Instructor's Manual

Computers as Components

Principles of Embedded Computing System Design, 3rd edition

Marilyn Wolf
2011

Chapter 1

Q1-1 Briefly describe the distinction between requirements and specification.

The distinction between requirements and specifications stem from the fact that requirements are inherently an informal description of what the product should be. This description should be something that could be given to and understood by a marketing specialist and should include such items as size, weight, speed, UI, and performance. In contrast, specifications are a more formal, precise description of the product. This description should be written so as to allow the system's architects to implement the feature described in the requirements.

Q1-2: Give an example of a requirement on a computer printer.

Q1-3: Give an example of a requirement on a digital still camera.

Q1-4: Given an example of a specification on a computer printer, giving both type of specification and any required values. Take your example from an existing product and identify that product.

Q1-5: Given an example of a specification on a digital still camera, giving both type of specification and any required values. Take your example from an existing product and identify that product.

Q1-6 Briefly describe the distinction between specification and architecture.

The distinction between specification and architecture is that the specification is a detailed description of how a system behaves. In contrast, a system's architecture describes how a system is actually built. While in many systems a specification may lead directly to one type of architecture, ideally this is not the case.

Q1-7 At what stage of the design methodology would we determine what type of CPU to use (8-bit vs. 16-bit vs. 32-bit, which model of a particular type of CPU, etc.)?

We would determine what type of CPU we would use in the architecture design stage of the design methodology

Q1-8 At what stage of the design methodology would we choose a programming language?

We would choose a high-level programming language in the architecture design stage of the design methodology.

Q1-9: Should an embedded computing system include software designed in more than one programming language? Justify your answer.

We often design performance- or power-critical code in assembly language and large, complex code in a high-level language.

Q1-10 At what stage of the design methodology would we test our design for functional correctness?

We would test our design for functional correctness in both the component design/building phase and the system integration phase of the design methodology

Q1-11 Compare and contrast top-down and bottom-up design.

In a bottom-up design, we begin with individual components that could comprise a system and work our way up to eventual complete a full system. In contrast, in a top-down design, we begin with an abstract notion of a device and work our way down to the components that comprise the system. Of course, both methods are similar from the standpoint that the net result is a finished product. However, they both obviously achieve this result in completely different ways. Furthermore, both methods are related since it would be very difficult to do a good top down design without any knowledge of underlying components and vice versa.

Q1-12: Give an example of a design problem that is best solved using top-down techniques.

Construction of large, complex software systems is a good problem for top-down design.

Q1-13: Give an example of a design problem that is best solved using bottom-up techniques.

Estimation of performance and power costs is based on bottom-up information.

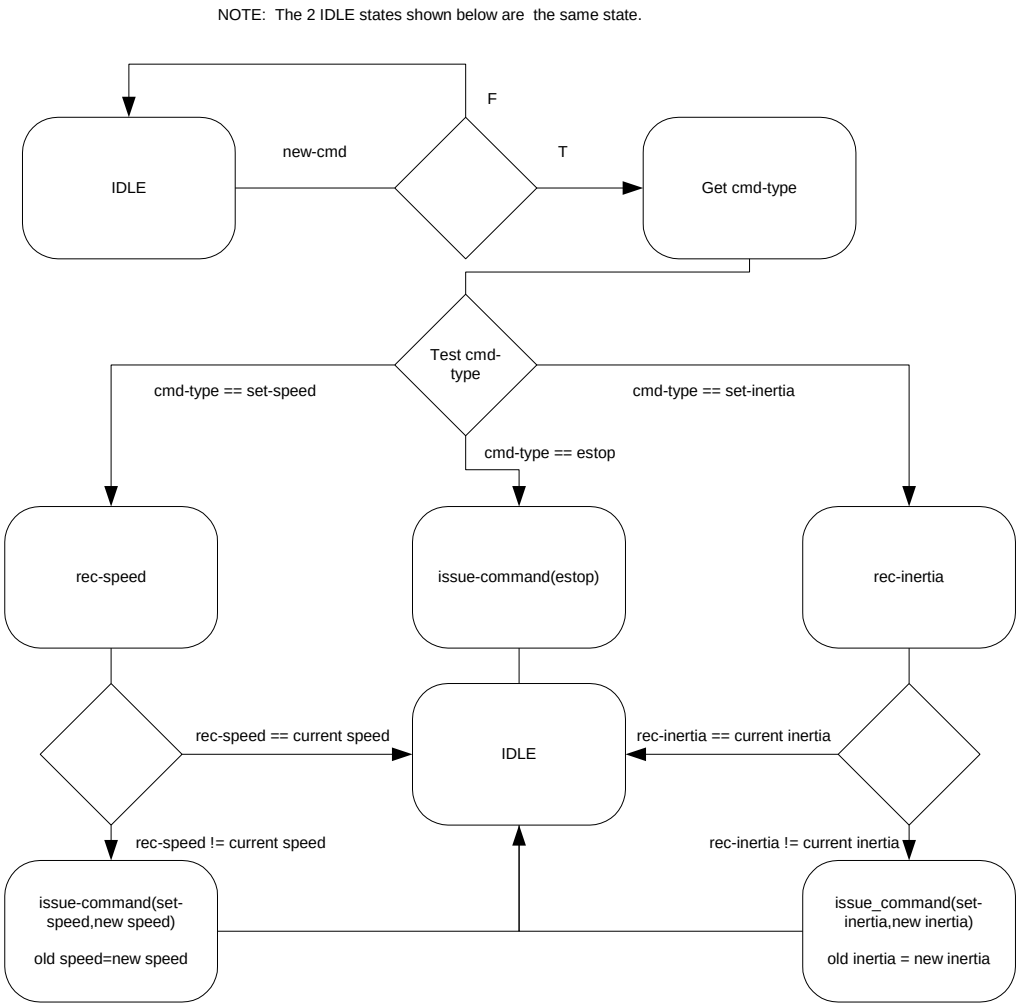
Q1-14 Provide a concrete example of how bottom-up information from the software programming phase of design may be useful in refining the architectural design.

