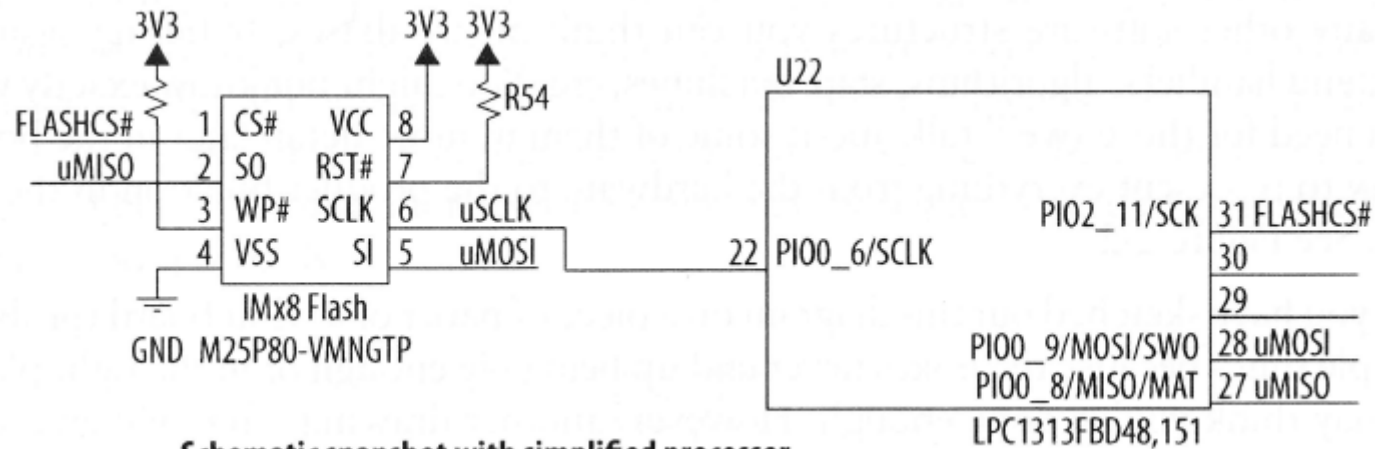


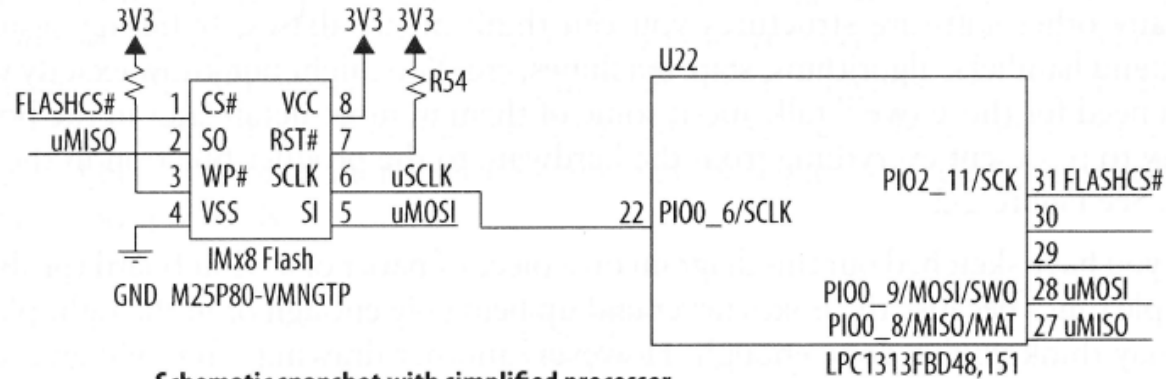
# Creating a System Architecture\*



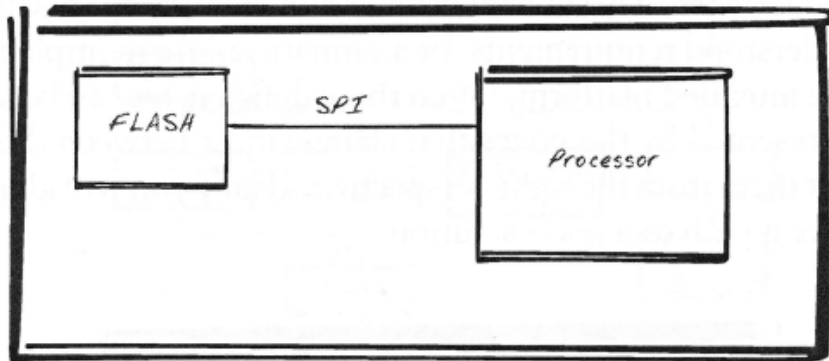
Schematic snapshot with simplified processor

