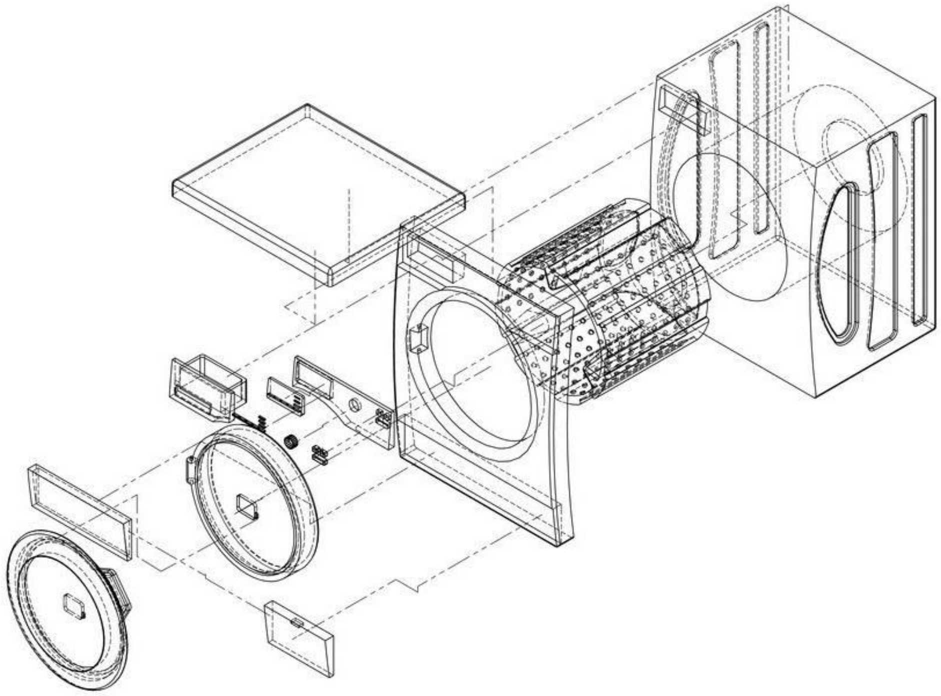
ESP Design Document



12-2019

Hussam Ayoub 2953536

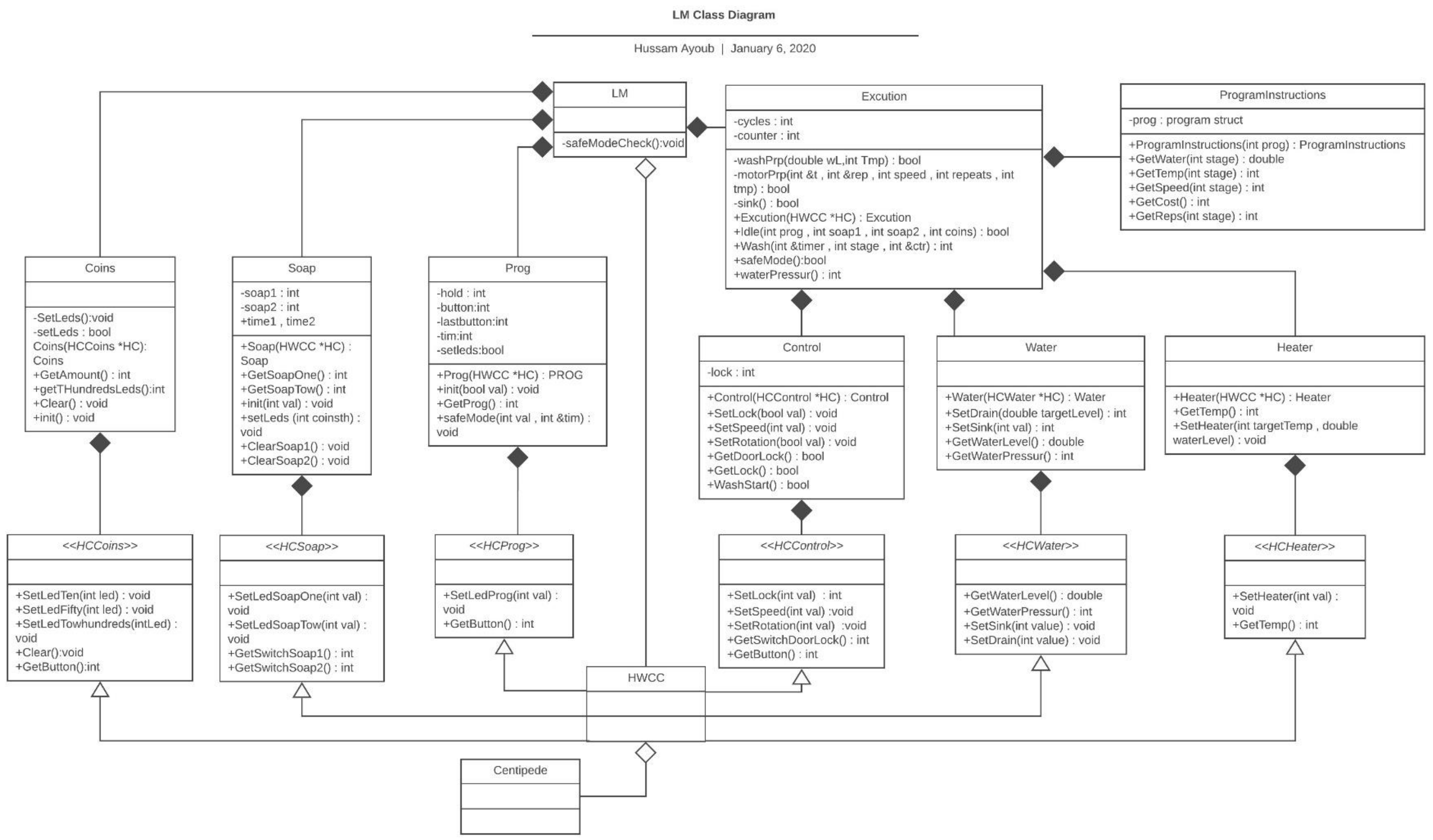
Table of Contents

Introduction:

This document describes diagrams, and the methods used to create a laundry machine software running on Vetetronics simulator.

Class Diagram:

Below is the UML class diagram of the laundry machine software and following a short description of classes and functions.


