**DATA STRUCTURES**

The data structures used in the project comprise of a number of variables stored in data memory. All variables are 1 byte in size.

Firstly, it uses variables to obtain input from the keypad, namely row, col, rmask and cmask. Row and col allow specify which specific key is being pressed, whereas rmask and cmask act as masks for extracting the value of the pin of the key that is being pressed.

A variable doorIsOpen is used to specify whether the door to the microwave is currently open or closed. Interrupts and timers will ignore any input if this variable is set to 1, meaning that the door is open.

A mode variable keeps track of which mode the device is currently operating in. Entry mode refers to 1, power selection is 2, running mode is 3, paused mode is 4 and finished mode is 5.

A currentInput variable keeps track of the latest key being read from the keypad, to be handled later. The reading of the keypad and handling its input occurs in the main program loop.

The variable power specifies which power level the device is currently set to. It is set to 100 for 100%, 50 for 50% and 25 for 25%.

In entry mode, four variables, a, b, c and d, are used to keep track of the digits currently entered into the program, corresponding to the thousands, hundreds, tens and ones digit respectively. As digits are entered in entry mode, these variables will be populated accordingly. These digits are reset every time the program re-enters entry mode from paused or finished mode.

When the program is entering running mode from entry mode, these four variables a, b, c and d will have their representative time converted into minutes and seconds, to be stored into two registers, mins and secs. Since mins and secs are used regularly in running mode for displaying the time, they are assigned registers rather than a space in data memory. Running mode uses these registers rather than the four variables. See <<<<<<<<<<SECTION HERE>>>>>>>>>>>> for a more indepth discussion.

Once the program enters running mode, a variable functionsRunning will be set to be true. This variable refers to the fact that functions, such as the magnetron and turntable are on. The variable resets every time operation is stopped completely, such as when going to finish or back to entry mode.

A number of counters are used to store the number of counts of a specific time period. The counter counts the raw signal from the prescaled clock frequency. The counter250 specifies how many 250 ms has passed from the last time it was reset. The other 3 timers have a similar purpose. See <<<<<<<<<SECTION>>>>>>> for more information.

Two variables, magnetronCounter and magnetronRunning, are used to determine the frequency and behavior of the magnetron. The variable magnetronCounter determines at which time intervals it should turn on or off, while magnetronRunning determines whether it should turn on or off.

The index variable specifies which character will be displayed as the turntable on the LCD screen. Further, the CCWrotation variable determines which direction the turntable is currently rotating. This direction alternates every time it re-enters running mode, from any other mode. When CCWrotation is 1, the turntable is rotating counter-clockwise, and vice versa.

Note that an open door state is not interpreted as a separate mode, but rather just as a state where no input can be made.

Two macros, getVar and setVar, allow access of these variables easily.

; getVar <VARIABLE LABEL> <REGISTER>

.macro getVar

ldi XH, high(@0)

ldi XL, low(@0)

ld @1, X

.endmacro

; setVar <VARIABLE LABEL> <REGISTER>

.macro setVar

ldi XH, high(@0)

ldi XL, low(@0)

st X, @1

.endmacro

**ALGORITHMS**

The program initializes at RESET, where various ports, timers and LCD are initialized, variables are set to their initial value and mode is placed in entry mode. The main program loop reads keypad input and handles the button being pressed depending on the mode and whether the door is open or not.

The program comprises of the following interrupts and timers. INT0 is used as an interrupt to trigger the event where the door was closed, while INT1 is used to trigger the door being opened. TIMER0 is used to handle running mode. It decrements the time shown on the display, updates the turntable, and handles the magnetron.

.cseg

.org 0x0000

jmp RESET

.org INT0addr

jmp CLOSE\_DOOR

.org INT1addr

jmp OPEN\_DOOR

.org OVF0addr

jmp TIMER

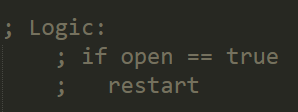
**Handling keypad input**

At convert\_end, the keypad input has been interpreted and the input must now be handled. Firstly, the program checks to see if the door is open. If it is, the input is not handled as it jumps to end and restarts the main loop.

; Logic:

; if open == true

; restart



Otherwise, the program will clear the display and check which mode it is currently running in. Then it will branch to that section of code that handles input according to the mode.

; Branch

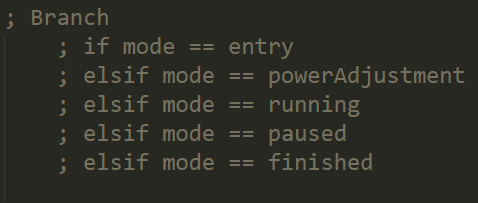
; if mode == entry

; elsif mode == powerAdjustment

; elsif mode == running

; elsif mode == paused

; elsif mode == finished



At the end of each mode, the program will check for debouncing, and, once the key does not register as being pressed anymore, it will jump back to the start of the loop and start scanning for a keypad press again.

**Entry mode**

If the mode is entry mode, then the program will allow input into variables a, b, c and d, the time variables, if a digit was pressed, and it will clear the time if a hash was pressed. If A was pressed, it will switch the mode over to the power adjustment mode, while if an asterisk was pressed, the running mode will start. Appropriate text will be displayed on the LCD when the button is pressed.

