2023 Session 1 ELEC3042 Minor Project

Introduction

For the minor project, you are to design and implement a digital stopwatch. The main I/O for your stopwatch will be the HW262 expansion shield. You will use the 4-digit, 7-segment display as the main display for the time, the 3 push buttons and POT as user input, and the 4 LEDs for additional output.

Your system should not, and may NOT use pins Port D bits 0 and 1, as they are reserved for the programming system. You may only use xc.h, avr/io.h, avr/interrupt.h and avr/sleep.h system header files. You may write your own header files. **Your code must compile properly with no errors or warnings.**

This document summarises the specifications and requirements for the minor project. You should read the entire document carefully before starting the project. If you feel any part of the specifications are ambiguous, please post your question to the General Discussion Forum on iLearn. Do not make assumptions.

This is an individual project. Your C code and report must be your own work. You must not share your files with anybody else.

Pass Specifications

Operation Requirements

The general operations of the stopwatch are as follows. The initial state of the stopwatch is 0 minutes, 0 seconds and 0 fractions ($1/10^{th}$) of a second. The stopwatch should start counting upwards from its initial state when the start/stop button is pressed. While counting up, pressing the start/stop button will pause the count and the time the button is pressed should be displayed. Pressing the start/stop button again should resume the count. Pressing the reset button should reset the stopwatch to its initial state.

The 7-segment display will indicate the current time of the stopwatch. The time should be shown with the format M.SS.f, where the left-most digit shows minutes (M) passed, the middle two digits show the seconds (SS), and the right-most digit for fraction of a second (f). The dot (.) is the decimal point on the 7-segment display. In its initial state, the display should show 0.00.0.

Button S1-A1 is used to reset the stopwatch to its initial state.

- Pressing S1-A1 at any time should stop the count and reset the display.

Button S2-A2 is used to start or pause/stop the stopwatch.

- If the stopwatch is not currently counting, pressing S2-A2 once will start the stopwatch counting up.
- If the stopwatch is currently counting up, pressing S2-A2 will pause the count and display the time when the S2-A2 was pressed.
- If S2-A2 is pressed and the display is showing a time other than 0.00.0, it should resume counting upwards.
- Pressing S2-A2 after a reset should start the count from 0.00.0.

LED D4 is used to indicate that the stopwatch has been paused.

- When the stopwatch is in its initial state, D4 should be off.
- When counting up, D4 should be off.
- When paused, D4 should be on.

Specific Implementation Requirements

1. You must use a timer interrupt to keep time. You may not use busy loops.
2. You must debounce the buttons. As a hint, each button needs to be debounced individually.

Specific Reporting Requirements

Your report must include the state diagram and state transition table for your system, and include line references for where these have been implemented in your code.

Credit Specifications

You must complete all Pass specifications before attempting the following.

Operation

The POT will be used to change the brightness of the 7-segment display and LEDs.

- The brightness should increase when turning the POT in a clockwise direction, and decrease when the POT is turned in the counter-clockwise direction

Specific Reporting Requirements

Explain how you managed to change the brightness of the 7-segment display. In particular, you should justify the number of bits of the ADC used and how they map to the frequency at which the display is being updated.

Distinction Specifications

You must meet all Pass and Credit specifications before attempting the following.

Operation

When the stopwatch is counting up, button S3-A3 is used to store the current time in memory. LED D1 should light up when a time is stored in memory.

When the stopwatch is paused, button S3-A3 is used to recall the time stored in memory.

- Pressing S3-A3 when the stopwatch is paused recalls the time stored in memory and shows it on the 7-segment display
- When showing the time stored in memory, LED D4 should be turned off and any button press of S2-A2 (start/stop button) should be ignored
- Pressing S3-A3 again returns the display to the time when the stopwatch was paused. LED D4 should be on.

Pressing S1-A1 (reset button) at any time should reset the stopwatch to its initial state, clear the memory, and turn LED D1 off.

Specific Implementation Requirement

You may only use one timer to maintain the time of the stopwatch, debounce buttons, and change the brightness of the 7-segment display.

Specific Reporting Requirement

Explain how you were able to use one timer to implement the system.

High Distinction Specifications

You must meet all Pass, Credit and Distinction specifications before attempting the following.

The stopwatch is to have three memory slots.

When the stopwatch is counting up, button S3-A3 is used to store the current time in memory.

- Each time S3-A3 is pressed, the current time is stored to a memory slot
- When a time is stored, one of the LEDs (D1-D3) should light up to indicate a time is in a memory slot. That is, when one time is stored, LED D1 is lit. When two times are stored, LEDs D1 and D2 are lit
- When all memory slots are filled, any additional presses of S3-A3 is ignored

When the stopwatch is paused, button S3-A3 is used to recall the time stored in memory.

- Pressing S3-A3 once when the stopwatch is paused recalls the time stored in memory slot 1 (if stored) and shows it on the 7-segment display. When showing memory slot 1, only LED D1 should be on. If no time was stored in memory slot 1, this button press should be ignored and the paused time should remain shown.

