Development report

**Introduction:**

The project source code contains 7 primary functions and one utility function. In this document, I will explain in details the logic behind each function and the sequence of program.

**Functions:**

The source code has seven 8 functions, as follow:

1. void main(): contain the main logic of the program.
2. int coldDrink(): contains the logic of serving cold Drink for the customer and return the price of selected drink.
3. int hotDrink(): contains the logic of serving hot Drink for the customer and return the price of selected drink.
4. void handlePayment(int amount): contains the logic of handle payment with specific required cost amount.
5. void returnAllMoney(): contains logic of returning the money to customer in case of time out.
6. void interrupt (): mikroc pro reserved function to handle timer TMR0 interrupt handler.
7. void delay(): simple function implement the delay logic.
8. char\* strConstCpy(const char\* ctxt): utility function to cast the string stored in Rom to be represented in the LCD. This function is important because we need to store all our string in ROM not in RAM , so we overcome storage limitations.

In the following sections, I will explain each function in detail and will draw the flowchart for the function if required.

**1-Void Main()**

This function contains the main logic for the program. First , the program shows the Main menu asking the user to select cold or hot drink. The function makes endless loop and waiting the input from the user. If the user presses RD0 or RD1, the program will highlight first or second row respectively in the LCD. Once the user press RD2 for selection, the function will determine which selection the user selects and will call “coldDrink()” or “hotDrink()” function respectively. These functions will show the drink menu for the user and the user will select the drink and will return the cost of selected drink. Once the control return back to the main() function , the function will call “handle payment” and will pass the cost to it. Once the “handle payment” function return, the main() function will show the menu again and return to the beginning.

Here is the flow chart for the function:

Cold drink

Hot drink

RD2 pressed ?

yes

No

selection

hotDrink()

coldDrink()

Handle payment

Handle up,down selection

Start

Show cold,hot menu selection

**2-int coldDrink():**

simple function that show the menu of cold drinks for the customer and will make endless loop waiting for user input. The LCD menu will go up and down if the user press RD0 or RD1. If the user press RD2 , message will shown to the customer to print the choice and price of drink in the LCD. The function will return the cost of the selected drink.

Here is flowchart for the function:

yes

RD2 pressed ?

start

Show orange juice,fizzy drink,water menu

Handle up,down selection

No

Show price

Return price

**3-int hotDrink():**

simple function that show the menu of hot drinks for the customer and will make endless loop waiting for user input. The LCD menu will go up and down if the user press RD0 or RD1. If the user press RD2 , message will shown to the customer to print the choice and price of drink in the LCD. The function will return the cost of the selected drink.

Here is flowchart for the function:

Return price

No

yes

Show price

Handle up,down selection

Show tea,coffee,chocolate,soup menu

start

RD2 pressed ?

**4-void handlePayment(int amount)**:

this function makes endless loop waiting for input from customer as coins. During that time, the function will look for timeout condition. if timeout has occur, the function will display message to the customer and will call “returnAllMoney()” to return all inserted coins to the customer. The function will keep watching also for all inserted coins and if reach to the required “amount” parameter, it will inform the user that money has inserted and will serve the drink to the user.

The function uses paymentFlag variable to tell the “interrupt” function that we are interested on the “payment time out” event. The “PaymentTimeOut” variable is Boolean variable that will be switched on by “interrupt” function if time out has occurred. Because of this, the function will make endless loop looking for the “PaymentTimeOut” variable. Once the “PaymentTimeOut” become true, the function will return back the money and return.

The function also uses variable “totalAccumulatedPayment” to monitor how much money has inserted. This variable will increase if RD0,RD1,RD2,RD3 or RD4 has pressed accordingly. Once that variable has reach to “amount” variable , the function will return back the excess and will serve the drink and print message to the customer, then will return.

The return excess function will switch on the LED once for every 5 pennies. The following loop:

for(loopVar=amount;loopVar<totalAccumulatedPayment;loopVar+=5)

will take care of returning the excess and will switch on the LED once for each 5 pennies.

Here is flowchart of the function:

Yes

return

start

Return excess

No

Collect enough money?

Yes

No

Accept coins

returnAllMoney

Show time out message

return

Timeout?

Show Drink dispensed message for 3 seconds

**5-void returnAllMoney():**

this is very simple function, it will enable the LED in RD7 and will wait for 5 seconds and then will switch off the LED.

The function uses “returnMoneyFlag” variable to tell the “interrupt()” function that we are interested in the “retuen money timeout” event. The “returnMoneyTimeOut” variable is Boolean variable which will be set from the “interrupt” function. Our function will make endless loop until this variable switched to true.

Flowchart of the function.

Yes

No

Switch off LED

delay

Timeout ?

Switch on LED

Enable timer

**6-void interrupt ():**

this function is reserved function that the compiler will call after calling the delay function. This function is used to implement the timer with TMR0. We have three timers in the program that we care about. The first timer is “payment timer”. During the “handlePayment()” function, we will take care about the payment timeout. The second timer is “return money timer” which is used in the “returnAllMoney()” function to take care about the timer of returning money. The third timeout is “drink dispensed message timer” which is used time out the message when the drink has served for the customer.

For each timer , I use three different variables, the flag,timeout and TimeOutCounter.

The flag variable will be set from the function that is interested in the timer. Once the interrupt function call , the interrupt function will check which flag has set on , so the function know which timer is being interested.

Because every timer has different amount, I will use the “TimeOutCounter” variable to keep track how many time the interrupt has been called. From trial and error , I have examined that every 20 times the interrupt has been called is equivalent to one second. So if we are interested in 5 seconds timer , we will check “TimeOutCounter” to be 100.

The “timeout” flag will be set from the interrupt function to indicate that the time out event of specific timer has occurred.

**7-void delay():**

simple function implement the delay logic.

**8-char\* strConstCpy(const char\* ctxt):**

utility function to cast the string stored in Rom to be represented in the LCD. This function is important because we need to store all our string in ROM not in RAM , so we overcome storage limitations.

All variables with “const char\*” data type will be saved in the ROM. We use this function to cast the variable to be printed in the LCD.