ENB 350 Problem based learning Project – Assignment

Work piece Testing Station

(A) Testing Automation Module
(B) LCD Keypad User Interface

Date Available on Blackboard: Week 3
Date Demonstration due: Week 13 at scheduled time for your group
Date Report due: before Week 14 Monday 5pm
Weight: 30%

Problem Description

You are automating one unit of a production line, the testing station – where work pieces arrive, measurements are taken, and pieces are sorted into accept and reject bins. Sensors and actuators have already been installed on the unit (the testing station) and an electrical interface that makes the inputs and outputs compatible with the requirements of a microcontroller application development system (RCM 4000 based on the Rabbit 4000) has also been built and installed. A program must be written to drive the unit (referred to as the Testing Automation Module). This module performs such tasks as sensing whether a work piece is in place, taking sensor readings, raising and lowering platforms and ejecting a piece. For the purpose of this assignment work pieces will be manually placed one at a time into the slot where they are tested. Further, it is desired to have a user interface with an LCD Keypad such that work piece characteristics can be displayed as they are processed and an operator can set quality control parameters such as the minimum and maximum height for accepted work pieces. Your task is to design and implement (A) the Testing Automation module and (B) the User Interface.

Functionality

The testing station checks work piece characteristics **colour**, **material** and **height**. This information must be recorded along with a **work piece number** that increases by one with each piece processed. Work pieces outside a specified range of heights are to be rejected and ejected from the bottom while those that are accepted are ejected from the top. Work piece data should be recorded in a data structure and a **recent history of the last 16 work pieces** processed should be in memory such that the station can be stopped and this data examined via the LCD Keypad interface. The user interface should be able to **start and stop** the system. It should also be able to **display** one of work piece data depending on the mode selected. The **different modes** will display one of colour, material, height, time of processing, decision and upper/lower thresholds. It should be possible to switch from one display mode to another using the 'F1' key or the 'F2' key. The interface should also work in a **settings** mode when the station is stopped, allowing **upper and lower thresholds to be changed** and the changed setting should take effect when the station is started again.

Group work

Each assignment group should be a union of 2 laboratory exercise groups of 2 students each. You need to discuss with other students and form the assignment group. A 3 student group will be permitted by the unit coordinator as an exception when unavoidable.

Each sub-group of 2 students will be responsible primarily for one of the 2 sub-systems - (a) Testing Automation module or (b) LCD Keypad Interface. The two sub-groups must work together.

You use the group allocation system on Blackboard to inform the unit coordinator of your group members after you have had discussions and formed groups. Changes are discouraged but in the event that a group must rearrange, you must notify the unit coordinator by email and group memberships will be changed. In order to ensure that all students are in groups without delays holding up progress on assignment work, please do the allocations before the end of week 4.

Details and Clarification

Hardware

FESTO miniaturized production module – testing station
Serial cables, programming cable
RCM4000 core modules and prototyping boards
FESTO-Rabbit interface, Power supplies

Development Software

Dynamic C (version 10.66) and sample programs
µCOS-II pre-emptive multitasking kernel library
HyperTerminal
Shareware serial port communication protocol analysis tool (PAT)

Data structure
The data format suggested is

Field	**size**
Piece number	32 bits
Colour, material, decision	2 bits (For serial transmission use other bits for error control) Orange – 00, Black = 01, Metallic = 1X
Decision	1 bit (Use other bits for error control) 1 = Accept, 0 = Reject
Height	16 bits
Time Stamp	32 bits (in seconds since system start – to be converted to date/month/year hr:min:sec format)
Upper threshold	16 bits
Lower threshold	16 bits

Communication

The RCM4000 boards in the laboratory have been interfaced with the LCD keypad module using a **serial port**. The RS232.LIB library implements serial port communication functions and laboratory exercise 3 covers this topic. The TAM will need to use this to send and receive data from the user interface.

Stop and Start Switches

The TAM must use **one switch on the board to stop** the testing station. It should be possible to **start** the system with the other switch. Switches here refer to the push button switches on the RCM4000 prototyping board accessible via the hole on the top of each box. You should be able to freeze the action by pressing it such that the automation module remains in its current state. If the riser was up, it remains up and if it was down it remains down. If there was a work piece in, it will not be processed further. These switches are not intended to replace the emergency stop button on the testing station which disengages the pneumatic circuit.

Concurrent tasks / threads of execution

Each program should be a multitasking program with at least two threads of execution. Particular attention must be paid to 'waiting periods' and their exploitation by concurrent tasks. You may use co-operative multitasking or use the Micro C/OS-II pre-emptive multitasking kernel. Depict your designs in the report using state charts. Justify your choices.

LCD display and keypad entry format

Use a suitable display format on the LCD. It has two lines. There are seven buttons (four arrow keys, F1, F2 and Enter) for keypad input. The LCD/keypad is connected via a serial port to the Rabbit 4000. It needs to be in a few different modes of operation – display real time mode, settings mode and display offline mode.