Suppose that the information obtained from the software programming phase indicates that the application being written does the majority of its work in floating point. If this knowledge had been known during the architecture design phase, it would be possible to make the hardware more efficient when doing floating-point calculations.

Q1-15 Give a concrete example of how bottom-up information from I/O device hardware design may be useful in refining the architectural design.

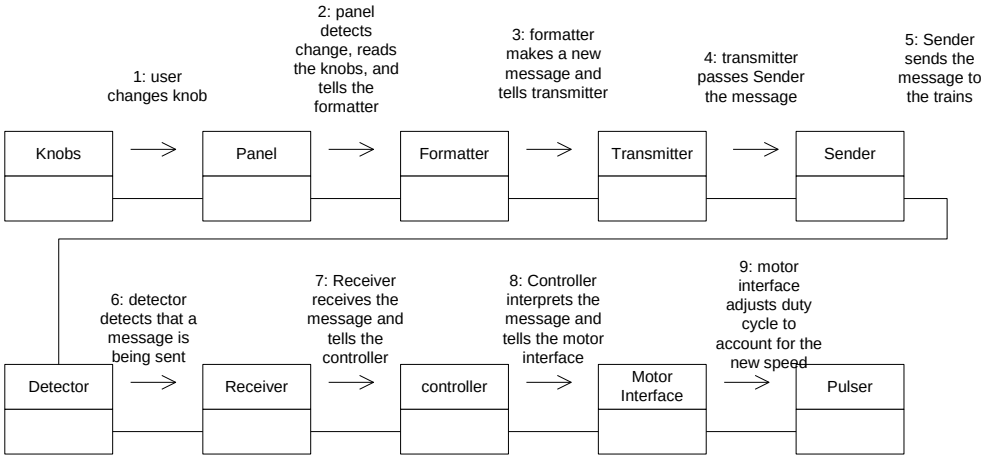
Suppose the data I/O design revealed that the device being made needed some high speed external data connection that required a higher bus bandwidth than a regular interrupt driven system can provide. If this info had been known during the architecture design phase, it might have been possible to add something like a DMA controller to interface directly with the data I/O device.

Q1-16 Create a UML state diagram for the *issue-command()* behavior of the *Controller* class of Figure XXX.

