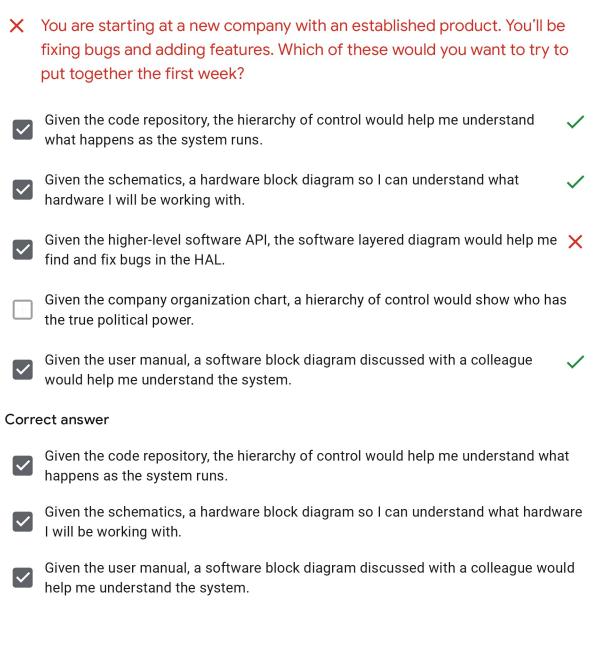
Quiz - Layer Diagrams

Reminder: if you have questions about anything on the quiz, try not to spoil any answers for others who haven't yet taken the quiz.

Email *

bgitego@gmail.com



Feedback

The hardware and software block diagrams are incredibly useful for understanding the system. The hierarchy of control will help you figure out where the bugs and features are in the code.

What does the layered view add that the software block and hierar control don't have?	chy of		
An obvious hardware abstraction layer	✓		
Path showing flow of control through the code			
A broad overview of the system			
Detailed state machine descriptions			
A view of the groups of modules that work together	✓		
Feedback The layered software diagram identifies modules that can be grouped together as one subsystem if they don't interact with other areas. It can also provide horizontal layers to the code (like a HAL) that might divide a project up by low-level software and higher-level software.			

×	When we talk about modular software, which of the following points are important:	
/	Information hiding	✓
/	Dependency hiding	×
/	Few dependencies between modules	✓
/	Modules acting independently of each other	✓
/	Encapsulating modules with a minimum API	✓
Corre	ect answer	
/	Information hiding	
/	Few dependencies between modules	
/	Modules acting independently of each other	
/	Encapsulating modules with a minimum API	
	eedback iding dependencies makes it harder to understand, maintain, and debug a system.	

Global variables are often very useful. What are the downsides?			
It creates dependencies between modules that are hard to remember and predict.			
They may be modified in places you don't expect (interrupts, other threads) and that may not be safe.	✓		
Sometimes there is only one type of a particular processor interface so it makes sense to have a global point to it.			
Variables that are global can be modified from anywhere.	×		
The static and volatile keywords make globals more useful.			
ect answer			
It creates dependencies between modules that are hard to remember and predict.			
They may be modified in places you don't expect (interrupts, other threads) and the may not be safe.	nat		
Feedback A global variable may be modified in places you don't expect including interrupts and other modules. This causes a dependency that can be difficult to remember (especially if you have a lot of global variables).			
	It creates dependencies between modules that are hard to remember and predict. They may be modified in places you don't expect (interrupts, other threads) and that may not be safe. Sometimes there is only one type of a particular processor interface so it makes sense to have a global point to it. Variables that are global can be modified from anywhere. The static and volatile keywords make globals more useful. ect answer It creates dependencies between modules that are hard to remember and predict. They may be modified in places you don't expect (interrupts, other threads) and the may not be safe. eedback global variable may be modified in places you don't expect including interrupts and other odules. This causes a dependency that can be difficult to remember (especially if you		

This form was created inside of Classpert.

Google Forms