LM Class Diagram
Hussam Ayoub | January 6, 2020
LM
-safeModeCheck():void
Excution
-cycles : int
-counter : int
-washPrp(double wL,int Tmp) : bool
-motorPrp(int &t , int &rep , int speed , int repeats , int tmp) : bool
-sink() : bool
+Excution(HWCC *HC) : Excution
+Idle(int prog , int soap1 , int soap2 , int coins) : bool
+Wash(int &timer , int stage , int &ctr) : int
+safeMode():bool
+waterPressur() : int
ProgramInstructions
-prog : program struct
+ProgramInstructions(int prog) : ProgramInstructions
+GetWater(int stage) : double
+GetTemp(int stage) : int
+GetSpeed(int stage) : int
+GetCost() : int
+GetReps(int stage) : int
Coins
-SetLeds():void
-setLeds : bool
Coins(HCCoins *HC): Coins
+GetAmount() : int
+getTHundredsLeds():int
+Clear() : void
+init() : void
Soap
-soap1 : int
-soap2 : int
+time1 , time2
+Soap(HWCC *HC) : Soap
+GetSoapOne() : int
+GetSoapTow() : int
+init(int val) : void
+setLeds (int coinsth) : void
+ClearSoap1() : void
+ClearSoap2() : void
Prog
-hold : int
-button:int
-lastbutton:int
-tim:int
-setleds:bool
+Prog(HWCC *HC) : PROG
+init(bool val) : void
+GetProg() : int
+safeMode(int val , int &tim) : void
Control
-lock : int
+Control(HCControl *HC) : Control
+SetLock(bool val) : void
+SetSpeed(int val) : void
+SetRotation(bool val) : void
+GetDoorLock() : bool
+GetLock() : bool
+WashStart() : bool
Water
+Water(HCWater *HC) : Water
+SetDrain(double targetLevel) : int
+SetSink(int val) : int
+GetWaterLevel() : double
+GetWaterPressur() : int
Heater
+Heater(HWCC *HC) : Heater
+GetTemp() : int
+SetHeater(int targetTemp , double waterLevel) : void
<<HCCoins>>
+SetLedTen(int led) : void
+SetLedFifty(int led) : void
+SetLedTowhundreds(intLed) : void
+Clear():void
+GetButton():int
<<HCSoap>>
+SetLedSoapOne(int val) : void
+SetLedSoapTow(int val) : void
+GetSwitchSoap1() : int
+GetSwitchSoap2() : int
<<HCProg>>
+SetLedProg(int val) : void
+GetButton() : int
<<HCControl>>
+SetLock(int val) : int
+SetSpeed(int val) :void
+SetRotation(int val) :void
+GetSwitchDoorLock() : int
+GetButton() : int
<<HCWater>>
+GetWaterLevel() : double
+GetWaterPressur() : int
+SetSink(int value) : void
+SetDrain(int value) : void
<<HCHeater>>
+SetHeater(int val) : void
+GetTemp() : int
HWCC
Centipede

HCCoins, HCSoap, HCProg, HCControl, HCWater, HCHeater: are abstract classes designed as base classes for HWCC. All the function in the previously mentioned classed are pure virtual functions implemented in HWCC “hardware control class”.

The goal of this design is to separate objects based on hardware to limit hardware access of other objects.

HCCoins Functions:

SetLed..(int led): used to set the LED’s of coins tray in the simulator the passed variable “int led” is used to pass the number of LED’s the need to be set for each 10, 50, and 200 cents.

HCSoap Functions:

*SetLedSoap..(int val):*Sets Soap led on or off according to “int val’.

*GetSwitchSoap..():*returns the state of soap switches.

HCProg:

SetLedProg(int val): Sets the program led on according to the passed value “ int val”.

HCControl:

SetLock(int val): Sets door lock led on or off based on the passed value.

*SetSpeed(int val):*Sets the rotation speed of the motor simulator.

*SetRotation(int val):*Sets rotation direction of the motor simulator.

*GetSwitchDoorLock():*returns the value of the door switch on the simulator.

HCWater:

GetWaterLevel(): returns the water level in the tank.

*GetWaterPressur():*return the water pressure state.

*SetSink(int value):*opens / closes the sink.

*SetDrain(int value):*start / stop drain.

HCHeater:

SetHeater(int val): sets the heater on to reach the passed temperature value this function checks if the water level is more than third and automatically turns the heater on or off based on the current temp.

GetTemp(): returns the current temperature of water.

Shared function:

GetButton(): returns an integer value to be used by objects to determine which button is pressed.

HWCC:

This class implements all the previously mentioned functions and has an instance on centipede class to communicate with the simulator via Arduino shield.

Coins:

This class has an instance of HCCoins class to communicate with the hardware it has a private function used to set the coins LED’s called SetLeds(), a constructor and three public functions:

GetAmount(): returns the coins amount in the tray.

getTHundredsLeds(): used to overcome a hardware limitation when setting soap2 and 200cents LED’s.

Clear():clears the coins LED’s and sets the amount to 0.

Init(): is called in LM class in order for the coins class to act automatically regarding buttons checking and LED’s setting.

Soap:

GetSoapOne() , GetSoapTow() : return the number of soap pieces.

Init(int val): is called in LM class in order for the soap object to monitor the switches and set the leds automatically .