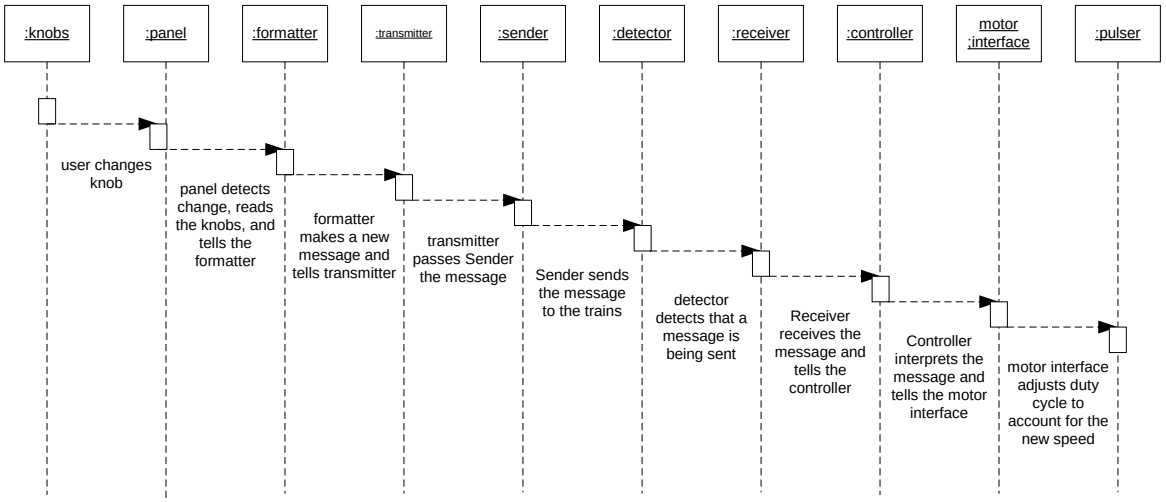


Q1-17 Show how a ***Set-speed*** command flows through the refined class structure described in Figure XXX, moving from a change on the front panel to the required changes on the train:

a) Show it in the form of a collaboration diagram.

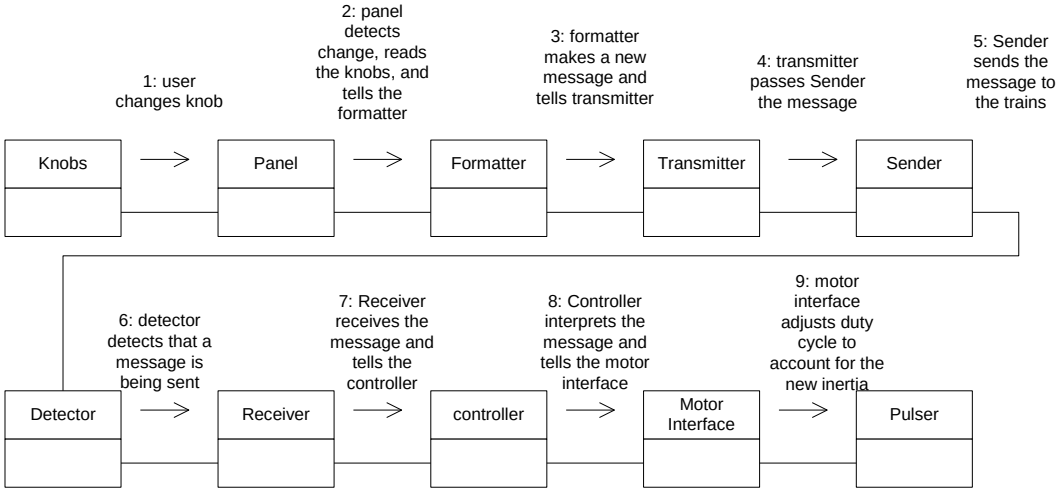


Show it in the form of a sequence diagram.

