ELEXATHON

PROBLEM STATEMENT

This problem statement should be conducted using the Keil µVision compiler and simulator. In all cases, the code should be structured. All code should be appropriately documented.

Use the Logic analyser in KEIL µVision to plot the output, response and the start and stop button signals.

PART 1:

As part of an early prototype of a control system for lowering an aircraft undercarriage, you are asked to create a system which generates the following sequence of outputs on an 8051 MCU:

OUTPUT	OUTPUT TIME	RESPONSE TIME	RESPONSE PIN
P1.0	10 seconds	1 second	P2.0
P1.2	17 seconds	2 seconds	P2.1
P2.7	30 seconds	1 second	P2.2
P3.1	32 seconds	3 seconds	P2.3
P2.6	40 seconds	1 second	P2.4

- “Output time” refers to the time after the application “Start” button is pressed (see below).
- “Response time” refers to the maximum time before a positive pulse is detected on the corresponding Response pin after each output is generated.
- The sequence of outputs will be initiated by pressing the “Start” button (connected to Pin 3.0).
- The lowering of the undercarriage must be aborted within 10 ms if – at any time during the above process – the “Stop” button is pressed. The Stop button is connected to Pin 3.2.

CRESCENDO MATRIX

The Technical Realm of glitches

You are required to:

- Create suitable software for this system that meets these timing constraints. Ensure that your code is as portable as possible. Ensure that your code is appropriately structured and flexible enough to deal with specification changes (e.g. change of port pins, addition or removal of output steps in the sequence).
- Fully document all of the tests you perform (using the Keil simulator) to confirm that your program operates correctly and meets its timing constraints.
- Please use a “Super Loop” architecture (no interrupts).

PART 2:

Once you have confirmed the correctness of the functionality as specified in Part 1,repeat the same problem statement using a timer interrupt using the most appropriate timer in the 8052 Microcontroller that will generate a ‘tick’ at the most suitable interval. Use this ‘tick’ to control the firing of the appropriate tasks. Use the idle mode of the microcontroller to put the microcontroller to sleep when it is not executing any task to achieve power saving.

Please start by considering how many tasks you will require in your system. Trigger the tasks at the correct time to achieve the same functionality as in Part 1.

Comment on the reliability of the systems created in Part 1 and Part 2.

Sample of inputs/outputs to/of the system for understanding what needs to be done.
The system has got to switches Start and Stop.
1) Start starts the system operation
2) Stop aborts the system operation.

S1 to S5 are the steps of the sequence i.e. the output.
R1 to R5 are the Responses to the Corresponding sequences S1 to S5.
All the pins are visible in the Logic analyzer. The grid of the LA is set to 5sec

ELEXATHON

PROBLEM STATEMENT

Sample Timing Outputs and Responses achieved by Part 1(Super loop architecture)

S1=9.9 sec	S1 error=1%	R1=10.9 sec	R1 error=0.9%
S2=16.81 sec	S2 error=1.1%	R2=18.81 sec	R2 error=1%
S3=30.23 sec	S3 error=-0.76%	R3=31.2 sec	R3 error=-0.62%
S4=32.2 sec	S4 error=-0.62%	R4=35.2 sec	R4 error=-0.57%
S5=40.15 sec	S5 error=-0.37%	R5=41.15 sec	R5 error=-0.37%

Sample Timing Outputs and Responses achieved by Part 2(Timer Interrupt)

S1=10.00005 sec	S1 error=-0.0005%	R1=11 sec	R1 error=0%
S2=17.00005 sec	S2 error=-0.0005%	R2=19.00001 sec	R2 error=0%
S3=30.00005 sec	S3 error=-0.0005%	R3=31.00001 sec	R3 error=0%
S4=32.00005 sec	S4 error=-0.0005%	R4=35.00001 sec	R4 error=0%
S5=40.00005 sec	S5 error=-0.0005%	R5=41.00001 sec	R5 error=0%

CRESCENDO MATRIX

The Technical Realm of glitches