NuSMV

Coffee Machine v 1.0

목차

MODULE:main

```
MODULE main
--main module variable
    --on , off --> power
    --ready -> user can order drink
    --working -> machine is in progress
    --pause -> machine wait for supply
    VAR
        state:{ready, working, pause};
        --power indicator
        power : boolean;
        --shortage indicator
        out_of_storage : boolean;
        out_of_coin: boolean;
        --what user can do
        button_type : {REFUND, Black_coffee, Milk_coffee, Pure_Milk};
        --what manager can do
        managing : {fill_coffee, fill_coin};

        coin_insert : process coinManager();
        manage_storage : process storageManaging();

    SPEC AG EX TRUE
-----------------------------------------------------------------------------------------------------------------------
```

MODULE:coinManager

```
MODULE coinManager(button_type,machine_state)
--control coffee machine coin
    VAR
        --insert : user can insert coin and raise current fund
        --stop   : while coffee machine is working or pause, user can't insert coin & do not raise current fund
        --refund : machine refund all current_fund to user
        state : {insert, stop, refund , purchase};
        --type of coin & price of all coffee
        coin_type : {10, 50, 100, 500} ;
        coffee_price :{200,250,300};
        --boundary of coin that machine can hold it
        current_fund: 0..5000;
        order_price: 0..500;
        storage_tcoin : 0..100;
        storage_fcoin : 0..100;
        storage_hcoin : 0..100;
        storage_fhcoin: 0..100;

    ASSIGN

        init(state) := insert;
        next(state) :=
        case
        machine_state = ready & button_type = Black_coffee | Milk_coffee | Pure_Milk : purchase;
        machine_state = ready & button_type = REFUND : refund;
        machine_state = working | pause = stop;
        TRUE : insert;
        esac
        --user insert coin
        init(current_fund) := 0
        next(current_fund) :=
        case
        state = insert & coin_type = 10 : current_fund = current_fund + 10;
        state = insert & coin_type = 50 : current_fund = current_fund + 50;
        state = insert & coin_type = 100 : current_fund = current_fund + 100;
        state = insert & coin_type = 500 : current_fund = current_fund + 500;
        TRUE : 0;
        esac
        --user buy stuff
        init(order_price) := 0
        next(order_price) :=
        case
        state = purchase : order_price =  order_price + 200;
        state = purchase : order_price =  order_price + 300;
        TRUE : 0;
        esac
```

```
INIT
    storage_tcoin : 100;
    storage_fcoin : 100;
    storage_hcoin : 100;
    storage_fhcoin: 100;

DEFINE
    current_fund :=  current_fund - order_price;
```