\* Source: White, Making Embedded Systems

# Block Diagram

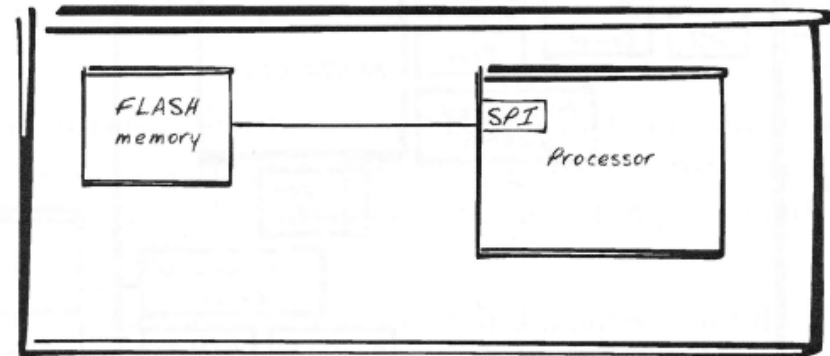


Schematic snapshot with simplified processor



Hardware block diagram

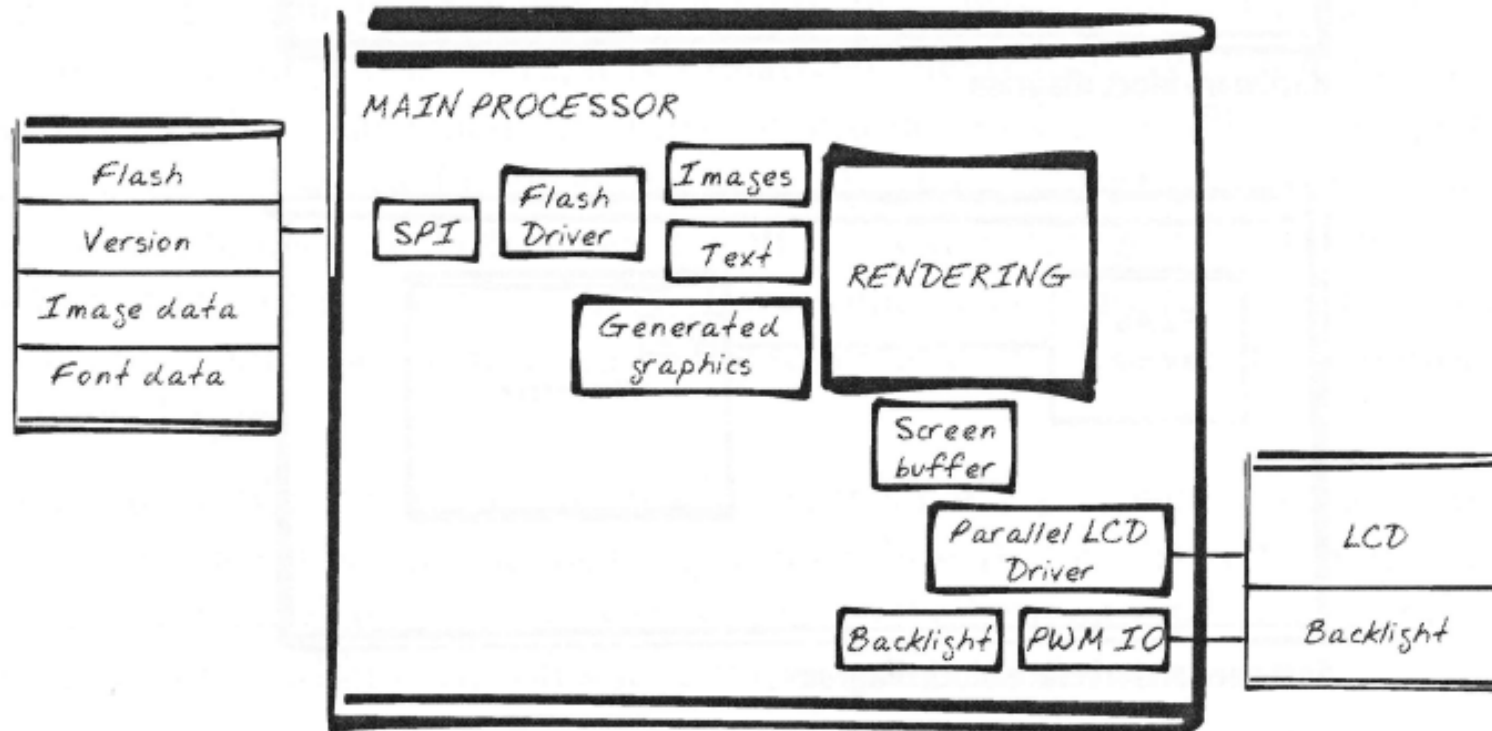
Hardware Block Diagram



Software architecture block diagram

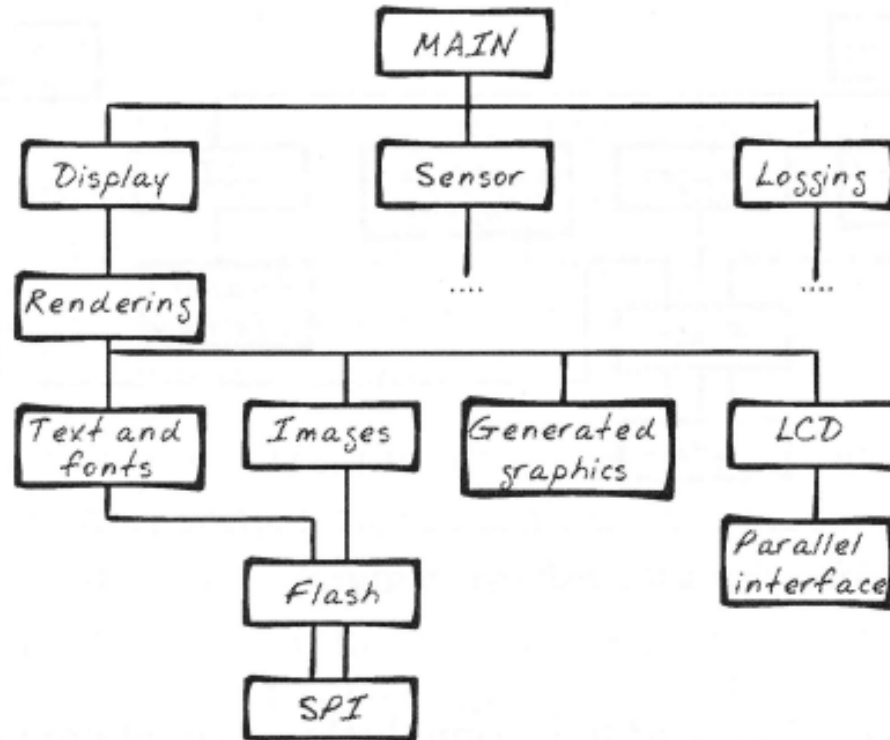
Software Architecture Block Diagram

# Software Block Diagram



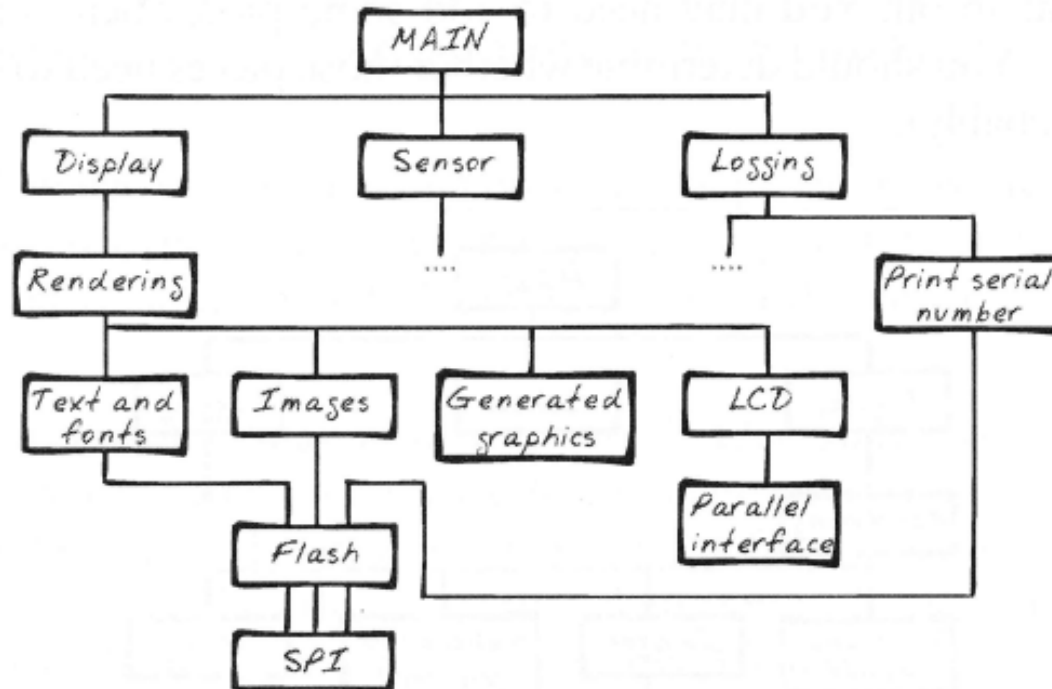
