20COA202 Coursework

B912997

Semester 2 2020-2021

1 Introduction

Enhanced with Query and Memory extensions.

The code will start off on the Home Menu. This will give you options for Ground floor, First floor and Stored Data which you can scroll through with up and down.

If you select ground floor or first floor, you will be able to scroll through the menus showing each room on the floor. If you select stored data, you will receive all the data stored in EEPROM on the serial monitor in the form of the output required on the specification. Once you select a room, you can have options between Heat and light. Once you select one of those, you have the option to change the Level or add a time on which to either turn the system on or off. Once you select you will be returned to the home menu.

You can also see the stored data on EEPROM by entering Q with newline on the Serial Monitor. This is also like if you enter M with newline on the serial monitor, it will return data including the amount of RAM unused.

1. Click SELECT or DOWN button to access the home menu.
2. You can choose from 3 options, ground floor, first floor, or stored data.
3. Navigate by clicking UP and DOWN buttons.
4. Ground floor will get you into ground floor menu.
5. First floor will get you into first floor menu.
6. Stored data will print out all the values and states of every heater and light and their time ON or OFF.
7. You can choose with SELECT button.
8. If you choose Stored Data, you will print the values to the Serial monitor and you will be sent back to the main menu again to choose from the first 3 options.
9. You can navigate with UP DOWN and SELECT to your desired room and type of item.
10. When you choose the type (e.g. kitchen heater) you will be presented with 3 options.
11. Level will allow you to change the brightness / temperature of the desired room.
12. ON Time will allow you to set a time which the heater/light will be turned on
13. OFF Time will allow you to set a time which the heater/light will be turned OFF
14. If you choose Level, you will be able to change the value between 0 and 100 and it will remain the same until changed by the user. You can increase and decrease by clicking LEFT and Right buttons. Click SELECT to go back to the main menu.

15. If you choose ON or OFF time you will be able to change the first value which is HOURS from 0 to 23 then click SELECT

16. Now you will change the values of MINUTES from 0 to 59

17. You can navigate through the hours and minutes by clicking UP and DOWN button on each side and click SELECT to finalize your choice and go back to the main menu.

2 Your code - base implementation

2.1 Data structures

Variables and their types:

int chooseFloor = 0;

This number is for the Home menu switch case.

char button_state;

This is for reading the buttons.

int count = 0;

This is for scrolling up and down. If increased, menu moves to next option. If decreased by moving up, menu goes to previous option.

int count1 = 0;

This is for counting increase or decrease on hours for Time.

int count2 = 0;

This is for counting increase or decrease on Minutes for Time.

bool activateButton = false;

This allows the right and left buttons to only be in use when increasing or decreasing level.

bool activateScroll = false;

This is to differentiate use of the up and down buttons for when ure scrolling through menus and not for anything else.

bool hourSelect = false;

This is to only be able to use up and down buttons for increasing and decreasing the count1 when hours is being selected.

```
bool minuteSelect = false;
```

This is to only be able to use up and down buttons for increasing and decreasing the count1 when minutes is being selected.

```
int menuNumber = 0;
```

This is to store a number for the option on the menu without actually going to that menu.

```
int menuSelected = 0;
```

This is to actually go to the Menu which has been selected.

```
int chooseRoom = 0;

int chooseRoom1 = 0;

int chooseItem1 = 0;

int chooseItem2 = 0;

int chooseItem3 = 0;

int chooseItem4 = 0;

int chooseItem5 = 0;

int chooseItem6 = 0;
```

These are the variables used in the switch cases for different room and floor menus.

```
int levelControl[12];
```

This is to make an array of the Levels for every room heat and light.

```
int chosenElement;
```

This is for knowing the address of the data for which a certain rooms level number is stored.

int chosenElement2;

This is for knowing the address of the data for which an on and off information about a room is stored. You can also find the store of the Hours and Minutes by adding one and two respectively.

int chooseOption;

This is to choose between the options of picking Level, On or Off time for a certain rooms heat or light.

int Hour;

This is for displaying and saving the hour on an ON or OFF.

int Minute;

This is for displaying and saving the minute on an ON or OFF.

int ON;

this is for saving whether the time is for an ON or an OFF. If its 0 it means OFF, if its 1 it means ON.

int mark = 0;

This is to stop the code for time not working as the pervious select can cause the HourSelect to be set as false without running this code.

Functions:

freeMemory(): returns the amount of free memory.

allMenus(): has the entire Menus and functions so they can be called repeatedly and easily.

homeMenu(): Menu showing floors and storedData options.

groundFloorMenu(): Menu showing rooms on ground floor.

firstFloorMenu(): Menu showing rooms on first floor.

