Rochester Institute of Technology

SWEN 563/CMPE 663/EEEE 663 Project 1

Overview:
Using the STM32 Discovery board design and implement an embedded, bare-metal (no operating system) program that will display a list of counts for one thousand rising edge pulse inter-arrival times. The inter-arrival time between pulses is expected to average around 1.0 millisecond, but the listing should represent the range of 101 “buckets” (one bucket per microsecond) between the default values of 950 and 1050 microseconds. However, these upper and lower limits must be user configurable via the virtual terminal.

User interface:
After successfully completing the POST routine (see below) design your user interface to operate as follows (using the virtual terminal):

- On startup the program displays the upper and lower limits and allows the user to either accept those values or to change the lower limit to a new value.
- The lower limit can be from 50 microseconds to 9950 microseconds.
- The upper limit will always be 100 microseconds longer than the current lower limit.
- After reviewing and possibly changing the limits the program issues a start prompt and waits for the user to press the Enter key.
- After completing the 1000 measurements (1001 rising edges) the program displays every non-zero via the puTTY logging feature. Do not display entries with 0 counts.
- The list of counts display should have the time in microseconds as the first column and the count in the second column.
- The list of counts should be in ascending time order. A typical display would have 3 lines like this:
  ```
   998     5
   999   950
  1000    45
  ```
- After completing the display of the non-zero entries provide the option to run again with either the same or different limits.
- Test the oscilloscope test output (it is 1 KHz) and compare your results with the signal generator.

POST:
Perform a Power On Self Test (POST) when the program starts. This test must confirm that the GPIO port is seeing pulses at least once in 100 milliseconds. If it fails this test give the user the option to try the POST again. If the POST is successful then proceed to the normal user interface operation described above.

Note that this program will be relevant to subsequent projects so be sure to use good software development practices. Refer to the C programming standard for the course.

Report and Submission:
In addition to the standard report requirements the report for this project must include the following:

- Screenshot of results for two significantly different signal frequencies.
- Comparison of your results at 1 KHz when using the signal generator and using the oscilloscope test signal.

Submit your report and source code to the Project 1 Dropbox in myCourses. Refer to the Report Specification for the format and contents of your report.

This project will be demonstrated in class to the instructor or Course Assistant. The report is due on the same day as the project.

Grading Criteria:

- Program Operation and Demo – 50%
  - Hardware setup is orderly and well organized – 10%
  - Demo sheet functions all completed – 30%
  - Demo operates without faults or restarts – 10% (except for POST verification)
- Program Design --- 15%
  - Proper initialization
  - Correct use of functions (no copy/paste/edit slightly)
  - Separation of hardware related code from pure software (e.g. the results reporting code)
- Source Code Structure and Readability – 10%
  - Appropriate use of white space – 2%
  - Consistent and good indentation – 2%
  - Appropriate comments at the function and paragraph levels (such as a for loop) – 2%
  - Following C style guide (good names, etc.)
- Report Content – 25%
  - Report is at least 2 pages (not counting pictures, cover page, diagrams) – 5%
  - Demonstrates team understands the problem, solution, and technology (hardware and software) – 10%
  - Report contains all required sections per the report guidelines including all project specific reporting requirements – 10%