Sketch the system , try to figure out as many boxes as possible, combine them later  
Looking at various views may show you some hidden spots with critical bottlenecks,  
poorly understood specifications or failure to implement on intended platform  
Identify tricky modules and see a path to a good solution.

# Hierarchy of Control



A hierarchical view of the software. Main is highest level.  
Fill in next levels with algorithm-related objects.  
It shows discrete components and which components call others

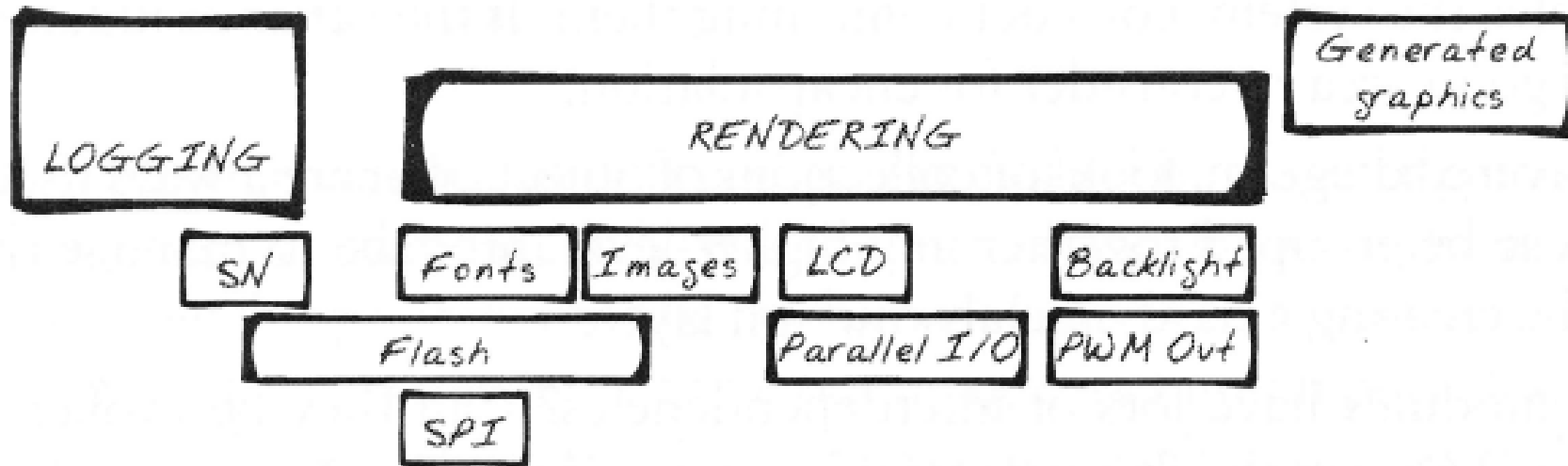
# Hierarchy of Control, shared resource



Each time something that this is added, some little piece where you are using A and B and have to consider a potential interaction with C, the system becomes a little less robust.

