalcryptography.com/hashes/md5-hash/
Purpose: Implement and test MD5
Req. 15: The system should have one administrator with username admin. You should initially have a default password for which you store the hash. However, the user should be able to

change this password. To change the password, the user should first login, then select a new password. Only the hash of the new password should be stored.
Purpose: Resetting the password
Req. 16: Extended test mode. Create a component testing mode so that the operation of each component can be validated (unit test).
Purpose: To simplify testing.
Req .17: Documentation of grade 4 requirements.
Purpose: Document your work.