Laboratory 08_09
C Language Stepper Motor

Objective: You will outline, flow chart and write a C-language program to drive the stepper motor. You will use C-language data type variables, assignment, control flow logic, bitwise logic and functions to interface with microcontroller I/Os 8-bit dip switches and the stepper motor on our training board.

Pre-Lab:
At start of lab session: You must provide proof that you have started and/or completed the Pre-Lab work. Do not program anything until you have completed Week 09_09 | Pre-lab assignment.

Read the instructions completely! Then use your completed Pre-Lab Program Development Cycle worksheet for C-Language steps 1-3 to program the following stepper motor control

Pre-Lab Check Point: At start of week 09 lab session: You must provide proof that you completed this week's Pre-Lab work.

Instructions and Program Description:
Design and then write a C language program to drive the stepper motor on our training board.

- The motor will have two speeds: fast and slow and two directions: forward or backward.
- Bits from the 8-bit dip switch connected to Port B are used to check the desired motor speed and direction.
  - If Port B bit 0 is high the motor should turn in a forward direction and if bit 0 is low the motor should turn in the backwards direction.
  - If Port B bit 7 is high, the motor should turn at the faster speed and if bit 7 is low it should turn at the slower speed.
  - Note: For the stepper motor to work, the Motor Enable switch must be set to the left position.

  <u>Stepper Motor operation details:</u>
  The stepper motor is controlled by bits 3-0 of Port D (address $05). To run the stepper motor forward (CW), the appropriate bits of Port D will need to be made output ports by writing ones to the proper bits (only the four bits 3-0) in the data direction register DDRD (address $07).
  To turn the stepper motor (Port D) through the appropriate phase excitation sequence, the following sequence must be used. Note: Each byte sent should step the motor one step. There must be a delay between each byte sent. (these values are determined by the hardware manufacturer and should not be changed.)

$05, $09, $0A, $06 (hex values)

Your program should have no global variables and at least three C language functions. One should step the motor and another should delay. A main function should call these two functions.

Function Descriptions:
Using the Program Development Cycle worksheet (C-language) you are to outline and define your program and each of the following function descriptions:

1. <u>A Step Motor Function</u> will determine and send one appropriate value to the stepper motor and then ends (release control back to the main function).
   - This function receives no values and returns nothing.

Laboratory 08_09
C Language Stepper Motor

- This function is called from main()
- Within the step motor function:
  - A character array defined with the values to control the stepper motor.
  - A variable to index into the array declared as a static variable (static char is fine) so that its value is retained between calls.
  - A C language *if* statement can be used in this function to check bit 0 of port B and execute a forward direction block of code if it is high and a reverse block of code if it is low.
    - if the motor is going in the forward direction block of code:
      - The index variable will be incremented and
    - if the motor is going in the reverse direction block of code:
      - The index variable will be decremented
    - Important: For both direction blocks of code: When the index is outside of the array limits, it must be reset back to the beginning or ending of the array.

2. A Delay Function will :
   - This function is called from main(). The main() function will determine and pass the appropriate delay value to this function.
   - The delay function receives a delay count that is type *unsigned int*.
   - In this delay function,
     - uses a C language *for* loop that loops the number of times of the received delay count.
     - and then ends (releases control back to the main function).

3. The default *main( )* Function must do the following:
   - Function *main* must call the step function and then call the delay function with an appropriate argument.
   - Before the call to the delay function: you will need to check bit 7 of port B, and if high, pass the value 30000_{10} and if low pass the value 60000_{10}.
   - The *main* ( ) function runs an endless *for* loop provided by CodeWarrior.
   - The *main( )* function must start the motor upon program startup and keep it going regardless of direction or speed until the program closes.
   - Write your program such that only code that requires continuous operation are in the endless for loop.

Do not program anything until you have thoroughly outlined the program operation flow.

Program Development cycle worksheet steps 1-3: Thoroughly outline and flow chart your program including each function's tasks. Pre-Lab Check Point: At start of lab session you will be checked off by TA or Instructor for these completed tasks. It is the only way you will get the pre-lab points.

Program Development Cycle worksheet steps 4-6: Write and Debug the program by stepping through it and examining registers. A small number such as one can be passed to the delay function while debugging. Once the code is debugged, the delay counts can be increased, and the processor run at speed. The motor should not miss-step when reversing direction.

The program should use standard C language conventions, follow code organization format as outlined in lecture and posted examples, and be fully documented, including code comments and commented header.

Submit: your complete Program outline including all Program Development cycle worksheet items in one well formatted word .docx., your projects main.c file and your entire zipped project folder.