Shared resources may cause pains in the design, implementation and maintenance phases of the project.

# Layered Software Architecture Diagram



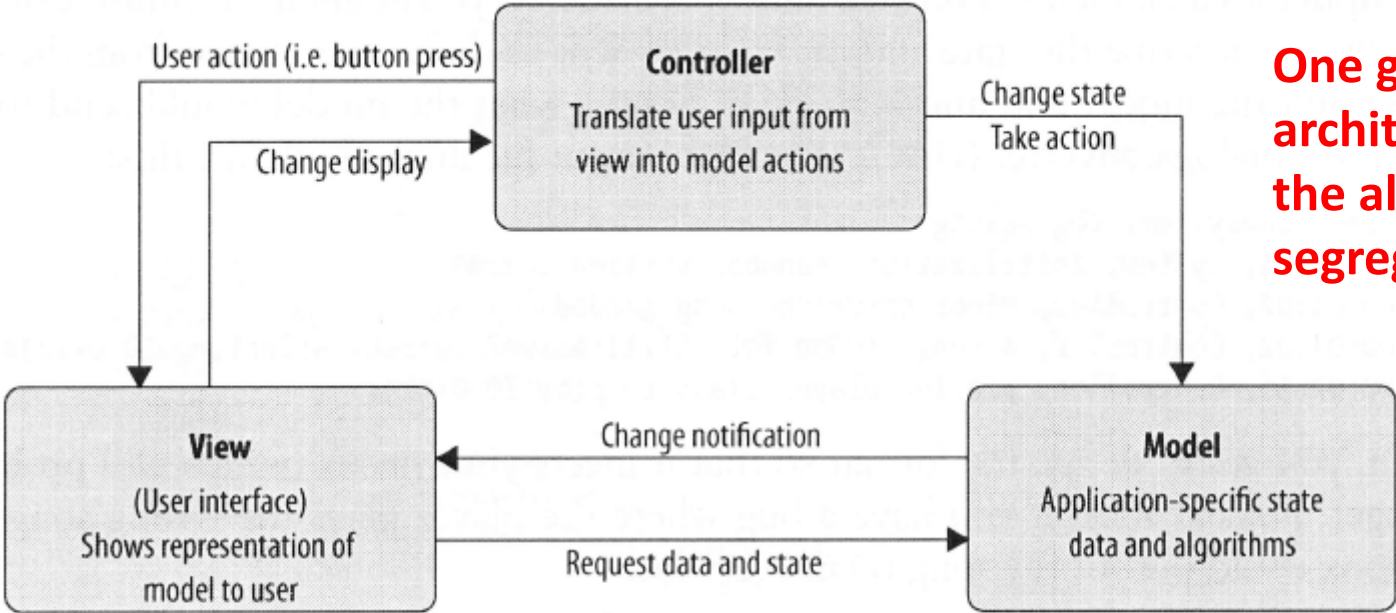
Represent objects by their estimated size. Let the size reflect the complexity. Start at the bottom of the page and draw boxes for the things that go off the processor (communication boxes).

Add to the diagram the items that use the lowest layer.

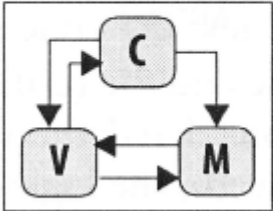
Each object that uses something below it should touch all of the things it uses.

**This shows you where the layers in your code are.**

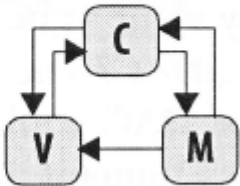
# Model-View-Controller, algorithm boxes



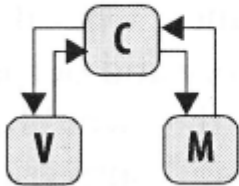
**One goal of good architecture is to keep the algorithm as segregated as possible**



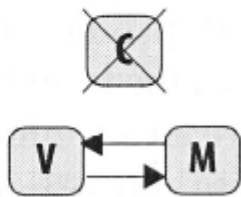
Shown here



Model receives data only from controller



Controller is translator



Model-View pattern