Up arrow – to go to the next setting up
Down arrow – to go to the next setting down
Left arrow – to go to the previous work piece in the recent history buffer
Right arrow - to go to the next work piece in the recent history buffer
F2 – to switch between states within a state
F1 – to switch between top level states

Height Readings

The height reading coming from the A/D converter is an integer of 11 bits in singled ended operation. To read the A/D a driver is used. Sample programs show how to calibrate the system and how to read scaled or raw values from the sensor through serial port B. The structure suggested for each work piece has two 16 bit fields for the upper and lower thresholds. These are enough to store raw readings in integer form. If you don't scale the values into height units, you will display decimal digits of this integer (unsigned) on the LCD display.

To store actual height (say in mms) and use a 16 bit field you will need to represent the number after scaling in 8.8 fixed point format. Then these fields can be (signed) int. Note that the LCD display will only display characters (and symbols) or strings. You will need to convert to decimal digits and 'dot' and send these across as a string to display if you use a fixed point notation.

There are reference work pieces of known height (measured using Vernier Callipers). You need to calibrate the height measurement using two reference work pieces. This is a combined requirement for the group.

RTOS

You can use Cooperative multitasking with Dynamic C or preemptive multitasking with µcOS-II. Marks for the report can be higher for the latter provided there is evidence of additional or deeper learning as a result of the use of a preemptive kernel. Marks for demonstration are based on meeting requirements only.

Testing station

There are two testing stations in the laboratory but they are not identical in operation. Both have Rabbit-FESTO interfaces. If you develop your system using one or the other – it will be easier to demonstrate on that. In the rare event that one of them goes out of operation, the other one is a fall back option.

Baud rate for serial communication

Use the default setting of 19,200 baud. This setting should not be changed. If it is changed others will not be able to use that board and the module will need to be reset. Power on reset to factory settings happens only if a particular jumper on the board is in place (page 13 of the manual).

Sharing and Use of the testing station

This will need to be on mutual agreement and on a first come first served basis. If towards the end of the semester, there is a need to set up a roster to book the station in advance in slots of 1 hour or 2 hours, that will be done. It is highly recommended that you start work early and do not leave work towards week 12 when such problems can arise.

Order of measurements

Use the order – colour, material, height, timestamp and decision, for display. The order of measurements is your decision subject to constraints imposed by the system.

Timestamp

This should be the time when the piece is ejected, when all measurement processing has been completed. It can be displayed in the format yyyy/mm/dd::hh:nn:ss where yyyy is year, mm is month, dd is day, hh is hour, nn is minutes and ss is seconds.

Graphical representations for design and inclusion in the report

You should include flow charts and state diagrams where appropriate. State charts are explained in the reference book on real-time system design by Philip Laplante. Microsoft Word diagrams will suffice. A freeware tool called Edraw supports the creation of Universal Modelling Language (UML) state chart diagrams.

Requirements Criteria for assessment

Requirements of the TAM (10 marks)

C-R1. Start/stop the testing station using switches on the board
C-R2. Operate ejectors and platform movement correctly from start through all steps to stop
C-R3. Obtain sensor readings from work piece placement, to colour, material and height
C-R4. Calculate and display timestamp and work piece number correctly
C-R5. Send and Receive data correctly through a serial port to interface with the UI

Requirements of the UI (10 marks)

M-R1. Configure the LCD keypad on initialization
M-R2. Implement the menu structure required for display
M-R3. Receive work piece data correctly through a serial port
M-R4. Send settings correctly through a serial port
M-R5. Interface correctly with the TAM including sensing when it is stopped

Requirements of the integrated system (5 marks)

I-R1. The integrated system should work correctly and continuously
I-R2. Height measurements should be calibrated using reference work pieces

Demonstration and Class Presentation

You must present and demonstrate your work in week 13 during laboratory contact times as indicated in the week 1 study document. The duration is 12 minutes for a presentation by the group (about 3 minutes per member) and 13 minutes for a demonstration. You should prepare a PowerPoint presentation explaining your work. A demonstration should follow the presentation and should be in the form of experimental evidence against the listed criteria while showing the unit in operation with your software.

Assessment Criteria

PRESENTATION (5%) – This will be awarded based on effectiveness of oral communication, structure and quality of visual material and evidence of your understanding of the theory and practice.

ACHIEVEMENT (15%) -- This is based on demonstration of evidence that requirements are met.

REPORT (10%) – A suggested report format is provided at the end.

Documentation

Dynamic C documentation and sample programs are part of the installation.
Documentation is placed on the Blackboard site for the unit. Look under Learning Resources > Labs and Assignment for the ‘Rabbit and Dynamic C documents’ and ‘Other documents’ folders. There are documents describing the FESTO-Rabbit interface, its pin diagram and circuits and a test program, the LCD keypad manual and a reset program.
Micro-C OS-II documentation can be found on the web at www.ucos-ii.com and in the book on this real-time kernel by Jean Labrosse. A number of copies are available for use in the laboratory. Please leave them in the cupboard in the laboratory after you are done reading.

