Implementation of State Machines on Embedded Systems

João Carrasco 80305, Daniel Oliveira 89208, and João Carvalho 106310

Universidade de Aveiro, Portugal

Abstract. Development of an embedded application that emulates a vending machine.

1 State Machine

In this project we resorted to state machines as the core logic to solve the current task. In this scenario we have defined 5 different states:

- **IDLE** The waiting/default state.
- **COININSERT** This state is active after any coin type have been inserted.
- **CHANGEPRODUCT** This state is active after the user trigger an event to change the selected product.
- **SELECTPRODUCT** This state is active after a select event has been triggered with the objective of purchasing the selected product.
- **RETURNCREDIT** This state is active after a return event is triggered.

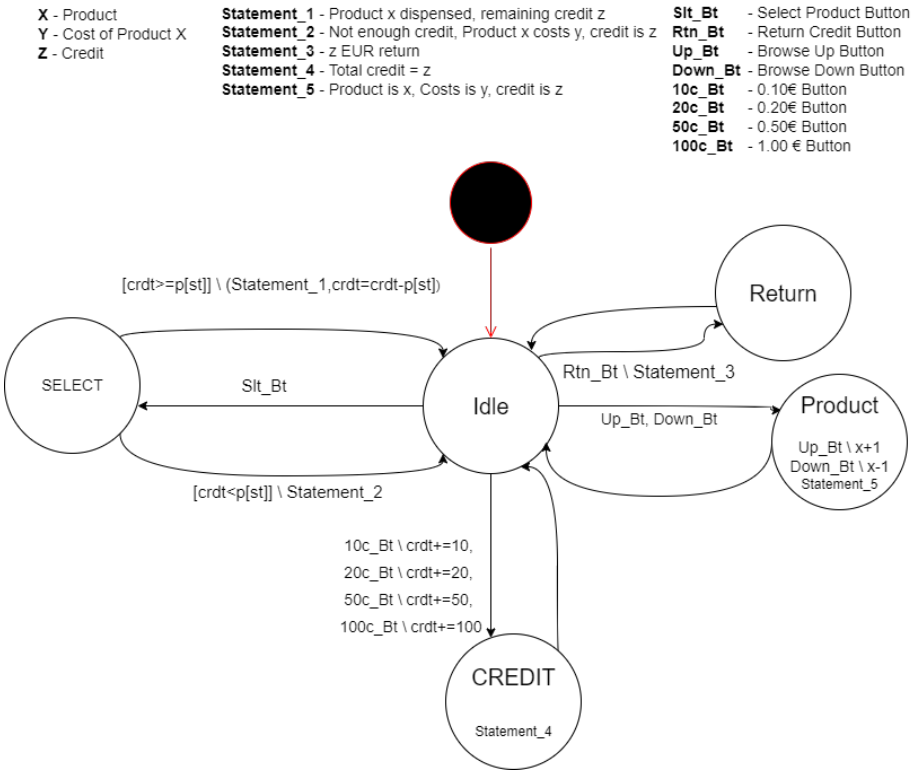
All states aside from the IDLE state are temporary and will transition to IDLE after its corresponding logic is executed. The only exception to simply returning to the IDLE state is in the SELECTPRODUCT state.

In this state, the behavior is different according to the current credit compared to the price of the product, if the credit is sufficient then it will be discounted to the current amount, otherwise it will display a "not enough credit" message.

2 Testing and results

First, we started to debug the developed code and created Unity tests to confirm that all parts of the code were functioning as expected (unit testing). All the tests were successful making the code ready to be modified for the Nordic Kit. The implemented debouncer works well to disable multiple unintentional triggers. The only issue faced during this assignment was the inclusion of header files on the project. We are aware that these kinds of files are part of good programming habits but for some reason, the studio was not helping. Aside from these issues, the project was built successfully using a single file. The code was documented using Doxygen and the state machine diagram was made in order to make simplify the comprehension.

May 17Th, 2022



References

1. https://github.com/Joaoprcf/setr-lab3