ClearSoap1() , ClearSoap2() : turn soap led off and set the pieces variable to 0.

SetLeds(int Coinsth) : set the soap led's according to the number of soap pieces the variable coinsth is passed to overcome the previously mention hardware limitation.

Prog:

Init(bool val): is called in LM class in order for the prog object to monitor the program buttons and set the leds automatically .

GetProg(): returns the selected program.

SafeMode(int val , int &tim): this function implements the safe mode routine according to "int val" value:

- Val 0: set leds off
- Val 1: blink the current program led every second to show that LM is in safe mode.
- Val 2: blink all program leds to show that wash has ended with error.
- Val 3: set leds back on .

Controle:

SetLock(bool val): sets the door lock true or false.

SetSpeed(int val): sets rotation speed.

SetRotation(bool val): set rotation direction.

GetDoorLock(): get the door close switch status.

GetLock(): get lock status.

WashStart(): get start button status.

Water:

SetDrain(double targetLevel) : starts draining water until target level is reached.

SetSink(itn val):

- Val 0: opens the sink until water level in the tank is 0.
- Val 1: keep the sink open regardless to water level.

GetWaterLevel() , GetWaterPressur() : return current level / pressure status.

Heater:

GetTemp: returns temperature.

SetHeater(int targetTemp , double waterLevel):sets the heater to reach the target temperature.

ProgramInstuctions:

This class stores the wash values "instructions" of each program for different wash stages:

- Water Level per stage.
- Temperature per stage.
- Motor speed per stage.
- Program cost.
- Repetitions per stage.

All the Get functions return the value of the requested variable at a certain stage.

Excution:

Idle():
this function is continuesly checking for coins soap switches program select button door lock basicly all conditions to start the wash it returns true when it is possible to start a program.

washPrp():first (sub state)functionset the heater and water drain functions returns true when desired water level is reached.

motorPrp():second sub return true when the moror did rotate as many times as requiered by the program.

sink():sink returns true when water level is 0.

Wash():this function is used for the 4 wash stages prewash / mainwash1 / mainwash 2 / finalwash the parameters are used to pass which stage is the function used for :

