Design Manual

Algorithm Descriptions

The main algorithms of the project are as follows:

Read_Timer_Input:

When in entry mode, the microwave will take input for the timer. To handle the shifting of numbers from seconds to minutes as more inputs are given, we wrote the Read_Timer_Input function. The current number of inputs is held in a register, and is incremented/cleared at the right times. When an input is given, before incrementing the counter, the algorithm checks how many inputs have been given and reacts accordingly:

If it is the first input, it is stored in the Seconds register.

If it is the second input, multiply the Seconds register by 10 and add the input number.

If it is the third input, extract the 10's from Seconds and place it into Minutes, then multiply Seconds by 10 and add the input to it.

If it is the fourth input, multiply Minutes by 10, extract the 10's from Seconds and add it two Minutes, then multiply Seconds by 10 and add the input.

Print_Screen:

Print_Screen has 3 responsibilities; displaying the timer, rotation symbol, and open/closed status on the LCD screen.

It does so in the following way:

It first uses the function Print_2_Digit_Number to print the Minutes, then prints a ':' and uses the Print_2_Digit_Number again to print the Seconds.

It then adds the appropriate spaces to shift the cursor to the top right hand corner of the screen, and fetches/prints the current rotation symbol of the turntable.

It then shifts the cursor to the next line using the next line command, shifts the cursor again to the bottom right corner of the screen, and fetches/prints the current status of the door, "O" or "C".

Print_2_Digit_Number:

This function prints a 2 digit number to the LCD screen. It does so by repeatedly subtracting 10 from the given number , keeping a count of how many subtractions have been done (Q, the quotient). Once the number (R, remainder) is less than 10, the quotient is printed to the screen, and then the remainder.

Add_To_Seconds:
This function adds a given number of seconds to the Minutes:Seconds timer. It does so by adding the input number to the current seconds, and reacting accordingly to the resultant number.

If adding to the timer, when an add is performed and the resultant is greater than 59 seconds, increment the minutes (provided Minutes < 99) and set the Seconds to be the remaining seconds.

If subtracting from the timer, when the subtract is performed and the resultant is less than 0, we decrement the minutes (provided Minutes > 0) and set the seconds to be the remaining seconds.

Data Structures

AVR registers are used to store information that is often required such as:

- Current mode of operation
- Number of inputs given
- Debounce flag

Other important information is stored in DSEG, alongside the Timer0 counters:

- 1 byte of storage for minutes
- 1 byte of storage for seconds
- 1 byte of storage for the current rotation symbol
- 1 byte of storage for the current opened/closed status
- 1 byte of storage for the current power level

Module Specification

If this project were split into multiple modules, they would include:

- A module for printing to screen:
    - This module would encase all functions/macros relation to printing to screen. Functions related to printing often grow very large due to their nature of using repeated macros. This causes fails to branches and jumps when too many macros are used, and splitting them off into another module would stop that problem.
    - This module would also encase character building functions for the LCD screen.

- A module for reading in the keypad input:
    - This module would encase all functions/macros related to reading digits input from the keypad (while in entry mode)
    - It would also handle manipulation of Seconds/Minutes when input is made.

- The main microwave module:
    - This would handle microwave operations, like reacting to keypad presses such as stop, start and quick add/subtract etc.
    - It would pass digit handling and arithmetic to the previous module, and printing output to the first module.

System Flow Control