Embedded Systems Design

Assignment 1 – Washing Machine Controller - Stage 1

Purpose

This assessment is intended to evaluate the student's capability in configuring a microcontroller, designing simple interface hardware and writing C code to operate those interfaces.

Grading of this assessment

This task will be assessed against the course objectives 1, 2, 3, 4, 6 and 7. This assessment will be graded (F, C, B, A, HD) using a checklist marking scheme against criteria such as: the appropriate selection and use of microcomputer hardware; the design of I/O hardware to meet a specification; implementation of software to meet a specification; quality of documentation including organisation of ideas and format; spelling, grammar and punctuation. *Note - this course is a communications benchmark course; hence marks will be awarded for the quality of documentation.*

Assignment Requirements

This assessment requires students to meet the requirements the specification below. **Select a suitable microcontroller from the PIC18 family**, **design simple interface hardware**, **write and test some C programs** (subroutines) to operate the hardware interface and then **document the hardware and software** as a proposed design. **Software is to be written in C using the MPLABX IDE**. You must create a project for your program which may include one or more C source files. The **testing is to be completed** using the PICSimLab simulator.

There is NO requirement to assemble any hardware, or layout a PCB for this assignment. The circuit design for the hardware may be drawn using electronics CAD software, OR neatly hand-drawn and scanned, for inclusion in the documentation. The interface for the application can be successfully tested on the PICSimLab simulator. **A recorded demonstration of the software under-going testing must to be included with the documentation.**

Each student is required to submit:

1. A **report in Microsoft WORD format (<u>with tracked changes enabled</u>)** which includes:
    - **a brief introduction** (150 – 200 words) outlining the design requirements based on the specification.
    - **the hardware design** (400 – 500 words) **briefly explaining** the circuit design and how it meets the specification. **Include a figure showing the proposed user interface panel layout.** Show any calculations you make to select values for components such as resistors.
    - **a single page circuit diagram** of the proposed hardware. This may be A4 or A3 in size. This diagram may be provided in a separate file (possibly an image file) if it is more convenient. **You must specify the electronic components (eg. IC's) and the values of any discrete components used** (ie. resistors, capacitors, crystals). Manufacturer part numbers are not required.
    - **the software interface routines** (600 – 700 words) **briefly explaining** the subroutines and how they operate and meet the requirements. Show any calculations you make to achieve the scaling and resolution of analog input values.
    - **a brief analysis of the testing of the software** (200 – 300 words) explaining <u>what is working and isn't working</u>.
    - **copy of the C source code** showing all the subroutines. (as an appendix to the report)

    The document should be formatted as a report and must include:

    - a title page showing the student name and number
    - numbered sections with appropriate sub-headings
    - grammatically correct English, complete sentences in paragraphs
    - clearly drawn diagrams including figure numbers and captions (may be a separate file)
    - the assignment name and page number in the header of the document
    - the student's name and student number in the footer of the document

2. A **copy of the project directory** (for example **myproject.X**) saved as a ZIP file. Make sure all the files in the directory are included.
3. **A Zoom recording** showing the code compiled in MPLABX and testing of the functionality of the program on the PICSimLab simulator, including a commentary by the student

The Specification

<u>Outline of the task</u>

You have been requested by your Senior Engineer to design a PIC18-based controller for a new washing machine design. You are to prepare a hardware design (including a user panel layout), write low-level code (functions/subroutines) to interface with the hardware, test those routines on a simulator and prepare a report detailing your interface routines for the Washing Machine Controller (WMC).

The sequence of steps and options for the washing machine's operation are yet to be fully defined by the client, however the washing machine's electro-mechanical equipment has been specified. The first step is to develop the low-level interface routines that correctly operate the electrical and mechanical devices which comprise the washing machine. Eg. valves, sensors, the motor and the user interface panel.

The client has specified a PIC18-based controller is to be used, as they are familiar with that type of MCU. The controller is to accept input from a human operator and control the operation of the washing machine. The user interface is expected to include a set of buttons and a display to various modes of operation and a numeric display. The hardware interface to the washing machine equipment is described in <u>Hardware Requirements</u>.

show

Stage 1 of the project (***<u>this assignment</u>***) requires you to (a) **design the hardware** for a PIC18 microcontroller and the input/output circuitry required to interface the buttons, indicators, display and any other devices specified in the <u>Hardware Requirements</u> and (b) **write and test a set of input/output routines** (in C) <u>to operate all interfaces</u> to meet the <u>Software Requirements</u>.

[*Stage 2 (assignment 2) will require you to write a state-based control program to operate the washing machine in a range of modes. Some additional hardware and/or interface routines may also be required.*]

Hardware Requirements

Microprocessor support circuitry

- The circuit diagram for the WMC must include the minimum components necessary to **operate a PIC18 processor at a clock speed of 10MHz**.