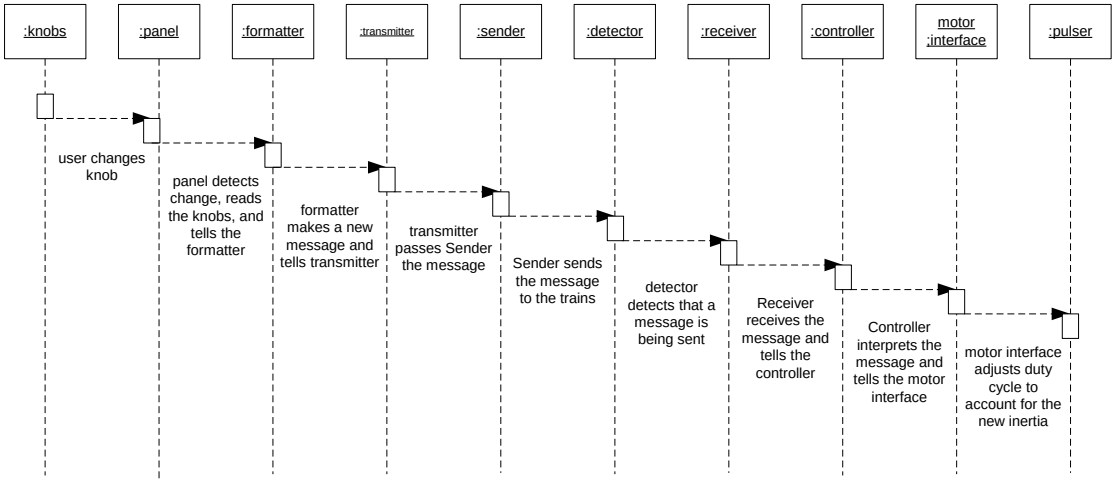


Q1-18 Show how a ***Set-inertia*** command flows through the refined class structure described in **Error! Reference source not found.**, moving from a change on the front panel to the required changes on the train:

a) Show it in the form of a collaboration diagram.

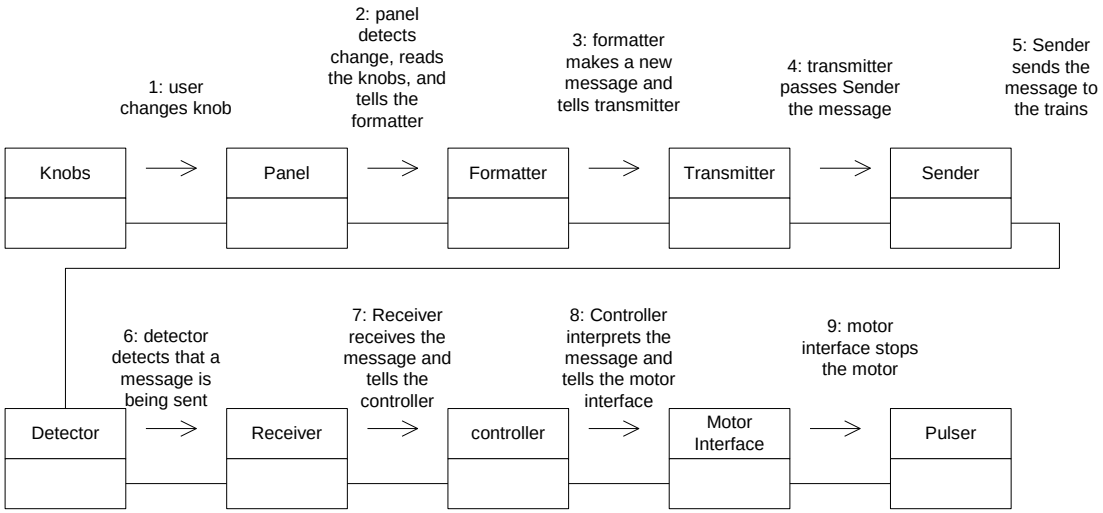


b) Show it in the form of a sequence diagram.