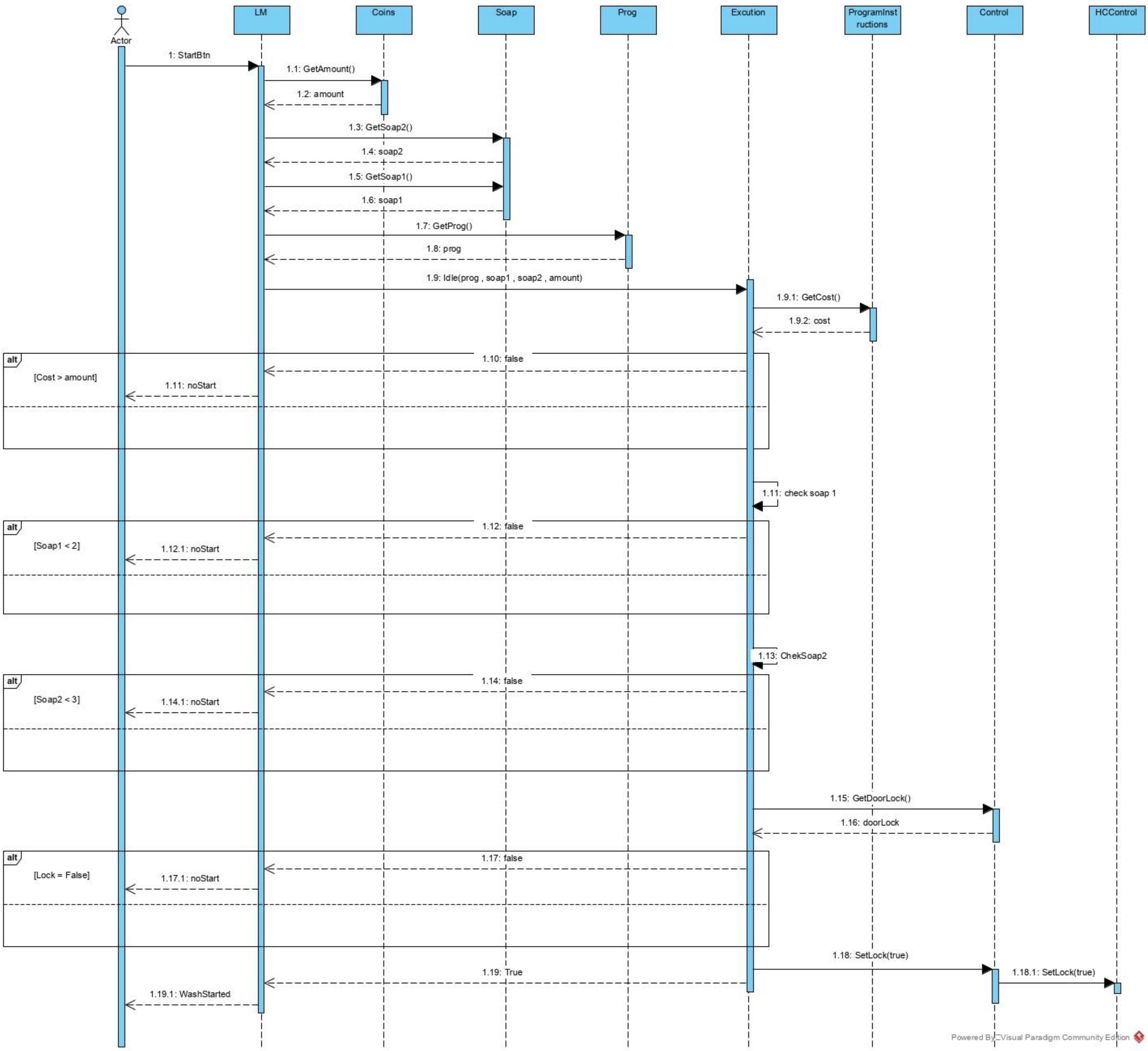
if stage is prewash the three sub state functions are called passing the parameters from program instuctions class .

so the function washprp motorprp and sink will act according to the description in program instructions