- A **push button reset circuit is NOT required** for the microcontroller. A power on reset is required.

- **Show the power supply connections** required to power the PIC processor using a 5Volt power supply. (*Note that the 5V power supply itself is not part of this design problem.)*

User interface panel

- The user interface must continually display the status of the machine on a set of indicator LEDs **OR** another form of display, – capable of indicating 4 states – STOP, WASH, RINSE, SPIN.

- The user interface must provide a set of **active-low momentary** push button switch inputs – which enable the operator to control the machine. Buttons are to be labelled **BACK**, ▲(up), ▼(down) and **ENTER**. No debounce hardware is required on button inputs.

- The user interface must provide some form of display which is can be used to assist the user to select different wash modes. ***(Note- NO detail for washing modes or what is to be displayed are given or required for this assignment.)*** The display must also be able to display numeric values up to 9999.

Washer equipment

The washing machine incorporates the following electrical and mechanical equipment

- The motor used to turn the washing drum is a 240 Volt 50 Hz AC (synchronous) motor. The motor has single direction operation.
  The motor is to be controlled by a logic signal from a PIC port pin.
  A relay must be used to switch the 240V supply to the motor. The relay coil is rated at 12Vdc, requiring 100mA to energise the coil.
  The MCU must be electrically isolated from the 240V supply.
  The speed of the motor is expected to be controlled by a simple On/Off control strategy.
  (Note- The speed control is not a requirement in this assignment)

- The water inlet valve is to be controlled by a logic signal from a PIC port pin.
  A logic 1 opens the valve to fill the wash drum, a 0 closes the valve.
  The 0 to 5Volt logic signal is suitable to directly control the valve itself.

- The water outlet valve and pump are to be controlled by a logic signal from a PIC port pin.
  A logic 1 opens the valve and pumps the water out of the wash drum, a 0 closes the valve and turns off the pump. The 0 to 5Volt logic signal is suitable to directly control the valve, but the pump must be switched using a relay. The relay coil is rated at 12Vdc, requiring 100mA to energise the coil.

- The mass of clothes and water in the wash drum is measured by a sensor which outputs an **analog signal between 0 and 5Volts**. This voltage is directly proportional to the mass of the drum and its contents, where a full scale value of 5V indicates a mass of 10.5 kg. **The mass of the drum (2kg) is included in the measurement.**

- The speed of rotation of the drum is measured by a sensor which outputs an **analog signal between 0 and 4Volts**. This voltage is directly proportional to the (revolutions per minute) rpm of the drum, where a full scale value of 4V indicates a speed of 1100rpm. ***(Note – The speed control is not a requirement in this assignment)***

Software Requirements

The following routines must be written.

NOTE - You are NOT allowed to use any library functions provided with MPLABX XC8.
(With the exception that you may use the timing delay routines)

You must write your own routines to write/read the PIC special function registers to control the interfaces.

Part 1. Write a C **subroutine** called **init_ports** to initialise all the I/O ports you have chosen for your hardware design, to prepare ALL inputs and outputs ready for operation.

Part 2. Write a C **subroutine** called **init_adc** to initialise the ADC ready to read the measure an analog input. (see part 3 and 4).

Part 3. Write a C **function** called **mass** to read the ADC channel you have used (to 10 bit resolution), calculate and **return the mass of the clothes and water (in kg) as a float value**.

Part 4. Write a C **function** called **speed** to read the ADC channel you have used (to 10 bit resolution), calculate and **return the speed (in rpm) as an unsigned integer value**.

Part 5. Write a C **function** called **button** to read the status of all buttons and **return a single byte (char) whose value is 0 if no button is pressed, 1 for the BACK button, 2 for the UP button, or 3 for the DOWN button, or 4 for the ENTER button**. Assume only one button is active at any instant.

Part 6. Write a C **subroutine** called **status** that displays the status of machine, appropriate to your choice of indicators or display.

Part 7. Write a C **subroutine** called **motor** that accepts a parameter **m,** where m = 0 turns off the motor and m = 1 turns on the motor.

Part 8. Write a C **subroutine** called **inlet** that accepts a parameter **i,** where i = 1 turns the inlet on and i = 0 turns the inlet off.

Part 9. Write a C **subroutine** called **outlet** that accepts a parameter **o,** where o = 1 turns the outlet and pump on and o = 0 turns the outlet and pump off.

Part 10. Write a **main** C program which utilises the functions/subroutines above to initialise all ports and the ADC, then calls all the functions/subroutines in some logical sequence so that their functions can be fully tested on the simulator. The main program should include an endless loop containing a series of ~~other loops and~~ tests to verify each possible button press, status display output, test the controls for the motor, inlet valve, outlet valve/pump and test for the range of measured values for mass and speed.

This is to be achieved by testing values returned from input routines and using the results of those tests (or the values returned) to control outputs, via the output routines. Loops may be useful to sequence a series of tests.