Kitchen(): gives option for changing heat or light for kitchen.

Hall(): gives option for changing heat or light for hall.

livingRoom(): gives option for changing heat or light for living room.

bedRoom1(): gives option for changing heat or light for first bedroom.

bedRoom2(): gives option for changing heat or light for second bedroom.

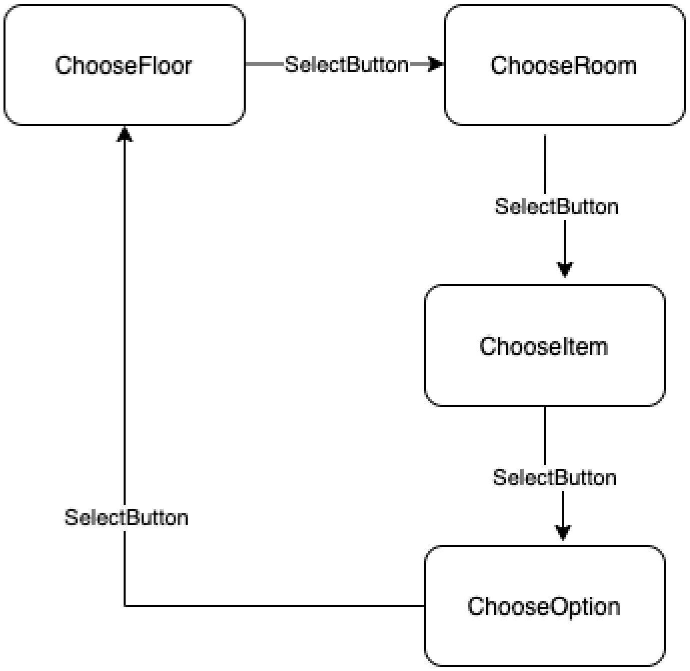
storedData(): shows all stored data on serial monitor.

options(): Lets you pick between changing Level, On time or Off time.

Level(chosenElement): Lets you change levels and displays it as it changes.

Time(): lets u pick the time u want it to turn On or OFF.

2.2 FSMs


ChooseFloor
SelectButton
ChooseRoom
SelectButton
ChooseItem
SelectButton
ChooseOption
SelectButton

2.3 Testing and testing

I would check the menu and check if the buttons clicked are doing what they are supposed to do if the button failed to do as expected. I would trace the activation of the button and follow the steps to where it would go. I would also add a Serial.print() inside a button to see if it would print a statement to the serial monitor or not. If it, did not it mean that I had a problem earlier on with the button activation.

Other debugging methods I followed. Is printing a counter to check if it is going out of the supposed range or if it would count wrong. Or maybe it would carry out another value from previous menu or function. And I made sure to reset the counter every time it is needed so it would always have a fresh start that cannot go wrong.

I would also use JQuery to inspect the elements and values of the saved integers that are saved in EEPROM, it would save me a lot of time to be able to see all values after every input change rather than go and check every menu element.

When a piece of code stopped working. I would comment out relevant or suspected code lines that might affect the code that stopped working. It helped me in many ways to quickly identify where the issue is accruing and what might be causing it.

I would also comment out parts of code that I suspect to be useless and see how the program acts without them. I was able to shorten the program by approximately 300 lines doing so and I was able to figure out ways to implement multiple actions in one function.

I tried each possible value that can be set in every saved value to make sure they are doing what they were supposed to be doing, I would also check on them after any large piece of code to make sure it did not cause a value to have unwanted changes.

3 Extension Features

3.1 LAMP

NOT IMPLEMENTED

3.2 OUTSIDE

NOT IMPLEMENTED

3.3 QUERY

I included a simple code that monitors entry into the serial monitor and checks if it is equal to “Q\n” to then show all stored data on serial monitor.

The main functions it uses to check for and take the input is Serial.available to see if characters come on Serial Monitor and Serial.readString to take the input in as a string.

3.4 MEMORY

This is very similar to Query.

Once FreeMemory() is created, it is very similar to Query as just instead of “Q\n”, you check for “M\n” and then print what is returned from freeMemory().

3.5 SOFT

NOT IMPLEMENTED

3.6 EEPROM

NOT IMPLEMENTED

4 Conclusions

Everything in BASIC and the extensions Query and Memory is full working. There is nothing that is only partially working. I put a lot of effort into making sure everything runs smoothly and no errors are present which could cause an unnatural option in the code to occur.