

Game Development with TI-RTOS

Akshay Hegde, Umang Deshpande Mentors: Sanam Shakya, Vishwanathan Iyer

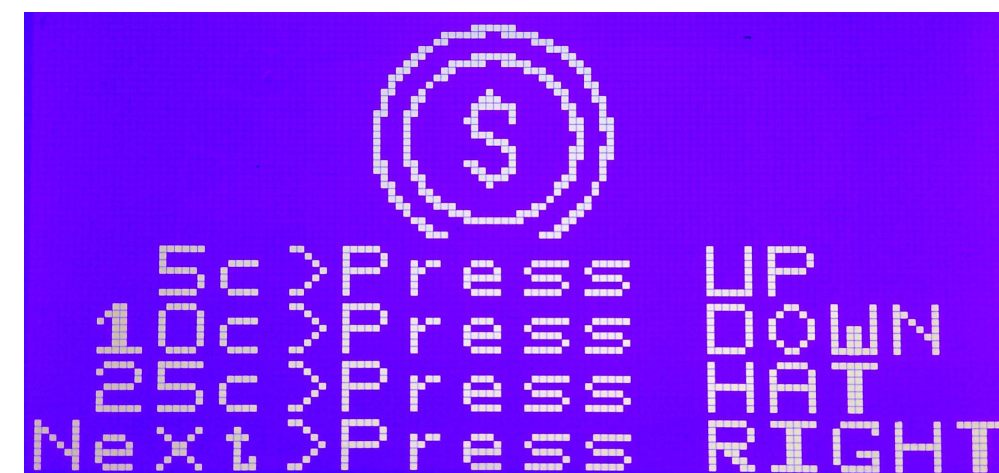
WHY RTOS?

- *Task Scheduling*- Better program flow and event response
- *Multitasking*-Executing a number of task simultaneously
- *Deterministic Behaviour*- Handles interrupt with defined time
- *Shorter ISRs*- No delays,Faster response
- *Intertask Communication*-Sharing of data,memory,resources
- *System Management*-Focus is on application

WHY STATECHARTS?

- Standard to visualize any system.
- Describe system in a modular way.
- High level of abstraction.

VENDING MACHINE



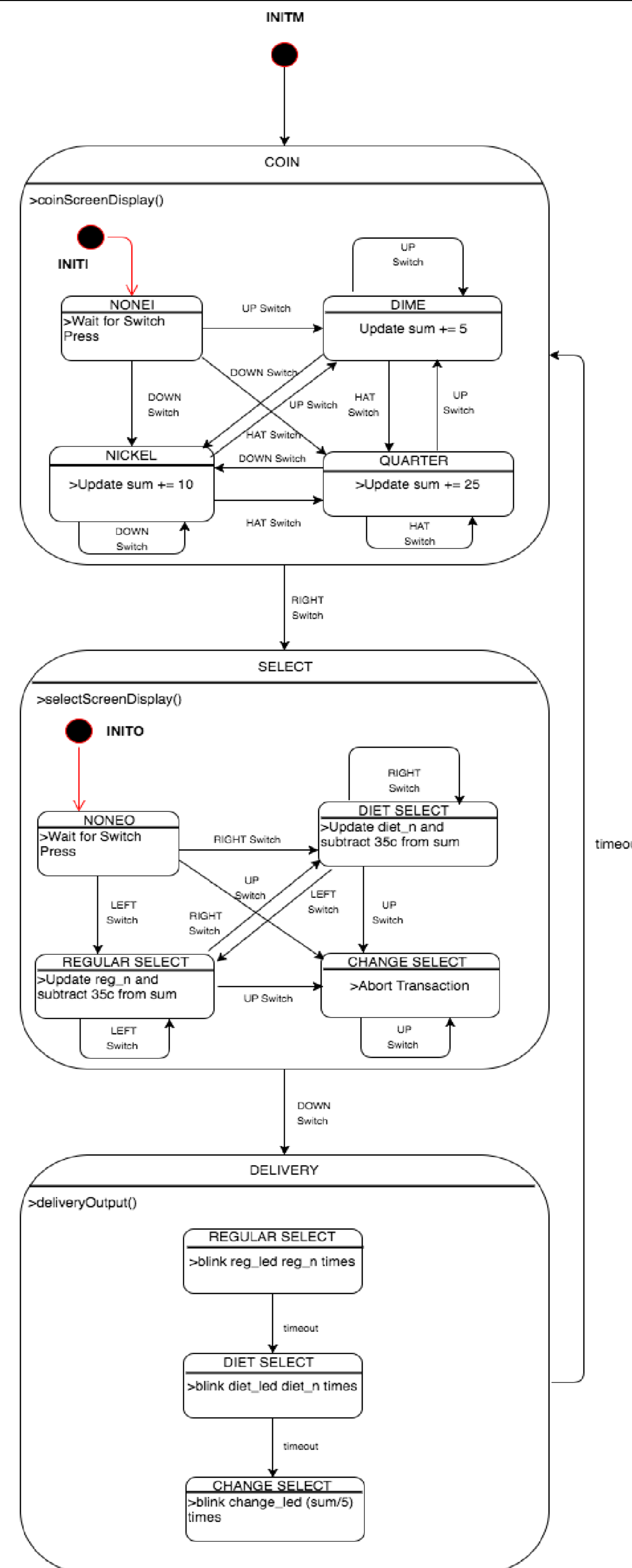
Abstract implementation using
Switch Case Statechart and TI-RTOS

STATECHART:

- *COIN* state - User enters money using Switch presses
- *SELECT* state - Selection of Soda using Select Screen.
- *DELIVERY* state - Soda and Change delivered using LED blinks.

RTOS:

- *readInputTask* - Handles Switch Input.
- *updateOutputTask* - Handles GLCD and LED output.
- 30 ms Task Scheduling is used.



VENDING MACHINE STATECHART

THE BREAKOUT GAME



The Classic Breakout Game Implemented using
Switch Case Statechart and TI-RTOS

STATECHART:

- *BALL DIRECTION STATE MACHINE* - Init, UpLeft, UpRight, DownLeft and DownRight states
- *BRICK TYPE STATE MACHINE* - Easy(1 hit), Medium(2 hit), Hard(3 hit), Magic1, Magic2 and None states.
- *PADDLE MODE STATE MACHINE* - Normal and Special states.
- *CURSOR POSITION STATE MACHINE* - One, Two and Three states.
- *SWITCH DEBOUNCING STATE MACHINE* - Idle, Press and Release states.
- *GAME SCREEN STATE MACHINE* - Menu, Settings, Instructions, Gameplay, Victory and Game Over states.

RTOS:

- *readInputTask* - Handles 5 switches and Joystick ADC.
- *displayOutputTask* - Handles GLCD, Buzzer, LED and Vibration Motor.
- 40 ms Task Scheduling is used.