**3-2 Assignment: Designing Your Code**

Paul Mauviel

paul.mauviel@snhu.edu

Southern New Hampshire University

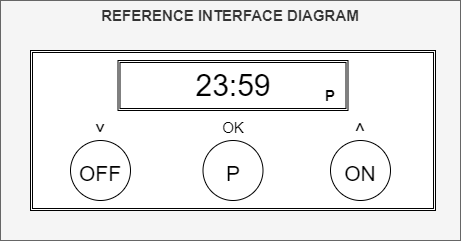
CS-210: Programming Languages

Professor Eric Gregori

January 22, 2022

Electric Coffee Maker

I’ll be using this reference diagram to inform the choices made in user experience. The software is designed as a simple state machine that transitions from one state to the other.



# Reflection

Many pains have been taken to ensure that my program flows in a logical order so I would say that yes, my program flows in a logical order. That being said, there was unspecified criteria on which I applied some personal interpretation. The best example is what the user interface was to be to set the time. The specification only describes the device having three buttons so I imagined an interface as illustrated above.

I left little to no variables left unstated. That being said, they were:

1. Current Time global variable
2. Current State global variable
3. Program Time global variable

In addition to these there are various local variables used through the various functions that I won’t state here.

In the Pseudocode section that follows, you will see that I organized the application as a set of functions called from main. Therefore, the state functions are called from main. The state functions being:

1. Run On
2. Run Brew
3. Run Program
4. Run Set Time (accepts a parameter)
5. Run Idle

If I had to choose, I would say I prefer pseudocode over flowcharts. While rarely created formally like this, pseudocode is often used in to form of comments while blocking out large portions of functionality. I find flowcharts less useful in my day-to-day work and more time consuming.

# Pseudocode

FUNCTION Change State

PARAMETERS: New State

CALL FUNCTION Leave State

CALL FUNCTION Enter State with parameter New State

END FUNCTION

FUNCTION Leave State

IF global Current State IS BREWING

Turn off heating element

SET global Program Time to null

END IF

SET global Current State to null

END IF

FUNCTION Enter State

PARAMETERS: New State

SET global Current State to New State

END FUNCTION

FUNCTION main

SET global Current Time to 00:00

SET global Current State to null

SET global Program Time to null

CALL FUNCTION Change State with parameter IDLE

// We loop forever since a coffee maker should always be working while the power is running

LOOP FOREVER

IF Current State IS IDLE

CALL FUNCTION Run Idle

ELSE IF Current State IS PROGRAM

CALL FUNCTION Run Program

ELSE IF Current State IS ON

CALL FUNCTION Run On

ELSE IF Current State IS BREWING

CALL FUNCTION Run Brew

ELSE IF Current State IS SET\_TIME

CALL FUNCTION Set Time with parameter SET\_CLOCK

ELSE IF Current State IS SET\_PROGRAM\_TIME

CALL FUNCTION Set Time with parameter SET\_PROGRAM

ELSE // Bad value, go back to Idle

CALL FUNCTION Change State with parameter IDLE

END IF

END LOOP

END FUNCTION

FUNCTION Run On

SET First Press Time to Current Time since epoch

SET Boolean timingOut to True

SET Next State to null

LOOP IF timingOut

IF BUTTON\_PRESSED IS ON

SET timingOut to False

SET Next State to SET\_TIME

ELSE IF BUTTON\_PRESSED IS OFF OR BUTTON\_PRESSED IS PROGRAM

SET timingOut to False

IF Program Time IS null

SET Next State to IDLE

ELSE

SET Next State to PROGRAM

END IF

ELSE

IF time since epoch – First Press Time >= 0.5 seconds

SET timingOut to False

SET nextState to BREWING

END IF

END IF

END LOOP

CALL FUNCTION Change State with parameter Next State

END FUNCTION

FUNCTION Run Idle

SET global Program Time to null

SET Boolean IsIdling to True

SET Next State to null

LOOP IF isIdling

Update Current Time

Display Current Time

IF BUTTON\_PRESSED IS Program

SET isIdling to False

SET Next State to SET\_PROGRAM\_TIME

ELSE BUTTON\_PRESSED IS ON

SET isIdling to False

SET Next State to ON

END LOOP

CALL FUNCTION Change State with parameter Next State

END FUNCTION

FUNCTION Run Program

SET Boolean isProgrammed

SET Next State to null

LOOP IF isProgrammed

Update Current Time

Display Current Time with P icon to indicate a program is set

IF BUTTON\_PRESSED IS OFF

SET isProgrammed to False

SET Next State to IDLE

ELSE IF BUTTON\_PRESSED IS PROGRAM

SET isProgrammed to False

SET Next State to SET\_PROGRAM\_TIME

ELSE IF BUTTON\_PRESSED IS ON

SET isProgrammed to False

SET Next State to ON

ELSE

IF Current Time >= Program Time

SET isProgrammed to False

SET Next State to BREWING

END IF

END IF

END LOOP  
 CALL FUNCTION Change State with parameter Next State

END FUNCTION

FUNCTION Run Brew

SET Boolean isBrewing to True

SET Next State to null

LOOP IF isBrewing

Update Current Time

Display Current Time

IF BUTTON\_PRESSED IS OFF

SET Next State to IDLE

SET isBrewing to False

ELSE

READ Tank Water Level > 0 AS Tank Has Water

IF Tank Has No Water

READ Coffee Temperature

IF Coffee Temperature is above 185 degrees Fahrenheit

Turn Off Heating Element

ELSE

Turn On Heating Element

END IF

ELSE

READ Tank Water Temperature > 212 AS Tank Water Is Boiling

IF Tank Water Is Not Boiling

Heat Water Tank

ELSE

Pour some water

END IF

END IF

END LOOP

CALL FUNCTION Change State with parameter Next State

END FUNCTION

FUNCTION Set Time

PARAMETERS: TIME\_SET\_MODE

SET Hours to Current Time Hours

SET Minutes to Current Time Minutes

SET CurrentlyUpdating to H

SET isSettingTime to True

SET Next State to null

LOOP IF isSettingTime

Update Display to Hours:Minutes

SET ValueChange to 0

IF BUTTON\_PRESSED IS ON

SET ValueChange to 1

ELSE IF BUTTON\_PRESSED IS OFF

SET ValueChange to -1

ELSE IF BUTTON\_PRESSED IS PROGRAM

IF CurrentlyUpdating IS H

SET CurrentlyUpdating to M

ELSE

SET isSettingTime to False

IF TIME\_SET\_MODE is SET\_CLOCK

SET global Current Time to Hours:Minutes

IF Program Time is null

SET Next State to IDLE

ELSE

SET Next State to PROGRAM

END IF

ELSE

SET global Program Time to Hours:Minutes

SET Next State to PROGRAM

END IF

END IF

END IF

IF ValueChange IS NOT 0

IF CurrentlyUpdating IS M

SET Minutes to Minutes + ValueChange

ELSE

SET Hours to Hours + ValueChange

END IF

END IF

END LOOP

CALL FUNCTION Change State with parameter Next State

END FUNCTION