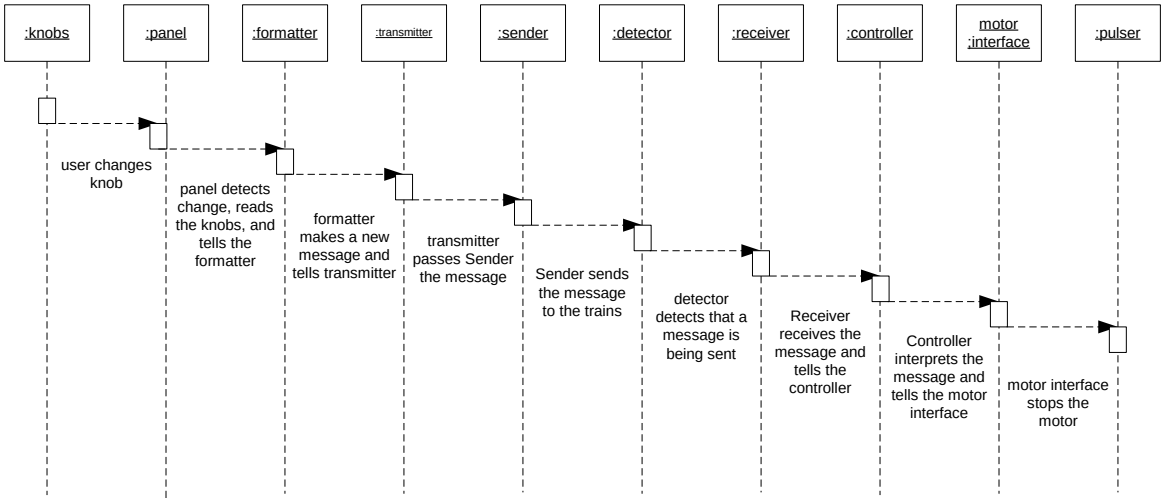


Q1-19 Show how an ***Estop*** command flows through the refined class structure described in Figure XXX, moving from a change on the front panel to the required changes on the train:

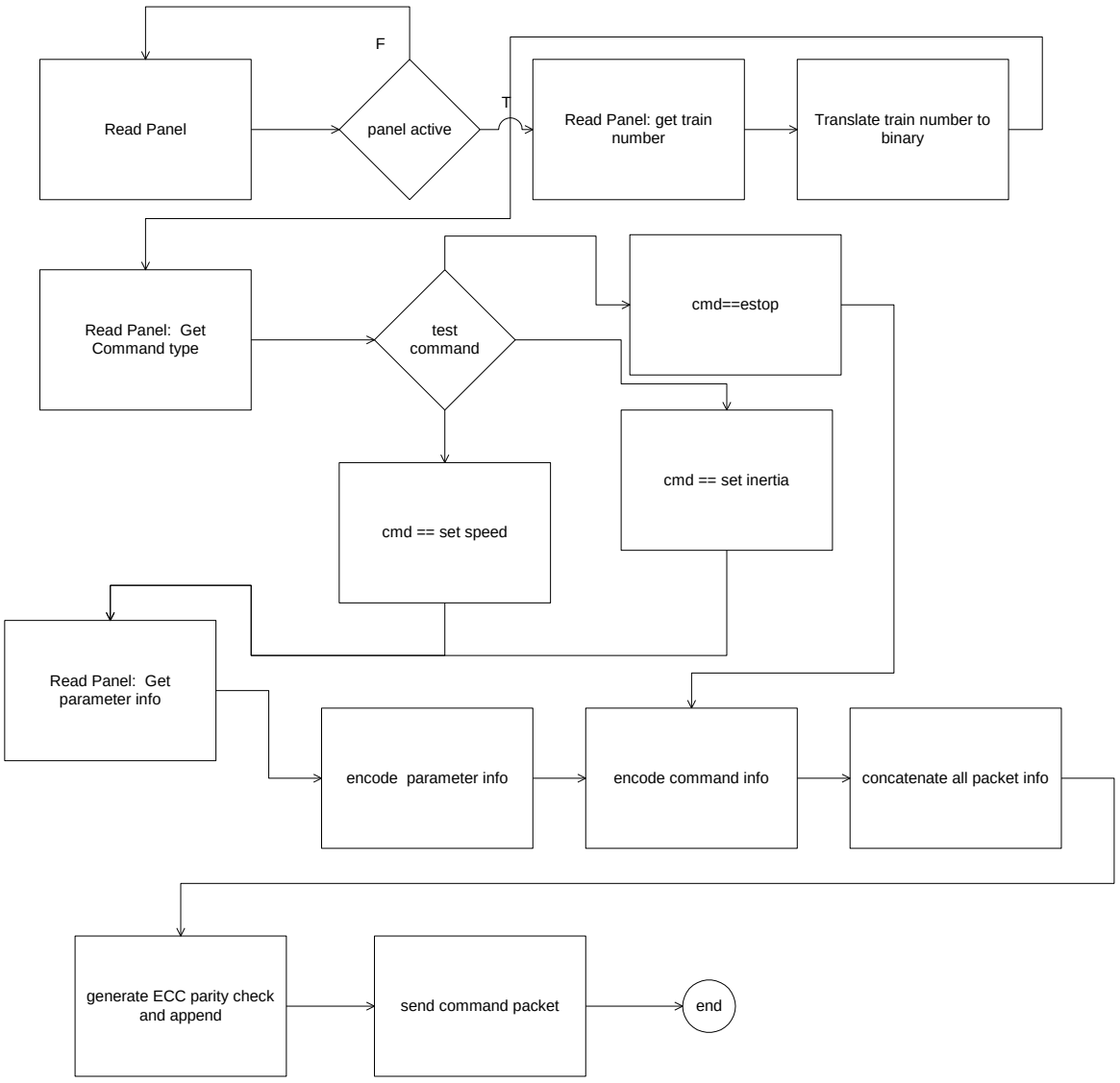
a) Show it in the form of a collaboration diagram.