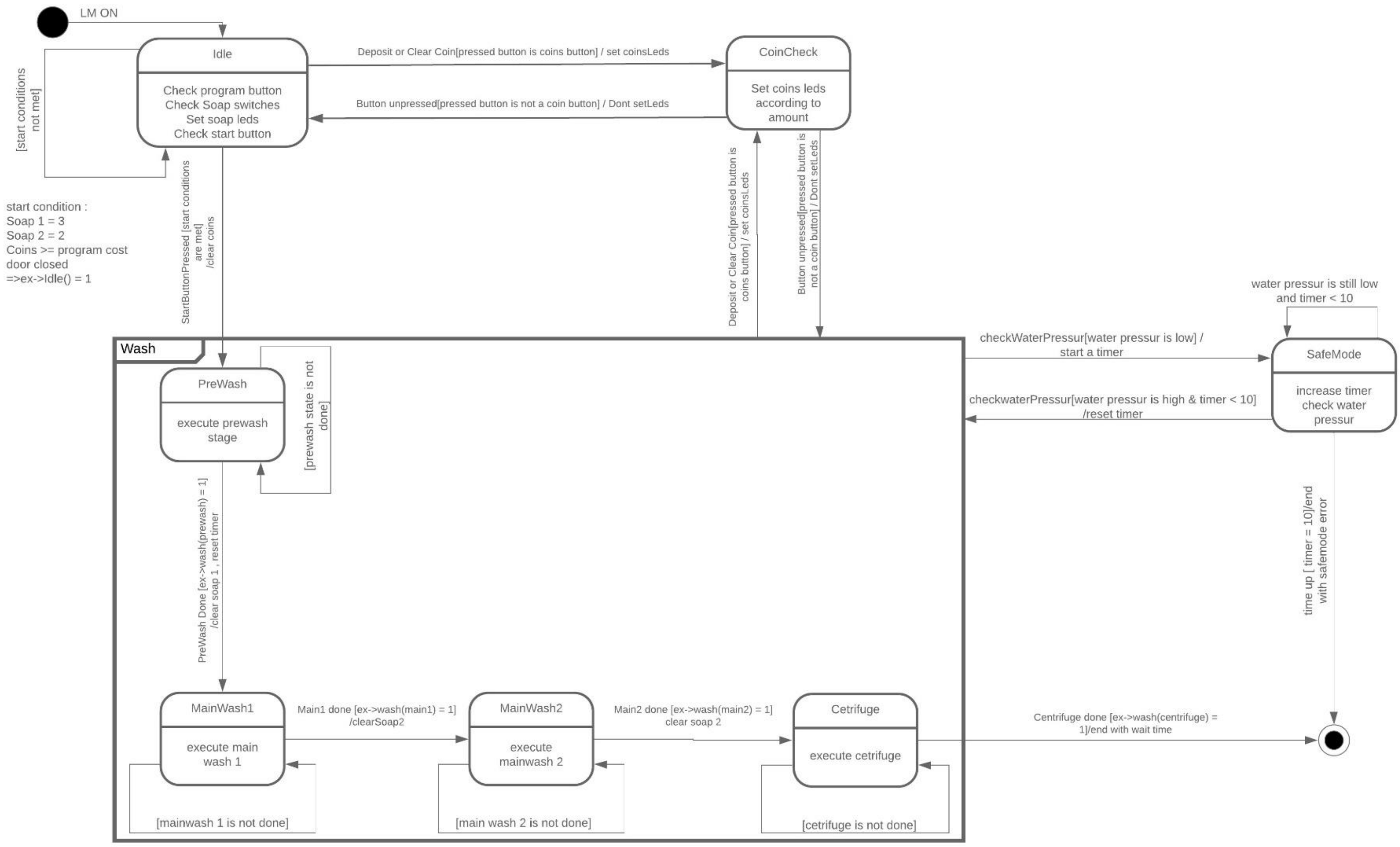
“cntr” is a variable in LM to keep track of repeats already done.

safeMode(): execute the safemode routine : turn heater drain off opens sink and return true when water tank is empty.

Sequence Diagram:



State Diagram:


LM ON
Idle
Check program button
Check Soap switches
Set soap leds
Check start button
[start conditions not met]
start condition :
Soap 1 = 3
Soap 2 = 2
Coins >= program cost
door closed
=>ex->Idle() = 1
Deposit or Clear Coin[pressed button is coins button] / set coinsLeds
Button unpressed[pressed button is not a coin button] / Dont setLeds
CoinCheck
Set coins leds according to amount
StartButtonPressed [start conditions are met] /clear coins
Deposit or Clear Coin[pressed button is coins button] / set coinsLeds
Button unpressed[pressed button is not a coin button] / Dont setLeds
Wash
PreWash
execute prewash stage
[prewash state is not done]
PreWash Done [ex->wash(prewash) = 1] /clear soap 1 , reset timer
MainWash1
execute main wash 1
[mainwash 1 is not done]
Main1 done [ex->wash(main1) = 1] /clearSoap2
MainWash2
execute mainwash 2
[main wash 2 is not done]
Main2 done [ex->wash(main2) = 1] clear soap 2
Cetrifuge
execute cetrifuge
[cetrifuge is not done]
Centrifuge done [ex->wash(centrifuge) = 1]/end with wait time
checkWaterPressur[water pressur is low] / start a timer
checkwaterPressur[water pressur is high & timer < 10] /reset timer
water pressur is still low and timer < 10
SafeMode
increase timer check water pressur
time up [timer = 10]/end with safemode error