Report

Submit a report in soft copy form (pdf or Word document) with contributions from each member of the group identified in it. Submit the report via the Blackboard site for the unit. The expected length is between 20-30 pages. There is no need to reproduce information from the documentation made available – you can refer to it. Your design, implementation and testing should be documented with program fragments and illustrations/diagrams. Use state diagrams and flow charts for graphical illustration. The report should comprise three sections (a) TAM, (b) UI and (c) Integrated system. They will be 6 marks, 6 marks and 4 marks respectively. Marks for (a) and (b) will be awarded to each sub-group separately and marks for (c) will be common to both. You are also required to submit a copy of the final version of your code. If you request a review of marks later or there were unforeseen problems at demonstration, the submitted code may be checked.

Assignment Return and Feedback

Assignment reports will be marked by the tutor. You will receive individual feedback from the tutor including a break-up of your marks and comments on the report by email within about 10 working days after the deadline for submission.

Late demonstration

This **will not be permitted** except for strongly valid medical reason with presentation of a medical certificate. Regardless of the state of completion of your assignment work, you are required to make a presentation as scheduled describing your work up to that time.

Penalty for late submission of the report

QUT policy for late submission of assignments will apply. A late report, without an approved exemption, will receive 0%.

Improvements after demonstration

There will be no time for improvements because the demonstrations are on Friday and the report is due the following Monday. It is advised that you spend any extra time that you may have in finishing the report. If you describe your work well enough – and your demonstration marks were low – you could be given partial credit to compensate for an inability to demonstrate when it was required.

Some tips

- Do all the laboratory exercises that lead up to the assignment
- Read all necessary documentation thoroughly
- Check hardware before starting.
- Check the Dynamic C installation and Sample programs directory. Try out appropriate sample programs. Look for the RCM4000 subdirectory and programs that test the LEDs and switches, programs demonstrating use of the serial ports, demonstration programs accompanying µcOS-II.
- Read documentation on the testing station.
- Make sure that you have understood the safety considerations in using the station. You should have proof of having completed the health and safety orientation for this laboratory. In particular, be aware of cylinders advancing when compressed air is switched on, and do not change electrical connections without disconnecting the power.
- Note that in normal operation, the FESTO stations rely on signals coming from previous stations and send signals to subsequent stations. The roles of Station Ready, Part Ready and Data Request flags are important. For this assignment, we will have to assume working by placing a work piece manually on the testing station.
- Find out the pin mapping for the Rabbit – FESTO station interface.
- Decide what you are going to do with the analog value of height as a voltage. Check out the A/D convertor sample programs. There are sample programs demonstrating how to acquire samples, how to calibrate etc. Decide how the converted value will be stored and transferred serially using the allocated 16 bit field.
- Read documentation on the RS232.LIB. Do the laboratory exercise where serial communication is introduced. Understand the role of the UART; interrupt driven and polled transfers, serial protocols.
- Set the same baud-rate, parity, data bits, etc at both ends of a serial connection. Check serial communication using a HyperTerminal as done in the laboratory exercise.
- Design threads using states and draw state charts.
- It is possible to use a HyperTerminal to test each module – TAM or UI – separately. This will allow display of the data being sent by each module. You may also be able to enter data in required format to test.
- Read the LCD keypad manual and check the various escape sequences required for different operations. The LCD keypad may have a default configuration. It is best that you learn to configure the device and use this in your implementation during an initialization phase. Otherwise, at the time of demonstration it is possible that the device is not configured as expected and your code will not work.
- Design test inputs. Compare expected and actual outputs. Modify if necessary to meet requirements.
- Demonstrate your work by operating the system and showing outputs that provide evidence against requirements. Leave finer details for the report.

Suggested format for the report

- Cover page – student info, title, the usual stuff
- Table of contents, list of figures, list of tables
- Statement of contribution, one paragraph for each member describing his or her contributions.
- Introduction
  - General Problem description (paraphrased from assignment brief), general context of industrial automation and real-time control for safety and accuracy/precision.
  - Requirements
    - FESTO controller.
    - Data Logger.
    - Additional requirements, such as provided structures, specific scenarios relating to system design considerations, and perhaps justified arguments for alterations to assignment spec (different data structures, etc.)
- Design
  - General design comments (methodology of design), system features, detail shared data structures
  - Controller – Test and Measure (TAM)
    - Functional requirements > including state diagrams/flow charts. Real-time aspects.
    - Implementation > program snippets/pseudo-code, discussion of theory and use of RTOS.
    - Tasks and their priorities.
  - Controller – Data Logger and user interface (UI)
    - Functional requirements > including state diagrams/flow charts. Real-time aspects.
    - Implementation > comment in implementation, code snippets, test results, use of RTOS etc.
    - Tasks and their priorities.
  - Integration
    - System interaction diagram, discussion of critical aspects for operation. Highlight key areas where real-time OS helps with robust control. Maybe describe the Serial interface between the two controllers here the most.
- Results (Post-Presentation)
    - Allows explanation of failures.
    - Conclude and confirm with summary of evidence that real time operating systems are important for robust, fast, reliable industrial control.
- References
- Appendices (maybe pseudo code, but should not be much/anything here)