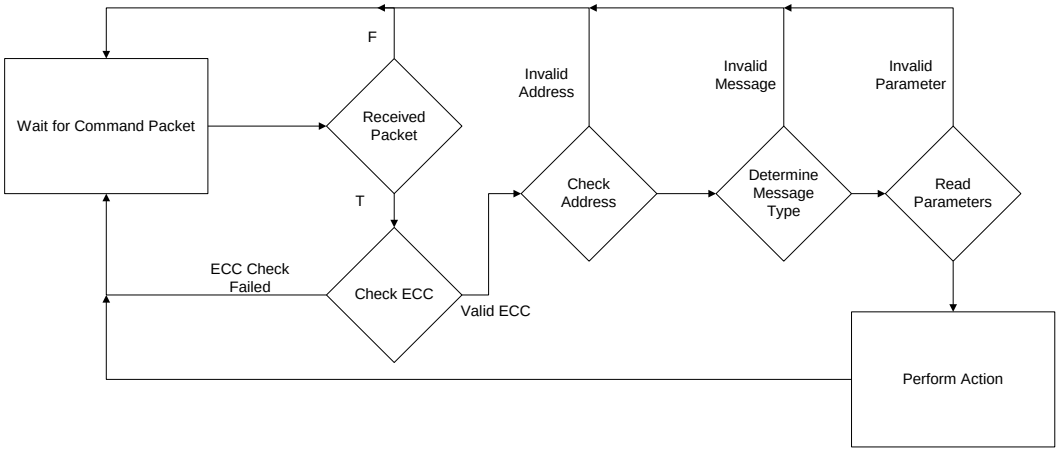
b) Show it in the form of a sequence diagram.



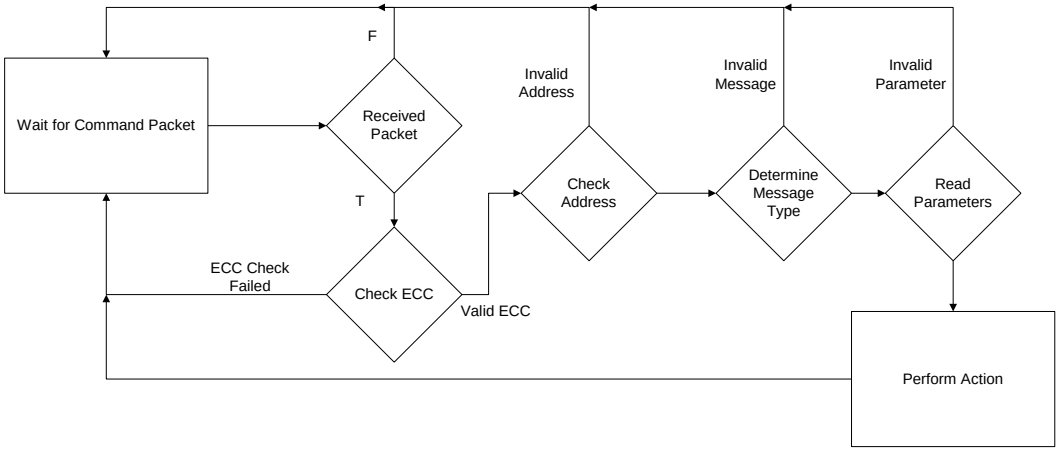
Q1-20 Draw a state diagram for a behavior that sends the command bits on the track. The machine should generate the address, generate the correct message type, include the parameters, and generate the ECC.



Q1-21 Draw a state diagram for a behavior that parses the received bits. The machine should check the address, determine the message type, read the parameters, and check the ECC.



Q1-22 Draw a class diagram for the classes required in a basic microwave oven. The system should be able to set the microwave power level between 1 and 9 and time a cooking run up to 59 minutes and 59 seconds in one-second increments. Include * classes for the physical interfaces to the telephone line, microphone, speaker, and buttons.



Q1-23 Draw a collaboration diagram for the microwave oven of Figure XXX. The diagram should show the flow of messages when the user first sets the power level to 7, then sets the timer to 2:30, and then runs the oven.