; Entry mode

; if 0-9

; insert digit to the time

; display updated time

; if #

; clear time

; display cleared time

; if A

; mode = powerAdjustment

; display "Set Power 1/2/3"

; if \*

; if time entered

; mode = running

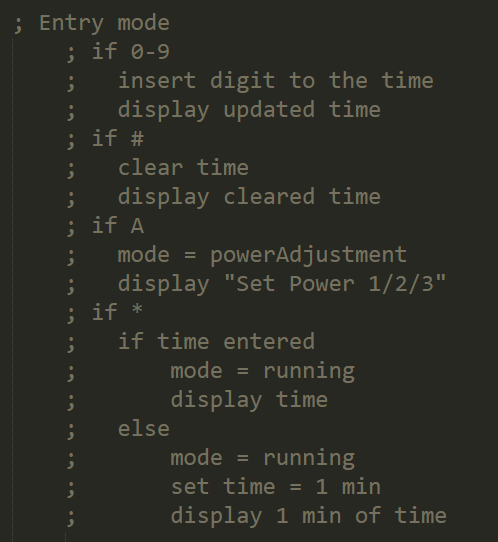
; display time

; else

; mode = running

; set time = 1 min

; display 1 min of time



**Power selection mode**

In power selection mode, the keys 1, 2 and 3 update the power variable accordingly. The A button will cause mode to be shifted back to entry, with the digits a, b, c and d re-displayed on the LCD.

; Power Selection Mode

; elsif mode == powerAdjustment

; if 1

; power = 100

; if 2

; power = 50

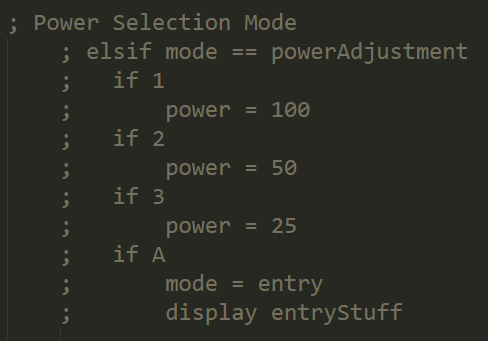
; if 3

; power = 25

; if A

; mode = entry

; display entryStuff



**Running Mode**

If the mode is running, C and D will cause 30 seconds to be added to and subtracted from the mins and secs registers respectively. If the asterisk was pressed, 1 minute will be added instead, and if hash is pressed, the mode will be placed in paused mode.

; Running Mode

; elsif mode == running

; if C

; add 30s to time

; if D

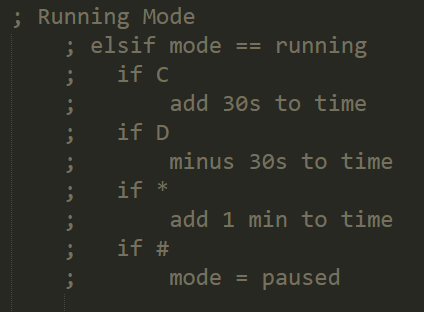
; minus 30s to time

; if \*

; add 1 min to time

; if #

; mode = paused



**Paused mode**

If the mode is paused instead, then asterisk will switch the mode to running, and hash will switch the mode to entry.

; Paused mode

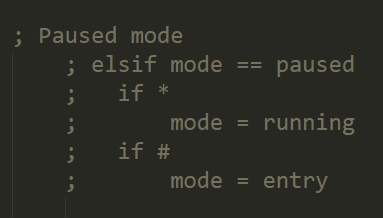
; elsif mode == paused

; if \*

; mode = running

; if #

; mode = entry



**Finished mode**

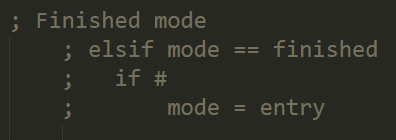
In finished mode, if the hash was pressed, then mode will be set to entry. Note that pressing the door in finished mode will also cause the mode to be set to entry, however this is not handled here, but rather in the open door interrupt.

; Finished mode

; elsif mode == finished

; if #

; mode = entry



**Functions**

The printDigitsToLCD function will, in entry mode, display the correct time, turntable, group number and door state.

On the other hand, the printTimeToLCD function will, in running mode, display the correct time, turntable, group number and door state.

The display\_numbers macro in actuality calls a checkTensDigit function which splits the current time held in the mins and secs registers into digits to be displayed on the screen.

The clearTimeAndDigits function will clear the a, b, c and d time variables used in entry mode as well as the mins and secs registers used during running mode.

The turntable function will, according to the current character it should be displaying, specified by the index variable, display that character on the display

The updateIndex function will, when called, either increment or decrement the index variable between the bounds of 0 and 3 inclusive according to whether the CCWrotation variable is set or not. The turntable will turn counter-clockwise if the CCWrotation variable is set to 1.

The updateTime function will decrement 1 second from the time represented by the mins and secs registers and print this on the display.

The updateMagnetron function, when called, will check which segment of the second has been reached, and then therefore will decide whether to turn the magnetron on or off. It will then set the magnetron to be on or off.

The writePowerLabels function will display “Set Power 1/2/3” as well as the turntable, group number and door state.

The writeFinishedText function will display “Done\nRemove Food” on the display.

The writeSecondLineToDisplay function will display the group number and door state to the display. This is used in the other functions that require writing the group number and door state.

The writeDoorStateToDisplay will write either a “C” or “O” depending on whether the door is open or not, effective immediately.

The toggleRotationDirection will toggle the value of CCWrotation by using an exclusive OR.

The setDoorLEDClosed and setDoorLEDOpen functions will cause the top LED to be on or off.

The setMotorOn, setMotorOff, resumeMotor and pauseMotor functions will turn the motor on or off. The difference between the first two and last two are how they handle the behavior of the motor when it is turned on again. When the motor is set to off using setMotorOff, it will begin again from the start of its cycles rather than where it left off when it was paused, which can be achieved by using pauseMotor.

The finished function will set the mode to finished, reset all counters, and turn off the magnetron and turntable.

**Special algorithms**

The add30Secs algorithm