- Pressing S3-A3 again shows the time stored in memory slot 2 (if stored) on the 7-segment display and only LED D2 should be on. If no time had been stored, this step should return to the paused time.
- Pressing S3-A3 again shows the time stored in memory slot 3 (if stored) on the 7-segment display and only LED D3 should be on. If no time had been stored, this step should return to the paused time.
- Pressing S3-A3 again returns the display to the time when the stopwatch was paused. LED D4 should be on. Make sure all occupied memory slot LEDs are on.

Pressing S1-A1 should reset the stopwatch to its initial state and clear all memory slots. Reset can be pressed at any time.

Specific Implementation Requirement

You must put the microcontroller to sleep when no work needs to be done by the CPU.

Specific Reporting Requirement

Justify the way you used the hardware to implement the stopwatch memory, and on the efficiency of your hardware usage.

General hints on how to approach this project:

- Decompose the system into smaller components and tackle the design and implementation of each component separately
- Keep display/output aspects of your system as components that are separate from the main state machine
- For each component, draw the state diagram and state transition table that describes its' functionality before writing the code
- Use the state diagram to help you structure your code
- After writing the code for one component, use the state transition table to test the component to verify that it behaves correctly
- When all components are correct, then combine them together to complete the final system
- Complete lower-level requirements before moving to higher level requirements. That is, complete all pass requirements first before attempting credit requirements. If you follow the steps in these hints, implementing the higher-level requirements should only require minor modifications to your code.

Assessment

Your project will be assessed in three parts: a report documenting your design, implementation, and being able to meet the requirements by passing a series of requirements tests.

1. Report (20%)

You must submit a report that documents your design. It should include:

- A high-level description of your design. This should include a table showing the major components of your design, or a description of their functionality. You should also include a block diagram showing the major components and the signal flows between them.
- State diagrams and state transition tables for each major component.
- Your code as an appendix. Your code should be commented so that it is easy to identify the major components as outlined in the high-level description.

Do not submit hand-drawn diagrams. You can use any software you like to draw your diagrams. One suggestion is to use the online tool found at: app.diagrams.net.

Your report will be marked according to the following rubric

Mark	Description
0	Report is poorly presented and is not understandable.
1	Overall design contains major flaws, report is missing parts, or diagrams are hand drawn.
2	Report is understandable but design contains errors
3	Report is good but minor errors in the design may be noted
4	Report is a pleasure to read and contains no errors

2. Implementation (20%)

Your code will be assessed for coding style according to the following rubric

Mark	Description
0	Code is unlikely to work and/or is poorly commented
1	Code may work but shows a lack of understanding of good embedded systems programming practices (for example, use of busy loops to keep time)
2	Code demonstrates a basic understanding of embedded systems programming by correctly setting up input/output pins, and use of timer subsystem
3	Code demonstrates a good understanding through proper use of interrupts
4	Code demonstrates an excellent understanding of embedded systems programming by efficient usage of hardware resources

3. Requirements Tests (60%)

Note: A 10% penalty will be applied to this component if your code generates compiler warnings or errors.

Pass requirements (65% of this section)
1. Displays 0.00.0 at start, LED D4 off
2. Stopwatch starts counting up when S2-A2 pressed
3. Seconds update correctly
4. Minutes update correctly
5. Stopwatch pauses when S2-A2 pressed
6. LED D4 on when stopwatch paused
7. Stopwatch resumes count when S2-A2 pressed
8. LED D4 off when stopwatch resumes
9. Stopwatch resets correctly when S1-A1 pressed while running
10. Stopwatch resets correctly when S1-A1 pressed while paused
Credit requirements (10%)
11. Turning POT clockwise increases display brightness
12. Turning POT clockwise increases brightness of LEDs
13. Turning POT counter-clockwise decreases display brightness
14. Turning POT counter-clockwise decreases brightness of LEDs

Distinction requirements (10%)
15. Pressing S3-A3 while stopwatch running should light up LED D1
16. Pressing S3-A3 while stopwatch paused should recall time in memory; LED D4 off
17. Pressing S3-A3 again while stopwatch paused should display time of pause; LED D4 on
18. Stopwatch resets correctly. Memory after reset should show 0.00.0
High Distinction requirements (15%)
19. Stopwatch correctly indicates the number of times stored in memory
20. Times stored in memory can be recalled correctly
21. LEDs update correctly when cycling through recalled times
22. Stopwatch resets correctly. All memory slots after reset should show 0.00.0

Document History:

Date	Revision	Author	Change
4/1/23	1.0	Alan	Initial revision. Entered first description of project
3/2/23	1.1	Rex	Minor updates.
10/2/23	1.2	Alan	Minor formatting fixes
27/2/23	1.3	Alan	Fixed discrepancy found by Gerry (1/100th to 1/10th)
14/3/23	1.4	Alan	Fixed button discrepancy found by Peter (A1 - A2 allocation changed between requirement sections)
16/3/23	1.5	Rex	Corrected button discrepancy in requirements tests. Added clarification of how memory recall should work. Added sleep requirement for HD.